Teoria de la computació

Llenguatges regulars i incontextuals

Rafel Cases
Lluís Màrquez

Índex

4 Autòmats finits

5 Minimització d'autòmats finits

6 Expressions regulars i gramàtiques regulars

7 Propietats d'iteració

8 Autòmats amb pila

Capítol 1 Llenguatges formals

1.1 Introducció

A fi de donar sentit als conceptes que introduirem a continuació, començarem considerant un exemple de *grafs hamiltonians.*[1] Sigui $G = \langle V, E\rangle$ un graf format per un conjunt finit V de vèrtexs i un conjunt E d'arestes. Sabem que un *camí* és una seqüència d'arestes diferents en què el segon vèrtex de cadascuna (a excepció del de l'última) coincideix amb el primer vèrtex de l'aresta següent, i en què tots els vèrtexs que hi intervenen són diferents (a excepció, eventualment, del primer i l'últim). Recordem també que un *cicle* és un camí en què el primer i l'últim vèrtex són el mateix. Anomenem *cicle hamiltonià* aquell que passa per tots els vèrtexs del graf sense passar dues vegades per cap d'ells. Diem finalment que un graf és *hamiltonià* quan conté algun cicle hamiltonià. A la figura 1.1 veiem dos exemples de grafs, un de hamiltonià i un que no ho és.

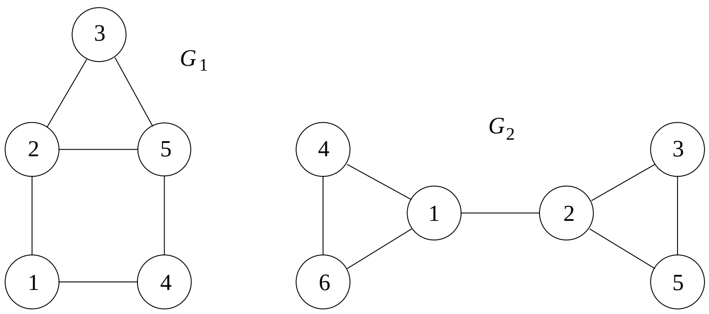

Fig. 1.1 *Un graf G_1 hamiltonià i un graf G_2 no hamiltonià*

Considerem un programa que prengui com a entrada un graf finit i calculi si és hamiltonià o no. Qualsevol que sigui aquest programa, haurà de partir d'una *codificació* preestablerta per als grafs finits. Considerem, per exemple, la codificació consistent a enumerar arbitràriament els vèrtexs i a escriure els conjunts de vèrtexs i arestes com fem a continuació per als grafs de la figura 1.1:

$$G_1 : \quad \langle\langle 1;2;3;4;5\rangle\,\langle(1\,;2)(1\,;4)(2\,;3)(2\,;5)(3\,;5)(4\,;5)\rangle\rangle$$

$$G_2 : \quad \langle\langle 1;2;3;4;5;6\rangle\,\langle(1\,;2)(1\,;4)(1\,;6)(2\,;3)(2\,;5)(3\,;5)(4\,;6)\rangle\rangle.$$

[1] Podeu trobar una exposició detallada del tema dels grafs hamiltonians a la secció 7.2 de [CFS94]

Observem que les codificacions de grafs grans són, lògicament, més llargues que les de grafs petits, però que el nombre de símbols utilitzats, quinze en total, és independent d'aquesta llargària. Aquest conjunt de símbols és

$$\{\, \langle\, , \rangle\, , (\, ,)\, , ;\, , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \,\}$$

i s'anomena *alfabet* de la codificació. Qualsevol seqüència construïda amb aquests símbols s'anomena un *mot* sobre aquest alfabet. Sols una petita part d'aquests mots té sentit com a codificació d'algun graf. I tan sols una part molt petita (tanmateix infinita) correspon a grafs hamiltonians. El conjunt de tots els mots que són codificacions de grafs hamiltonians s'anomena el *llenguatge* dels grafs hamiltonians. El que fa formalment el programa que estem considerant és determinar si un mot donat pertany o no a aquest llenguatge.

Aquesta manera de formalitzar els problemes de càlcul —i més en particular, els problemes *decisionals*— és el punt de partida de la *teoria de llenguatges formals*. N'estudiarem a continuació els conceptes bàsics.

1.2 Alfabets i mots

Un *alfabet* és un conjunt finit no buit, els elements del qual s'anomenen *símbols*. Exemples d'alfabets comuns són l'alfabet llatí de 26 símbols {A, ... ,Z}, l'alfabet decimal {0, ... ,9}, l'alfabet binari {0,1} i l'alfabet ASCII de 256 símbols. Utilitzem habitualment la lletra grega Σ per referir-nos a un alfabet qualsevol. Donat un alfabet, un *mot* és una seqüència finita de símbols juxtaposats d'aquest alfabet. La *longitud* d'un mot és el nombre de símbols que el componen. Anomenem *mot buit* el mot de longitud zero. El representem per λ. Representem per $|w|$ la longitud d'un mot w. Cal distingir aquesta notació de la que representa la *cardinalitat* o nombre d'elements d'un conjunt finit. Nosaltres representem per $\|C\|$ la cardinalitat d'un conjunt C.

Anomenem *submot* (o també *factor*) d'un mot donat qualsevol subcadena de símbols consecutius d'aquest mot. *Prefixos* i *sufixos* són casos particulars de submots, quan aquests es troben al principi o al final del mot. El terme *infix* és sinònim de submot. Un factor s'anomena *propi* quan no és ni λ ni el mot considerat.

Exemple 1.1 El mot aba té el conjunt de factors següent,

$$\{\lambda, a, b, ab, ba, aba\},$$

els prefixos següents

$$\{\lambda, a, ab, aba\}$$

i els sufixos següents

$$\{\lambda, a, ba, aba\}.$$

Si w i v són dos mots qualssevol sobre un cert alfabet Σ, representem per $|w|_v$ el nombre d'*ocurrències* de v com a submot de w, tenint en compte que poden haver-hi ocurrències superposades. Formalment, aquest nombre es pot expressar així:

$$|w|_v = \|\{x \mid \exists y \; xvy = w\}\|.$$

Exemple 1.2 Si representem per w el mot $aababa$ tenim:

$$|w| = 6 \qquad |w|_a = 4 \qquad |w|_b = 2 \qquad |w|_{aba} = 2 \qquad |w|_{bba} = 0.$$

En general, si considerem les ocurrències de la forma $|w|_a$, on w i a són, respectivament, un mot i un símbol qualssevol d'un cert alfabet Σ, tenim $|w| = \sum_{a\in\Sigma} |w|_a$.

Representem per Σ^* el conjunt infinit de tots els mots possibles sobre un cert alfabet Σ. Sovint ens caldrà explorar els mots d'alguns subconjunts de Σ^*. Necessitarem partir d'algun ordre entre aquests mots que ens permeti fer l'exploració sistemàticament. L'ordre usual dels diccionaris (l'anomenat ordre *lexicogràfic*) té l'inconvenient que no és un ordre *ben fonamentat*, i no permet aplicar el mètode d'inducció de manera segura sobre subconjunts infinits de mots. En efecte, la successió de mots següent, per exemple, és estrictament decreixent i infinita,

$$b > ab > aab > \ldots > a^n b > \ldots$$

L'ordre que considerarem sobre Σ^* parteix de l'ordre donat sobre Σ, classifica els mots de Σ^* per la seva longitud i ordena lexicogràficament els mots d'una mateixa longitud. Així, per a $\Sigma = \{a\,, b\}$, i considerant $a < b$, obtenim l'ordenació següent:

$$\lambda < a < b < aa < ab < ba < bb < aaa < \ldots$$

1.3 Operacions amb mots

Concatenació

En el conjunt de mots sobre un alfabet Σ qualsevol considerem l'operació que fa correspondre a cada parell de mots el mot format per la juxtaposició del primer i el segon. L'anomenem *concatenació* i la podem expressar formalment així

$$\begin{aligned} \Sigma^* \times \Sigma^* &\longrightarrow \quad \Sigma^* \\ (w_1, w_2) &\longmapsto w_1 \cdot w_2\,, \end{aligned}$$

on el punt que simbolitza l'operador se sol ometre quan no és necessari fer èmfasi en l'operació. Així, per exemple, si $w_1 = abb$ i $w_2 = ba$, tenim $w_1 w_2 = abbba$. És immediat verificar que es tracta d'una operació *associativa* i que té per element neutre λ. Així doncs, dota Σ^* d'estructura de *monoide*.[2] Observi's que es tracta d'un tipus molt particular de monoide, ja que tots els seus elements són *simplificables*. Recordem que, en àlgebra, un element x d'un monoide M s'anomena simplificable quan satisfà les dues condicions següents:

$$\forall y, z \in M \qquad \begin{cases} x \cdot y = x \cdot z \implies y = z \\ y \cdot x = z \cdot x \implies y = z. \end{cases}$$

[2] En àlgebra, s'anomena *monoide* una estructura formada per un conjunt i una operació interna que és associativa i té element neutre. (Podeu consultar la secció 9.4 de la referència [CFS94].)

És a dir, si l'element x apareix en una equació multiplicant, per l'esquerra o per la dreta, dos termes, podem *simplificar* l'equació esborrant-ne l'element x i reduint-la a la igualtat entre els termes considerats. També des d'aquest punt de vista algebraic, és interessant remarcar que l'aplicació que fa correspondre a cada mot la seva longitud és un *morfisme*[3] d'aquest monoide $\langle \Sigma^*, \cdot \rangle$ en el monoide additiu dels naturals $\langle \mathbb{N}, + \rangle$, ja que satisfà les dues condicions de morfisme

$$\forall x, y \in \Sigma^* \quad |x \cdot y| = |x| + |y| \qquad \text{i} \qquad |\lambda| = 0.$$

Utilitzem la notació exponencial per representar la concatenació repetida d'un mot amb si mateix. Així, per a qualsevol mot w fem

$$w^0 = \lambda \qquad \text{i} \qquad \forall i \geqslant 0 \quad w^{i+1} = w^i w.$$

Revessament

Anomenem *revessat* d'un mot w el mot format pels mateixos símbols de w escrits a l'inrevés. Representem per w^{R} el revessat de w. Formalment,

$$\lambda^{\mathsf{R}} = \lambda\,, \quad (a_1 a_2 \ldots a_n)^{\mathsf{R}} = a_n \ldots a_2 a_1.$$

Per a tot mot w, es té $(w^{\mathsf{R}})^{\mathsf{R}} = w$. Un *palíndrom* és un mot que és igual al seu revessat. Una definició recursiva de palíndrom és la següent:

1. λ és un palíndrom.
2. Tots els mots formats per un sol símbol són palíndroms.
3. Si w és un palíndrom, aleshores per a tot símbol a el mot awa també és un palíndrom.
4. No hi ha altres palíndroms que els obtinguts per les regles anteriors.

1.4 Llenguatges. Concatenació

Un *llenguatge* és qualsevol conjunt (finit o infinit) de mots sobre un alfabet determinat. Així, sobre l'alfabet $\Sigma = \{a, b, c\}$ podem considerar el llenguatge format per tots els mots que tenen el mateix nombre de a's, b's i c's. Formalment,

$$L = \{w \mid |w|_a = |w|_b = |w|_c\} = \{\lambda, abc, acb, bac, bca, cab, cba, \ldots\}.$$

Altres exemples de llenguatges (diferents entre si!) són $\varnothing$ i $\{\lambda\}$. El conjunt Σ^* de tots els mots sobre un alfabet Σ també constitueix un llenguatge. En el cas d'un alfabet d'un sol símbol, anomenat aleshores *uniliteral*, $\Sigma = \{a\}$, utilitzem la notació a^* per referir-nos al llenguatge $\{a\}^*$.

Sempre que es té definida una operació entre elements d'un conjunt, pot procedir-se a estendre-la com a operació entre subconjunts. D'aquesta extensió se'n diu *canònica*.

[3] Donats dos monoides qualssevol, $\langle M_1, \cdot \rangle$ i $\langle M_2, \cdot \rangle$, amb elements neutres e_1 i e_2, respectivament, una aplicació $h: M_1 \longrightarrow M_2$ s'anomena *morfisme* quan satisfà les condicions: $\forall x, y \in M_1 \quad h(x \cdot y) = h(x) \cdot h(y)$ i $h(e_1) = e_2$.

En el cas de la concatenació, passem d'una operació entre mots a una operació entre llenguatges. La concatenació de dos llenguatges és el llenguatge format per tots els mots obtinguts concatenant un mot del primer llenguatge amb un del segon. Formalment,

$$\forall L_1, L_2 \quad L_1 \cdot L_2 = \{x \cdot y \mid x \in L_1 \ \wedge \ y \in L_2\}.$$

Si considerem llenguatges definits sobre un mateix alfabet Σ, la concatenació de llenguatges, que també és associativa, converteix[4] $\mathcal{P}(\Sigma^*)$ en un monoide que té per element neutre el llenguatge $\{\lambda\}$.

Exemple 1.3 Siguin $L_1 = \{a, ab, baa\}$ i $L_2 = \{ab, ba\}$. Es té

$$L_1 L_2 = \{aab, aba, abab, abba, baaab, baaba\}.$$

Exemple 1.4 Sobre l'alfabet $\Sigma = \{a, b\}$, siguin: L_1 el conjunt de mots que tenen exactament tantes a's com b's; L_2 el conjunt de mots que tenen almenys tantes a's com b's; i L_3 el conjunt de mots que tenen almenys tantes b's com a's. Tenim

$$\begin{array}{ll} L_1 L_1 = L_1 & L_1 L_2 = L_2 L_1 = L_2 \\ L_2 L_2 = L_2 & L_1 L_3 = L_3 L_1 = L_3 \\ L_3 L_3 = L_3 & L_2 L_3 = L_3 L_2 = \Sigma^*. \end{array}$$

En molts casos és possible identificar la concatenació de dos llenguatges amb el seu producte cartesià. Podem observar aquest fenomen en el primer dels exemples anteriors, en què cada mot de $L_1 L_2$ admet una única descomposició en un prefix de L_1 i un sufix de L_2. Una condició suficient perquè això passi és que els dos llenguatges considerats estiguin definits sobre alfabets disjunts. En canvi, no és ni necessari ni suficient que els dos llenguatges siguin mútuament disjunts. Així, per exemple, tenim

$$\{a, ab\} \cdot \{b, ab\} = \{ab, aab, abb, abab\},$$

en canvi,

$$\{a, ab\} \cdot \{b, bb\} = \{ab, abb, abbb\}.$$

A diferència del que passa amb el monoide $\langle \Sigma^*, \cdot \rangle$ format pels mots sobre un cert alfabet Σ, en el qual tots els elements són simplificables, el monoide format pels llenguatges, $\langle \mathcal{P}(\Sigma^*), \cdot \rangle$, no gaudeix d'aquesta propietat. Ni tant sols existeix simplificabilitat quan només exigim aquesta propietat per un dels costats. Diem que un llenguatge L és simplificable *per la dreta* quan satisfà la condició

$$\forall L_1, L_2 \quad L_1 \cdot L = L_2 \cdot L \implies L_1 = L_2.$$

(Simètricament, la condició de ser simplificable *per l'esquerra* s'escriu amb L concatenant per l'esquerra.) Vegem, a continuació, un exemple d'aquesta no-simplificabilitat.

Exemple 1.5 Podem trobar exemples de llenguatges no simplificables fins i tot restringint-nos al cas de llenguatges finits, com posa de manifest l'equació següent:

[4] Representem per $\mathcal{P}(C)$ el *conjunt de parts* d'un conjunt C, és a dir, el conjunt format per tots els *subconjunts* de C.

$$\{\lambda, a, a^2\} \cdot \{\lambda, a\} = \{\lambda, a^2\} \cdot \{\lambda, a\}.$$

És fàcil demostrar que una condició *suficient* perquè un llenguatge L sigui simplificable per la dreta és que contingui algun mot que no sigui sufix de cap altre mot de L ni tingui cap altre mot de L com a sufix. Deixem com a exercici demostrar que el llenguatge $\{a, b, ab, ba\}$ és simplificable (a dreta i a esquerra) tot i no satisfer aquesta condició.

Si w és un mot de Σ^*, escriurem generalment wL en lloc de $\{w\}L$, i Lw en lloc de $L\{w\}$. Les igualtats següents es dedueixen directament de la definició de concatenació:

$$L \cdot \varnothing = \varnothing \cdot L = \varnothing,$$
$$L \cdot \lambda = \lambda \cdot L = L.$$

Anàlogament a com fem per a la concatenació de mots, també aquí utilitzem la notació exponencial per representar la concatenació repetida d'un llenguatge amb si mateix. Així, per a un llenguatge L fem

$$L^0 = \{\lambda\} \quad \text{i} \quad \forall i \geqslant 0 \quad L^{i+1} = L^i L.$$

1.5 Altres operacions amb llenguatges

Els llenguatges, en tant que conjunts, admeten les operacions pròpies dels conjunts: *reunió, intersecció, complementació, diferència,* etc. Les operacions que definim a continuació són, tanmateix, específiques dels llenguatges com a conjunts de mots.

Tancament de Kleene

Anomenem *tancament de Kleene* (o també *estrella de Kleene*) d'un llenguatge L, i el representem per L^*, el llenguatge següent

$$L^* = \bigcup_{n \geqslant 0} L^n.$$

Es tracta del monoide generat per L, és a dir, del llenguatge format per λ i totes les concatenacions possibles de mots de L. És, doncs, el mínim monoide que conté L.

La notació utilitzada per al tancament de Kleene és coherent amb el fet d'haver representat per Σ^* el conjunt de tots els mots sobre un alfabet Σ. En efecte, Σ^* és el tancament de Kleene de Σ, considerat aquí com un llenguatge els mots del qual (de longitud 1) són els símbols de l'alfabet. En aquest cas, es té

$$(\Sigma)^* = \bigcup_{n \geqslant 0} \Sigma^n = \bigcup_{n \geqslant 0} \{w \in \Sigma^* \mid |w| = n\} = \Sigma^*.$$

També resulta útil introduir la noció de *tancament positiu de Kleene* d'un llenguatge L, representat per L^+, que és

$$L^+ = \bigcup_{n > 0} L^n.$$

Si a és un símbol d'un alfabet Σ, escriurem generalment a^* i a^+ en lloc de $\{a\}^*$ i $\{a\}^+$, respectivament.

Les propietats següents, que se satisfan per a tot llenguatge L i que són totes elles de demostració immediata a partir de les definicions anteriors, es deixen com a exercici.

$$\begin{array}{ll}
L^* = \{\lambda\} \cup L^+ & (L^*)^* = (L^+)^* = (L^*)^+ = L^* \\
L^+ = L^* \;\Leftrightarrow\; \lambda \in L \;\Leftrightarrow\; \lambda \in L^+ & (L^+)^+ = L^+ \\
L^*L = LL^* = L^+L^* = L^*L^+ = L^+ & \varnothing^* = \{\lambda\}, \qquad \varnothing^+ = \varnothing \\
L^*L^* = L^* \quad \text{però} \quad L^+L^+ = L^2L^* &
\end{array}$$

Exemple 1.6 La igualtat següent permet caracteritzar el conjunt de mots d'un alfabet de dos símbols. Es tracta de

$$\{a\,,b\}^* = (a^*b)^*a^*.$$

Només cal demostrar la inclusió d'esquerra a dreta, ja que la de sentit contrari és òbvia, pel fet que $\{a\,,b\}^*$ és el conjunt de *tots* els mots sobre l'alfabet $\{a\,,b\}$. Ara, tot mot de $\{a\,,b\}^*$ pot ser presentat agrupant cada b que hi intervé amb les a's (zero o més) que la precedeixen. Cada un d'aquests grups és un mot del llenguatge a^*b, i n'hi poden haver zero o més. Per tant, tots els grups plegats formen un mot de $(a^*b)^*$. Cal afegir-hi les a's que eventualment hi pugui haver al final del mot i que no vagin seguides de cap b, que formen part de a^*.

Un raonament anàleg a l'anterior permetria demostrar que també les expressions

$$a^*(ba^*)^*, \quad b^*(ab^*)^* \quad \text{i} \quad (b^*a)^*b^*$$

representen el conjunt $\{a\,,b\}^*$.

Revessament

De manera anàloga a com hem estès la concatenació entre mots a una operació entre llenguatges, podem estendre als llenguatges l'operació de revessament definida sobre mots. Anomenem revessat d'un llenguatge L el llenguatge format pels revessats dels mots de L. Representem per L^{R} el revessat de L. És a dir,

$$L^{\mathsf{R}} = \{w^{\mathsf{R}} \mid w \in L\}.$$

1.6 Morfismes i substitucions[5]

Morfismes

L'estudi dels llenguatges formals inclou l'anàlisi de les característiques estructurals d'aquests llenguatges. Des d'un punt de vista estructural, el llenguatge

$$L_1 = \{(01)^n1(10)^n1 \mid n \geqslant 0\}$$

[5]El lector que no estigui familiaritzat amb la teoria de llenguatges formals pot saltar-se aquesta secció en una primera lectura.

és similar al llenguatge $L_1' = \{a^n b a^n \mid n \geqslant 0\}$, mentre que el llenguatge

$$L_2 = \{(01)^n 0 (10)^n 0 \ \mid \ n \geqslant 0\}$$

és similar al llenguatge $L_2' = \{aa\}^* b$. En els capítols que segueixen veurem que L_1' i L_2' pertanyen a categories estructurals diferents. Els conceptes que introduirem a continuació tenen per objecte transformar llenguatges aparentment complicats en altres de més senzills, tot conservant les característiques estructurals que els fan similars.

DEFINICIÓ. Considerem dos alfabets Σ_1 i Σ_2. Anomenem *morfisme* una aplicació de la forma

$$h\colon \Sigma_1^* \longrightarrow \Sigma_2^*$$

que satisfà la condició

$$\forall x, y \in \Sigma_1^* \ \ h(x \cdot y) = h(x) \cdot h(y).$$

Observeu que es tracta d'un *morfisme de monoides*, en el sentit algebraic del terme,[6] entre Σ_1^* i Σ_2^*. Per tractar-se de monoides amb tots els elements simplificables, la condició $h(\lambda) = \lambda$ es dedueix de la condició anterior. Aquests morfismes queden completament definits especificant els seus valors únicament sobre el conjunt finit Σ_1. En efecte, tot mot $w \in \Sigma_1^*$ que sigui diferent de λ té la forma $w = a_1 \ldots a_n$, amb $a_1, \ldots, a_n \in \Sigma_1$. Per tant, resulta

$$h(w) = h(a_1) \cdots h(a_n).$$

Exemple 1.7 Donats els alfabets $\Sigma_1 = \{a, b, c, d\}$ i $\Sigma_2 = \{0\,, 1\}$, podem definir un morfisme a partir dels valors $h(a) = 01$, $h(b) = 10$, $h(c) = 011$ i $h(d) = 110$. La imatge per aquest morfisme del mot *abbdd* és aleshores 011010110110. Aquesta imatge, per cert, coincideix amb la del mot *cacab*.

Com és habitual en la teoria de conjunts, podem considerar les dues extensions, directa i inversa, d'aquesta aplicació sobre mots als conjunts de llenguatges entesos com a parts de Σ_1^* i Σ_2^*. Com a extensió directa tenim

$$h\colon \mathcal{P}\left(\Sigma_1^*\right) \longrightarrow \mathcal{P}\left(\Sigma_2^*\right)$$
$$h(L) = \{y \in \Sigma_2^* \ \mid \ \exists x \in L \ \ y = h(x)\},$$

i com a extensió inversa (anomenada sovint *morfisme invers*),

$$h^{-1}\colon \mathcal{P}\left(\Sigma_2^*\right) \longrightarrow \mathcal{P}\left(\Sigma_1^*\right)$$
$$h^{-1}(L) = \{x \in \Sigma_1^* \ \mid \ h(x) \in L\}.$$

És fàcil comprovar que l'extensió directa manté el caràcter de morfisme entre els monoides $\mathcal{P}\left(\Sigma_1^*\right)$ i $\mathcal{P}\left(\Sigma_2^*\right)$. És a dir,

$$h(\{\lambda\}) = \{\lambda\} \qquad \text{i} \qquad h(L_1 \cdot L_2) = h(L_1) \cdot h(L_2).$$

[6] A la nota 3 hem donat la definició algebraica de morfisme.

En canvi, el morfisme invers no és, malgrat el seu nom, un morfisme en el sentit algebraic del terme. És a dir, pot ser $h^{-1}(\{\lambda\}) \neq \{\lambda\}$ o $h^{-1}(L_1 \cdot L_2) \neq h^{-1}(L_1) \cdot h^{-1}(L_2)$. Ho veurem més endavant, a l'exemple 1.10.

Exemple 1.8 Siguin $\Sigma_1 = \{a, b\}$ i $\Sigma_2 = \{0,1\}$, i considerem els dos morfismes de la forma $h_1, h_2 \colon \Sigma_1^* \longrightarrow \Sigma_2^*$ definits així:

$$\begin{cases} h_1(a) = 01 \\ h_1(b) = 11 \end{cases} \qquad \begin{cases} h_2(a) = 01 \\ h_2(b) = 00. \end{cases}$$

L'aplicació del morfisme invers h_1^{-1} al llenguatge

$$L_1 = \{(01)^n 1 (10)^n 1 \mid n \geqslant 0\}$$

dóna com a resultat

$$h_1^{-1}(L_1) = L_1' = \{a^n b a^n \mid n \geqslant 0\},$$

mentre que l'aplicació del morfisme directe h_2 al llenguatge $L_2' = \{aa\}^* b$ dóna

$$h_2(L_2') = \{0101\}^* 00 = L_2 = \{(01)^n 0 (10)^n 0 \mid n \geqslant 0\}.$$

Exemple 1.9 Sigui Σ_2 un alfabet qualsevol. Estem interessats en la transformació que, donat un llenguatge $L \subseteq \Sigma_2^*$, es queda només amb els mots que resulten d'esborrar els símbols que estan en posicions parelles dels mots de longitud parella de L. És a dir, que per a cada llenguatge $L \subseteq \Sigma_2^*$ volem obtenir un llenguatge L' tal que

$$L' = \{a_1 a_3 \ldots a_{2n-1} \mid a_1 a_2 a_3 \ldots a_{2n-1} a_{2n} \in L\}.$$

Per tal de concretar les idees, suposem que Σ_2 és un alfabet de tres símbols, $\Sigma_2 = \{a, b, c\}$. Construïm un alfabet de 9 símbols (en general tindrà $\|\Sigma_2\|^2$ símbols), que representem per $\Sigma_1 = \{1, 2, \ldots, 9\}$. Definim ara els dos morfismes de la taula 1.1.

Taula 1.1 *Dos morfismes h_1 i h_2*

Σ_1	$h_1(\Sigma_1)$	$h_2(\Sigma_1)$
1	aa	a
2	ab	a
3	ac	a
4	ba	b
5	bb	b
6	bc	b
7	ca	c
8	cb	c
9	cc	c

És fàcil constatar, per aplicació directa de les transformacions indicades, que

$$L' = h_2\left(h_1^{-1}(L)\right).$$

Propietats elementals dels morfismes

Comencem fent un repàs d'algunes propietats que tenen els morfismes com a meres aplicacions de la forma $h\colon \Sigma_1^* \longrightarrow \Sigma_2^*$. Les fórmules de la columna de l'esquerra fan referència a llenguatges $L \subseteq \Sigma_1^*$, mentre que les de la dreta es refereixen a llenguatges $L \subseteq \Sigma_2^*$.

$$\begin{array}{ll}
h(L) = \varnothing \iff L = \varnothing & h^{-1}(\varnothing) = \varnothing \\
L_1 \subseteq L_2 \implies h(L_1) \subseteq h(L_2) & L_1 \subseteq L_2 \implies h^{-1}(L_1) \subseteq h^{-1}(L_2) \\
h(L_1 \cup L_2) = h(L_1) \cup h(L_2) & h^{-1}(L_1 \cup L_2) = h^{-1}(L_1) \cup h^{-1}(L_2) \\
h(L_1 \cap L_2) \subseteq h(L_1) \cap h(L_2) & h^{-1}(L_1 \cap L_2) = h^{-1}(L_1) \cap h^{-1}(L_2) \\
L \subseteq h^{-1}(h(L)) & h\left(h^{-1}(L)\right) \subseteq L \\
 & h^{-1}\left(\overline{L}\right) = \overline{h^{-1}(L)}.
\end{array}$$

A aquestes propietats afegim les que es deriven de la condició de morfisme. Acabem de veure que

$$h(L_1 \cdot L_2) = h(L_1) \cdot h(L_2),$$

que s'estén immediatament a qualsevol concatenació finita,

$$\forall n \in \mathbb{N} \quad h(L^n) = (h(L))^n .$$

Destaquem, a més, la propietat següent.

PROPOSICIÓ. $h(L^*) = (h(L))^*$.

DEMOSTRACIÓ.

$$h(L^*) = h\left(\bigcup_{n\geqslant 0} L^n\right) = \bigcup_{n\geqslant 0} h(L^n) = \bigcup_{n\geqslant 0} (h(L))^n = (h(L))^* .$$

La primera igualtat i l'última s'obtenen de la definició de l'operació *estrella*. La segona és una propietat conjuntista. La tercera, finalment, requereix que h sigui morfisme. □

En canvi, l'aplicació dels morfismes inversos a la concatenació o a l'estrella de llenguatges no dóna lloc a igualtats. Sols obtenim inclusions en un dels sentits. Es tracta de les següents, que són fàcils de verificar:

$$h^{-1}(L_1 \cdot L_2) \supseteq h^{-1}(L_1) \cdot h^{-1}(L_2)$$
$$h^{-1}(L^*) \supseteq \left(h^{-1}(L)\right)^* .$$

L'exemple següent il·lustra la inexistència de la inclusió contrària.

Exemple 1.10 Considerem el morfisme definit per $h(a) = 01$ i $h(b) = 10$. Considerem dos conjunts ben simples, el $L_1 = \{0\}$ i el $L_2 = \{1\}$. Tenim

$$h^{-1}(L_1 \cdot L_2) = h^{-1}(\{01\}) = \{a\},$$

en canvi,

$$h^{-1}(L_1) = h^{-1}(L_2) = \varnothing.$$

Semblantment, si considerem el conjunt $L = \{0, 1\}$, tenim que

$$h^{-1}(L^*) = h^{-1}(\{0,1\}^*) = \{a\,,b\}^*,$$

mentre que

$$\left(h^{-1}(L)\right)^* = \varnothing^* = \{\lambda\}.$$

Substitucions

A l'exemple 1.6 hem vist com demostrar una igualtat senzilla entre llenguatges. Es tractava de

$$(a^*b)^*a^* = \{a\,,b\}^*.$$

Però si la igualtat considerada és la següent, que fa referència a llenguatges L_1 i L_2 qualssevol,

$$(L_1^*L_2)^*L_1^* = (L_1 \cup L_2)^*,$$

la demostració esdevé formalment molt més llarga. La utilització de les substitucions, que estudiem a continuació, ens permetrà reduir la segona igualtat a la primera.

DEFINICIÓ. Siguin Σ_1 i Σ_2 dos alfabets qualssevol. Una *substitució* és una aplicació de la forma

$$\sigma\colon \Sigma_1^* \longrightarrow \mathcal{P}\left(\Sigma_2^*\right),$$

que és morfisme de monoides. És a dir, que si x i y són *mots* de Σ_1^*, i $\sigma(x)$, $\sigma(y)$ i $\sigma(xy)$ són els *llenguatges* de Σ_2^* corresponents a x, y i xy, respectivament, es té

$$\sigma(\lambda) = \{\lambda\}$$
$$\sigma(x)\cdot\sigma(y) = \sigma(xy).$$

Remarquem que ara $\sigma(x)$, $\sigma(y)$ i $\sigma(xy)$ són llenguatges i no mots com en el cas dels morfismes. De nou tenim aquí que, pel fet de ser Σ_1^* finitament generat a partir de Σ_1, aquesta aplicació queda completament definida donant els valors de σ sobre el conjunt finit de símbols de Σ_1. És a dir, si Σ_1 és $\{a_1,\ldots,a_n\}$, una substitució σ queda caracteritzada per n llenguatges $L_{a_1},\ldots,L_{a_n}$.

Exemple 1.11 Els *arbres unari-binaris* (també anomenats *arbres de Motzkin*, vegeu la secció 5.14 de [SeF96]) són arbres ordenats, els nodes dels quals tenen arietat menor o igual que dos. Com a exemple de substitució, considerem la que permet obtenir els arbres unari-binaris a partir dels *arbres binaris*, mitjançant la substitució dels vèrtexs d'aquests últims per *cadenes unàries* de vèrtexs.

Cada arbre binari pot ser codificat amb un mot de l'alfabet $\{0\,,2\}$, els símbols del qual representen els nodes d'arietat zero i dos, respectivament. La codificació de l'arbre s'obté escrivint consecutivament els símbols dels seus vèrtexs recorreguts en *preordre*. A la figura 1.2 donem un arbre binari i la seva codificació.

La substitució que permet intercalar cadenes unàries en el lloc dels vèrtexs és la següent:

$$\sigma\colon \{0\,,2\}^* \longrightarrow \mathcal{P}\left(\{0\,,1\,,2\}^*\right),$$

amb els valors

$$\sigma(0) = L_0 = 1^*0 \quad \text{i} \quad \sigma(2) = L_2 = 1^*2.$$

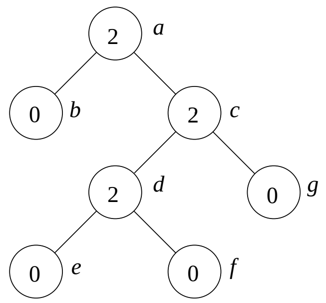

Fig. 1.2 *L'arbre binari de codificació* 2022000

En el cas de l'arbre de la figura 1.2, el conjunt d'arbres unari-binaris que es poden obtenir a partir d'ell és el donat per l'expressió següent:

$$\sigma(2022000) = 1^*21^*01^*21^*21^*01^*01^*0.$$

Un dels arbres d'aquest conjunt, obtingut a partir de substituir els vèrtexs b, d i g per cadenes unàries de longituds 1, 2 i 1, respectivament, s'ha dibuixat a la figura 1.3.

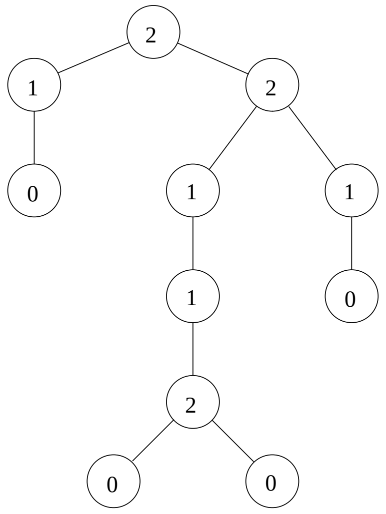

Fig. 1.3 *L'arbre unari-binari de codificació* 21021120010

Les substitucions, que han estat definides sobre mots d'un cert alfabet Σ_1, poden estendre's als llenguatges escrits sobre aquest mateix alfabet de la manera següent:

$$\forall L \subseteq \Sigma_1^* \qquad \sigma(L) = \bigcup_{x \in L} \sigma(x) = \{y \in \Sigma_2^* \mid \exists x \in L \; y \in \sigma(x)\}.$$

Remarquem que es tracta d'una extensió de tipus diferent a les vistes fins aquí. (La condició s'escriu $y \in \sigma(x)$, no $y = \sigma(x)$.) Tot i això, també ara és cert que *aquesta extensió és un morfisme del monoide* $\mathcal{P}(\Sigma_1^*)$ *en el monoide* $\mathcal{P}(\Sigma_2^*)$. En efecte, per una part tenim

$$\sigma(\{\lambda\}) = \sigma(\lambda) = \{\lambda\},$$

i per una altra,

PROPOSICIÓ. $\sigma(L_1 \cdot L_2) = \sigma(L_1) \cdot \sigma(L_2)$.

DEMOSTRACIÓ. En efecte, $\forall y \in \Sigma_2^*$

$$\begin{aligned}
y \in \sigma(L_1 \cdot L_2) &\iff \exists x \in L_1 L_2 \quad y \in \sigma(x) \\
&\iff \exists x_1 \in L_1\ \exists x_2 \in L_2 \quad y \in \sigma(x_1 \cdot x_2) \\
&\iff \exists x_1 \in L_1\ \exists x_2 \in L_2 \quad y \in \sigma(x_1) \cdot \sigma(x_2) \\
&\iff \exists x_1 \in L_1\ \exists x_2 \in L_2\ \exists y_1 \in \sigma(x_1)\ \exists y_2 \in \sigma(x_2) \quad y = y_1 y_2 \\
&\iff \exists y_1 \in \sigma(L_1)\ \exists y_2 \in \sigma(L_2) \quad y = y_1 y_2 \\
&\iff y \in \sigma(L_1) \cdot \sigma(L_2),
\end{aligned}$$

on la tercera equivalència es basa en el fet que σ és una substitució. □

Exemple 1.12 Podem reprendre la substitució de l'exemple 1.11 i estendre-la al llenguatge format per les codificacions de tots els arbres binaris. De la definició recursiva dels arbres binaris (vegeu la secció 2.3 de [Knu68]) obtenim directament l'expressió següent del llenguatge associat:

$$L_{\mathcal{B}} = \{0\} \cup \{w \in \{0\,,2\}^* \mid \exists y,z \in L_{\mathcal{B}} \quad w = 2yz\}\,.$$

En aplicar la substitució definida per

$$\sigma(0) = 1^*0 \qquad \sigma(2) = 1^*2$$

obtenim la següent definició recursiva dels arbres de Motzkin:

$$L_{\mathcal{M}} = \sigma(L_{\mathcal{B}}) = 1^*0\ \cup \{w \in \{0,1,2\}^* \mid \exists x \in 1^*2\ \ \exists y,z \in L_{\mathcal{M}} \quad w = xyz\}\,,$$

que és equivalent a la seva definició com a arbres unari-binaris,

$$L_{\mathcal{M}} = \{0\} \cup \{1x \mid x \in L_{\mathcal{M}}\} \cup \{2yz \mid y\,,z \in L_{\mathcal{M}}\}.$$

Altres propietats elementals de les substitucions

Les propietats següents es demostren, de manera similar a la que acabem d'analitzar, simplement amb l'aplicació de les definicions donades.

PROPOSICIÓ. *La substitució de la reunió o de l'estrella de llenguatges és la reunió o l'estrella de les substitucions dels llenguatges considerats.*

Formalment,

$$\begin{aligned}
\sigma(L_1 \cup L_2) &= \sigma(L_1) \cup \sigma(L_2) \\
\sigma(L_1^*) &= (\sigma(L_1))^*\,.
\end{aligned}$$

Observem que els morfismes poden assimilar-se a un cas particular de les substitucions. Es tracta d'identificar l'aplicació $h(x) = y$ amb l'aplicació $h(x) = \{y\}$, és a dir, considerar el mot imatge com el conjunt reduït a aquest mot. Feta aquesta assimilació, les propietats relatives a la imatge per morfisme de la reunió, la concatenació i l'estrella de llenguatges s'haurien pogut deduir de les propietats de les substitucions que acabem d'establir.

Exemple 1.13 Podem ara fer la demostració de l'equació amb què hem introduït el tema de les substitucions. Per tal de demostrar que per a tot L_1 i L_2, es té

$$(L_1^* L_2)^* L_1^* = (L_1 \cup L_2)^*,$$

només cal considerar un alfabet de dos símbols $\Sigma = \{a, b\}$ i definir la substitució $\sigma(a) = L_1$, $\sigma(b) = L_2$. D'aquesta manera, el primer membre és la imatge del llenguatge $(a^*b)^*a^*$, mentre que el segon membre és la imatge del llenguatge $\{a, b\}^*$. L'aplicació de la igualtat de l'exemple 1.6 clou la demostració. En efecte,

$$\begin{aligned}(a^*b)^*\, a^* = \{a, b\}^* &\Longrightarrow \sigma\left((a^*b)^*\, a^*\right) = \sigma\left(\{a, b\}^*\right)\\ &\Longrightarrow \left(L_1^* L_2\right)^* L_1^* = \left(L_1 \cup L_2\right)^*.\end{aligned}$$

Convé advertir, tanmateix, que aquesta traducció entre identitats expressades amb símbols i identitats referides a llenguatges només està justificada quan les úniques operacions que intervenen en les expressions són la reunió, la concatenació i l'estrella. No podem fer el mateix quan hi intervenen altres operacions com la intersecció, la complementació o el revessament. Així, per exemple, si a és un símbol d'un alfabet i L és un llenguatge qualsevol, tenim que $a = a^{\mathsf{R}}$; en canvi, en general no és cert que $L = L^{\mathsf{R}}$.

Exercicis

1.1 Demostreu que no existeix cap mot $w \in \{a, b\}^*$ tal que $aw = wb$.

1.2 Si $w \in \{a, b\}^*$ i $abw = wab$, demostreu que $w = (ab)^n$ per a algun nombre $n \geqslant 0$.

1.3 Es diu que un mot z és *arrel* d'un mot w quan es té que $w = z^n$ per a algun nombre $n \geqslant 0$. Demostreu que la concatenació de dos mots és commutativa si i sols si els dos mots tenen alguna arrel en comú. Expressat formalment,

$$xy = yx \iff \exists z\ \exists i, j \geqslant 0\ x = z^i \wedge y = z^j.$$

1.4 Considereu la següent definició recursiva del revessament de mots:

$$1.\ \lambda^{\mathsf{R}} = \lambda \qquad 2.\ (aw)^{\mathsf{R}} = w^{\mathsf{R}}a\,,\ \text{on } a \in \Sigma \text{ i } w \in \Sigma^*.$$

Utilitzeu la definició precedent per demostrar les propietats següents:

1. $\forall x, y \in \Sigma^*\ \ (xy)^{\mathsf{R}} = y^{\mathsf{R}} x^{\mathsf{R}}$.
2. $\forall L_1, L_2 \subseteq \Sigma^*\ \ (L_1 L_2)^{\mathsf{R}} = L_2^{\mathsf{R}} L_1^{\mathsf{R}}$.

1.5 Donats tres llenguatges qualssevol, A, B i C, es tracta d'estudiar la distributivitat, a dreta i a esquerra, de la concatenació respecte de la reunió i la intersecció.

1. Demostreu la distributivitat respecte de la reunió

$$\begin{aligned}(A \cup B)C &= AC \cup BC\\ A(B \cup C) &= AB \cup AC.\end{aligned}$$

2. Demostreu les inclusions

$$\begin{aligned}(A \cap B)C &\subseteq AC \cap BC\\ A(B \cap C) &\subseteq AB \cap AC.\end{aligned}$$

3. Trobeu contraexemples per a les inclusions recíproques.

1.6 Demostreu que una condició suficient perquè un llenguatge L sigui simplificable per la dreta és que contingui algun mot que ni sigui sufix de cap altre mot de L, ni tingui cap altre mot de L com a sufix. Escrita formalment, la condició és:

$$\exists w \in L \; \forall x \in L \; \forall y \neq \lambda \quad x \neq yw \wedge w \neq yx.$$

Per tal de constatar que es tracta d'una condició no necessària, demostreu que el llenguatge $\{a, b, ab, ba\}$, que no la satisfà, és simplificable.

1.7 Demostreu que les relacions següents se satisfan per a tot parell de llenguatges L_1 i L_2.

1. $(L_1 \cup L_2)^* = (L_1^* L_2^*)^*$.
2. $L_1(L_2 L_1)^* = (L_1 L_2)^* L_1$.
3. $L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*$.
4. $(L_1 \cap L_2)^* \subseteq L_1^* \cap L_2^*$.

Trobeu contraexemples per a les inclusions recíproques dels dos últims apartats.

1.8 Els enunciats d'aquest exercici fan referència a la relació entre un llenguatge qualsevol i el seu *quadrat*. Demostreu primer que els següents se satisfan per a tot llenguatge L.

1. $\left(L^2\right)^* \subseteq \left(L^*\right)^2$.
2. $\lambda \in L \implies L \subseteq L^2$.
3. $L \subseteq L^2 \iff \lambda \in L \;\vee\; L = \varnothing$.
4. $L^2 \subseteq L \iff L^+ = L$.
5. $\lambda \in L \;\wedge\; L^2 \subseteq L \iff L^* = L$.
6. $L = L^2 \implies L = L^* \vee L = \varnothing$.

Trobeu ara contraexemples de llenguatges L que no satisfan els enunciats següents.

7. $\left(L^*\right)^2 \subseteq \left(L^2\right)^*$.
8. $L^2 \subseteq L \implies L \subseteq L^2$.
9. $L \subseteq L^2 \implies L^2 \subseteq L$.
10. $L^2 \subseteq L \implies L^* \subseteq L$.
11. $L \subseteq L^2 \implies L^* \subseteq L$.

1.9 Demostreu que tot llenguatge L satisfà la igualtat $\left(\overline{L}\right)^{\mathrm{R}} = \overline{\left(L^{\mathrm{R}}\right)}$.

1.10 Doneu contraexemples que posin de manifest la falsedat de cada un dels enunciats següents:

1. $LL^{\mathrm{R}} \cup L^{\mathrm{R}}L = \Sigma^*$.
2. $L\overline{L} \cup \overline{L}L = \Sigma^*$.
3. $\lambda \notin L \Rightarrow \left(L\overline{L}\right)^* = L^*$.
4. $\lambda \in L \Rightarrow \left(LL^{\mathrm{R}}\right)^* = L^*$.

1.11

1. Trobeu un llenguatge $L \subseteq \{a\,, b\}^*$ que contingui $\{a\,, b\}$ i que satisfaci l'equació $\overline{L} = L \cdot L \cup \{\lambda\}$.
2. Demostreu que, en canvi, no hi ha cap llenguatge $L \subseteq \{a\,, b\}^*$ que contingui els mots a i b i que satisfaci l'equació $L = \overline{L} \cdot \overline{L} \cup \{\lambda\}$.
3. Demostreu que cap llenguatge L no satisfà l'equació $L = L \cdot \overline{L} \cup \{\lambda\}$.

1.12 Demostreu que tot llenguatge L satisfà la inclusió $\overline{\left((\overline{L})^*\right)} \subseteq L$. Trobeu un contraexemple per a la inclusió recíproca.

1.13 Demostreu que l'equivalència següent se satisfà per a qualsevol llenguatge L.

$$\left(L\overline{L} = L \;\wedge\; L^+ = L\right) \iff \left(L\Sigma^* = L \;\wedge\; \lambda \notin L\right).$$

1.14 Sigui $\Sigma = \{a\,,b\}$, sigui $L \subsetneq \Sigma^*$ i sigui $L' = \left(\left(\left(L^*\right)^c\right)^*\right)^c$ (on L^c representa el complementari de L). Demostreu amb contraexemples (altres, però, que $L=\Sigma^*$) que cap de les dues inclusions entre L i L' no pot ser establerta amb caràcter general.

1.15 Donats dos morfismes $f\colon \Sigma_1^* \longrightarrow \Sigma_2^*$ i $g\colon \Sigma_1^* \longrightarrow \Sigma_2^*$, diem que són iguals si $f(x) = g(x)$ per a tot x de Σ_1^*. Com demostraríeu que dos morfismes són iguals? Aquest és un exemple de problema de decisió. Existeix algun algorisme de decisió que resolgui aquest problema?

1.16 Sabem que tota substitució de la forma $\sigma\colon \Sigma_1^* \longrightarrow \mathcal{P}\left(\Sigma_2^*\right)$ s'estén a una aplicació de la forma $\sigma\colon \mathcal{P}\left(\Sigma_1^*\right) \longrightarrow \mathcal{P}\left(\Sigma_2^*\right)$ que també és morfisme de monoides. Demostreu, però, que no tot morfisme d'aquest tipus és una substitució.

1.17 Sigui $h\colon \Sigma_1^* \longrightarrow \Sigma_2^*$ un morfisme qualsevol.

1. Doneu contraexemples que posin de manifest que no pot establir-se amb caràcter general cap de les dues inclusions entre $h\left(\overline{L}\right)$ i $\overline{h(L)}$, on $L \subseteq \Sigma_1^*$.
2. Demostreu que els enunciats següents són equivalents:
   1. h és injectiu.
   2. $\forall L \subseteq \Sigma_1^* \quad h\left(\overline{L}\right) \subseteq \overline{h(L)}$.
   3. $\forall L \subseteq \Sigma_1^* \quad h^{-1}\left(h(L)\right) = L$.
   4. $\forall L_1, L_2 \subseteq \Sigma_1^* \quad h(L_1 \cap L_2) = h(L_1) \cap h(L_2)$.
3. Demostreu que els enunciats següents són equivalents:
   1. h és exhaustiu.
   2. $\forall L \subseteq \Sigma_1^* \quad h\left(\overline{L}\right) \supseteq \overline{h(L)}$.
   3. $\forall L \subseteq \Sigma_2^* \quad h\left(h^{-1}(L)\right) = L$.
   4. $\forall L \subseteq \Sigma_2^* \quad h^{-1}(L) = \varnothing \implies L = \varnothing$.

1.18 Donat un monoide M qualsevol, s'anomenen *factors primers* de M, i el seu conjunt es representa per $\mathsf{primers}(M)$, els elements de M —tret de l'element neutre— que no poden descompondre's en producte de dos factors que siguin tots dos diferents de l'element neutre (que representem en general per λ llevat que existeixi un símbol específic per al monoide considerat). Formalment tenim

$$\mathsf{primers}(M) = (M - \{\lambda\}) - (M - \{\lambda\})^2.$$

1. Sigui Σ un alfabet qualsevol i sigui $x \in \Sigma^+$ un mot qualsevol. Demostreu que el conjunt
   $$M = \{\lambda, x^3, x^5, x^6\} \cup x^8x^*$$
   és un monoide, i que $\mathsf{primers}(M) = \{x^3, x^5\}$.
2. Sigui $M = (ba^*)^*$. Demostreu que $\mathsf{primers}(M) = ba^*$.
3. Sigui $M = \langle \mathbb{R}, \cdot \rangle$ el monoide multiplicatiu dels reals. Demostreu que $\mathsf{primers}(M) = \varnothing$.
4. Sigui $M = \langle \mathbb{N}, + \rangle$ el monoide additiu dels naturals. Demostreu que $\mathsf{primers}(M) = \{1\}$.

1.19 S'anomena *monoide lliure* aquell en què tot element diferent del neutre admet una única descomposició en factors primers.

1. Demostreu que els monoides dels apartats 2 i 4 de l'exercici anterior són lliures, i que, en canvi, els dels apartats 1 i 3 no ho són.
2. Demostreu que el monoide $\{ab, ba, a\}^*$ no és lliure.

1.20 Sigui Σ un alfabet. Una part $C \subseteq \Sigma^*$ s'anomena *codi* si C genera lliurement C^*, és a dir, si tot mot de C^+ factoritza de manera única en mots de C.

1. Demostreu que ba^* i $\{aa, ba, baa\}$ són codis, però que, en canvi, $\{ab, ba, a\}$ no ho és.
2. Demostreu que si C és un codi, aleshores $\lambda \notin C$, i que $\forall m, n \;\; m \neq n \implies C^m \cap C^n = \varnothing$.
3. Demostreu que un conjunt C és un codi si i solament si C^* és lliure i $\mathsf{primers}(C^*) = C$.

1.21 Sigui Σ un alfabet i sigui $M \subseteq \Sigma^*$. Demostreu que M és un monoide lliure si i solament si el seu conjunt de factors primers és un codi.

1.22 Donats dos llenguatges qualssevol, A i B, s'anomena *residu dret*[7] de B per A el llenguatge

$$A \setminus B = \{y \mid \forall x \in A \ xy \in B\}.$$

Simètricament, s'anomena *residu esquerre* de B per A el llenguatge

$$B \,/\, A = \{x \mid \forall y \in A \ xy \in B\}.$$

Demostreu que els tres enunciats següents se satisfan per a llenguatges A i B qualssevol i trobeu contraexemples per a les dues inclusions recíproques

1. $(B \,/\, A)^{\mathsf{R}} = A^{\mathsf{R}} \setminus B^{\mathsf{R}}$.
2. $A \cdot (A \setminus B) \subseteq B$.
3. $B \subseteq A \setminus (A \cdot B)$.

[7] Trobareu una descripcio d'aquest concepte a [BeM97]

Capítol 2 Gramàtiques incontextuals

2.1 Introducció

La definició de les gramàtiques formals com a *sistemes generatius* és deguda a Noam Chomsky, que les va introduir com a eines per modelitzar l'estructura gramatical de les llengües. Un dels models definits per Chomsky va ser el de les *gramàtiques incontextuals*, anomenades en anglès *context-free grammars*, a les quals ens referirem sovint per mitjà de les sigles CFG. El seu ús en informàtica va ser motivat per la necessitat de construir compiladors eficients per a llenguatges de programació, i en aquest àmbit la seva aparició va lligada a la del llenguatge ALGOL, fruit del treball de Backus i Naur.

És freqüent que els textos de gramàtica de cursos secundaris incloguin *regles de reescriptura* de la forma:[1]

$$
\begin{aligned}
&\langle\texttt{oració}\rangle \rightarrow \langle\texttt{sintagma nominal}\rangle + \langle\texttt{sintagma verbal}\rangle\ [+\langle\texttt{sintagma adverbial}\rangle]\\
&\langle\texttt{sintagma nominal}\rangle \rightarrow \langle\texttt{nombre}\rangle + \langle\texttt{grup nominal}\rangle\\
&\langle\texttt{grup nominal}\rangle \rightarrow [\langle\texttt{determinador}\rangle+]\ \langle\texttt{nom}\rangle\ [+\langle\texttt{complement nom}\rangle]\\
&\langle\texttt{complement nom}\rangle \rightarrow \langle\texttt{adjectiu}\rangle \mid \langle\texttt{partícula subordinant}\rangle+\\
&\qquad\qquad\qquad\quad (\langle\texttt{sintagma nominal}\rangle|\langle\texttt{oració subordinada}\rangle)\\
&\langle\texttt{sintagma verbal}\rangle \rightarrow \langle\texttt{verb}\rangle\ [+\langle\texttt{atribut}\rangle]\ [+\langle\texttt{complement verbal}\rangle]\\
&\langle\texttt{verb}\rangle \rightarrow \langle\texttt{morfema verbal}\rangle + \langle\texttt{lexema verbal}\rangle\\
&\langle\texttt{complement verbal}\rangle \rightarrow \langle\texttt{sintagma nominal}\rangle \mid \langle\texttt{partícula subordinant}\rangle+\\
&\qquad\qquad\qquad\qquad\quad (\langle\texttt{sintagma nominal}\rangle|\langle\texttt{oració subordinada}\rangle) \mid\\
&\qquad\qquad\qquad\qquad\quad \langle\texttt{sintagma adverbial}\rangle
\end{aligned}
$$

que, juntament amb d'altres, expressen la forma en què és possible generar o analitzar enunciats d'un procés comunicatiu.

També és fàcil trobar en alguns manuals de llenguatges de programació la definició sintàctica d'aquests llenguatges en EBNF (forma estesa de Backus Naur). Així, a l'apèndix 1 de Wirth [Wir88], referent al llenguatge Modula-2 , apareixen entre d'altres les especificacions següents:

[1] Extret del Cap. 21 de BADIA, J. i GRIFOLL, J.: *JONC. Llengua catalana. 1r. de BUP*. Barcelona, Edicions 62, 1986.

$$
\begin{aligned}
\langle\texttt{declaració constant}\rangle &\to \langle\texttt{identificador}\rangle\text{"="}\langle\texttt{expressió constant}\rangle\\
\langle\texttt{identificador}\rangle &\to \langle\texttt{lletra}\rangle\ \{\langle\texttt{lletra}\rangle \mid \langle\texttt{digit}\rangle\}\\
\langle\texttt{expressió constant}\rangle &\to \langle\texttt{expressió constant simple}\rangle\\
&\quad [\langle\texttt{relació}\rangle\langle\texttt{expressió constant simple}\rangle]\\
\langle\texttt{relació}\rangle &\to \text{"="} \mid \text{"\#"} \mid \text{"<>"} \mid \text{">"} \mid \text{"<"} \mid \text{">="} \mid \text{"<="} \mid \mathsf{IN}\\
\langle\texttt{expressió constant simple}\rangle &\to [\text{"+"} \mid \text{"-"}]\langle\texttt{terme constant}\rangle\\
&\quad \{\langle\texttt{operador additiu}\rangle\langle\texttt{terme constant}\rangle\}\\
\langle\texttt{operador additiu}\rangle &\to \text{"+"} \mid \text{"-"} \mid \mathsf{OR},
\end{aligned}
$$

etcètera.

En els dos casos es tracta de definicions generatives que corresponen a exemples concrets de gramàtiques incontextuals.

2.2 Definició

Una *gramàtica incontextual* és una estructura de la forma

$$G = \langle V, \Sigma, P, S\rangle$$

els components de la qual es defineixen a continuació:

- V és un alfabet, els símbols del qual s'anomenen *variables*.
- Σ és un altre alfabet, disjunt de l'anterior, els símbols del qual s'anomenen *terminals*.
- P és el *conjunt de produccions* que definirem tot seguit.
- $S \in V$ s'anomena variable *inicial* o *de sortida*.

El conjunt de produccions és un subconjunt finit de parells de $V \times (V \cup \Sigma)^*$, que escriurem de la forma $A \to \alpha$, on $A \in V$ i $\alpha \in (V \cup \Sigma)^*$.

Se sol abreujar l'escriptura de les produccions agrupant en una mateixa línia totes les que comparteixen la mateixa variable al costat esquerre, de manera que aquesta variable s'escriu una única vegada, seguida de la fletxa. A continuació s'escriuen tots els mots de $(V \cup \Sigma)^*$ que apareixen a la dreta de les produccions considerades, separats entre si per una barra vertical.

Exemple 2.1 Especificació d'una gramàtica $G = \langle V, \Sigma, P, S\rangle$. El conjunt de variables és $V = \{S, T, X, Y, Z\}$. El conjunt de terminals és

$$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, \times, (,)\}.$$

Les produccions del conjunt P són

$$
\begin{array}{ll}
S \to X \mid X + S & Z \to Y \mid Y \times Z\\
X \to T \mid Y \times Z & T \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\\
Y \to T \mid (X + S)\,. &
\end{array}
$$

Conveni de notació

Per tal de facilitar la comprensió de les formulacions i d'agilitar-ne la descripció, ens atindrem sempre que sigui possible als criteris de notació següents:

a) Utilitzarem majúscules llatines $(A, B, C, \ldots Z)$ per representar les variables.
b) Utilitzarem minúscules llatines inicials $(a, b, c, \ldots)$ per representar els terminals.
c) Utilitzarem minúscules llatines finals $(\ldots w, x, y, z)$ per representar els mots de terminals.
d) Utilitzarem minúscules gregues $(\alpha, \beta, \gamma, \ldots, \omega)$ per representar els mots sobre $(V \cup \Sigma)^*$, que anomenarem *cadenes intermèdies*.

Exemple 2.2 Quan ens atenim al conveni de notació que acabem d'esmentar, i si reservem el símbol S per a la variable de sortida, podem descriure una gramàtica sense fer res més que especificar-ne el conjunt de produccions. Així,

$$\begin{aligned} &S \to aBS \mid bAS \mid \lambda \\ &A \to bAA \mid a \\ &B \to aBB \mid b\,. \end{aligned}$$

Derivació

Diem que entre dues cadenes intermèdies $\alpha, \beta \in (V \cup \Sigma)^*$ d'una gramàtica donada $G = \langle V, \Sigma, P, S\rangle$ hi ha una *derivació directa*, o que β *deriva directament* de α, quan podem descompondre α i β en la forma $\alpha = \gamma A \delta$ i $\beta = \gamma\varepsilon\delta$, on γ, δ i ε són també mots de $(V \cup \Sigma)^*$, A és una variable i $A \to \varepsilon$ és una producció de P. Ho escriurem de la forma $\alpha \underset{G}{\Rightarrow} \beta$.

Diem que una cadena intermèdia β *deriva* d'una altra α quan és possible passar de α a β per una seqüència finita (eventualment nul·la) de derivacions directes. És a dir, hi ha un nombre $n \geqslant 0$ i hi ha $n+1$ cadenes intermèdies α_0, α_1, … α_n tals que

$$\alpha = \alpha_0 \underset{G}{\Rightarrow} \alpha_1 \underset{G}{\Rightarrow} \alpha_2 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} \alpha_n = \beta.$$

Ho escriurem de la forma $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$. Remarquem que, com a relacions binàries definides sobre $(V \cup \Sigma)^*$, la derivació és el *tancament reflexiu i transitiu* de la derivació directa. Utilitzarem la notació $\alpha \overset{n}{\underset{G}{\Rightarrow}} \beta$ per representar el fet que la cadena β deriva de α a través exactament de n derivacions directes. També representarem per $\alpha \overset{+}{\underset{G}{\Rightarrow}} \beta$ una derivació d'un o més passos. En tots els casos que acabem de definir, prescindirem del símbol que representa la gramàtica quan no hi hagi possibilitat de confusió.

Llenguatge generat. Llenguatges incontextuals

Donada una gramàtica $G = \langle V, \Sigma, P, S\rangle$, s'anomena *llenguatge generat* per aquesta gramàtica, i es representa per $\mathsf{L}(G)$, el conjunt de mots per als quals existeix una derivació que duu de la variable inicial al mot considerat. Formalment,

$$\mathsf{L}(G) = \{w \in \Sigma^* \ \mid \ S \overset{*}{\Rightarrow} w\}.$$

Exemple 2.3 La gramàtica $G = \langle\{S\}, \{a\,, b\}, \{S \to \ aSb \mid \lambda\}, S\rangle$ genera el llenguatge $L = \{a^n b^n \mid n \geqslant 0\}$. Observeu la caracterització recursiva dels mots d'aquest llenguatge que es dedueix de la seva estructura gramatical. Es té, successivament,

$$w_0 = \lambda \quad w_1 = ab = aw_0 b \quad w_2 = a^2b^2 = aw_1 b \ \ldots \ w_n = a^n b^n = aw_{n-1} b.$$

La derivació que correspon al mot a^4b^4 és la següent:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaabbbb.$$

Exemple 2.4 La gramàtica de l'exemple 2.1 genera totes les expressions aritmètiques en què només intervenen sumes i productes de dígits sense parèntesis redundants.

Exemple 2.5 La gramàtica de l'exemple 2.2 genera tots els mots sobre l'alfabet $\{a, b\}$ que tenen tantes a's com b's.

Representem per $\mathsf{L}_G(\alpha)$ (o simplement per $\mathsf{L}(\alpha)$, quan la gramàtica G es considera sobreentesa) el conjunt de mots terminals que poden ser derivats d'una cadena intermèdia α. Formalment, $\mathsf{L}_G(\alpha) = \{w \in \Sigma^* \mid \alpha \underset{G}{\overset{*}{\Rightarrow}} w\}$. En particular, $\mathsf{L}_G(S) = \mathsf{L}(G)$.

DEFINICIÓ. Un llenguatge s'anomena *incontextual* quan existeix una CFG que el genera. Usarem les sigles CFL com a abreviació de 'llenguatge incontextual'. Representarem per CFL la família dels CFL.

Diem que dues gramàtiques són *equivalents* quan generen el mateix llenguatge. Formalment, G_1 és equivalent a G_2 si $\mathsf{L}(G_1) = \mathsf{L}(G_2)$.

2.3 Arbre de derivació. Ambigüitat

Considerem una derivació qualsevol de la forma $A \overset{*}{\Rightarrow} \alpha$, on A i α són, respectivament, una variable i una cadena intermèdia; considerem la seqüència de derivacions directes que constitueixen la derivació considerada. És molt útil representar aquesta derivació mitjançant un arbre, de la manera següent:

1. L'arrel de l'arbre té per etiqueta la variable A.
2. Cada node intern de l'arbre correspon a alguna de les variables que són substituïdes en el procés de derivació. Així, si una derivació directa consisteix a aplicar la producció $X \to \beta$, les etiquetes dels fills del node que correspon a aquesta variable X, llegides d'esquerra a dreta, formen β.
3. El producte de l'arbre, és a dir, el mot format per les seves fulles d'esquerra a dreta, és α.

Exemple 2.6 La figura 2.1 mostra l'arbre d'una derivació del mot *aabbbbaa* corresponent a la gramàtica de l'exemple 2.2. Es tracta de la derivació

$$\begin{aligned} S &\Rightarrow aBS \Rightarrow aaBBS \Rightarrow aabBS \Rightarrow aabbS \Rightarrow aabbbAS \\ &\Rightarrow aabbbbAAS \Rightarrow aabbbbaAS \Rightarrow aabbbbaaS \Rightarrow aabbbbaa. \end{aligned}$$

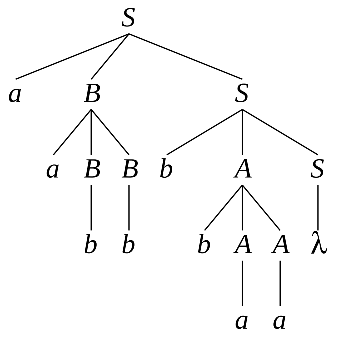

Fig. 2.1 *Un arbre de derivació*

En general, la correspondència entre arbres de derivació i derivacions no és unívoca en cap dels dos sentits. Una derivació donada pot correspondre a més d'un arbre. I a un mateix arbre de derivació li solen correspondre una pluralitat de derivacions.

Exemple 2.7 Considerem la gramàtica

$$\begin{aligned} S &\to AB \\ A &\to Aa \mid \lambda \\ B &\to aB \mid \lambda\,. \end{aligned}$$

La derivació

$$S \Rightarrow AB \Rightarrow AaB \Rightarrow aB \Rightarrow a$$

correspon als dos arbres de derivació de la figura 2.2.

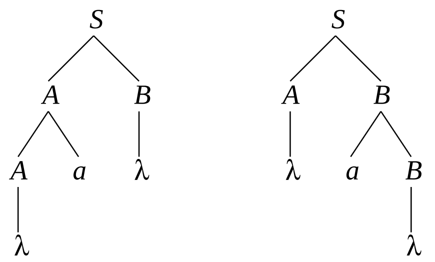

Fig. 2.2 *Dos arbres amb una mateixa derivació*

Exemple 2.8 Les dues derivacions següents també corresponen a l'arbre de l'exemple 2.6.

$$\begin{aligned}
1)\quad & S \Rightarrow aBS \Rightarrow aaBBS \Rightarrow aaBBbAs \Rightarrow aabBbAS \Rightarrow aabBbA \\
& \Rightarrow aabbbA \Rightarrow aabbbbAA \Rightarrow aabbbbaA \Rightarrow aabbbbaa \\
2)\quad & S \Rightarrow aBS \Rightarrow aBbAS \Rightarrow aBbA \Rightarrow aBbbAA \Rightarrow aBbbAa \\
& \Rightarrow aBbbaa \Rightarrow aaBBbbaa \Rightarrow aaBbbbaa \Rightarrow aabbbbaa\,.
\end{aligned}$$

És clar que les diferents derivacions que corresponen a un mateix arbre només es diferencien en l'ordre en què es van escrivint les derivacions directes que les componen. Des del punt de vista de la sintaxi gramatical, totes les derivacions que corresponen a un mateix arbre són equivalents.

DEFINICIÓ. Diem que una gramàtica és *ambigua* quan és possible construir dos arbres de derivació diferents que corresponguin a un mateix mot. Altrament, diem que la gramàtica és *inambigua*.

Exemple 2.9 La gramàtica $S \to\ aSbS \mid bSaS \mid \lambda$ genera el mateix llenguatge que la de l'exemple 2.2, és a dir, els mots amb el mateix nombre de a's i b's. A diferència d'aquella, però, es tracta d'una gramàtica ambigua, com posen de manifest els dos arbres de derivació de la figura 2.3, que corresponen al mot *abab*.

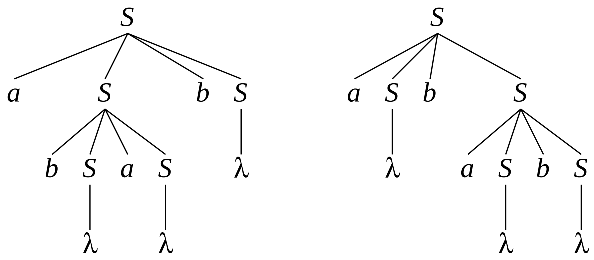

Fig. 2.3 *Un exemple d'ambigüitat*

Exemple 2.10 La gramàtica següent, equivalent a la de l'exemple 2.1, és força més senzilla que aquella, però aquesta és ambigua mentre que aquella no ho era.

$$\begin{aligned}
S &\to S + S \mid F \times F \mid T \\
F &\to (S + S) \mid F \times F \mid T \\
T &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\,.
\end{aligned}$$

Exemple 2.11 Hi ha nombrosos llenguatges de programació que inclouen, entre les seves instruccions de control, les dues següents:

si C **llavors** S_1 **si_no** S_2 i
si C **llavors** S,

on C representa una expressió lògica mentre que S_1, S_2 i S representen instruccions qualssevol. Suposem que l'especificació gramatical d'aquestes dues instruccions es limités a incorporar les regles

$\langle$`instrucció`$\rangle \to$ **si** $\langle$`expressió`$\rangle$ **llavors** $\langle$`instrucció`$\rangle$ **si_no** $\langle$`instrucció`$\rangle$
$\langle$`instrucció`$\rangle \to$ **si** $\langle$`expressió`$\rangle$ **llavors** $\langle$`instrucció`$\rangle$

on ⟨instrucció⟩ i ⟨expressió⟩ són les variables que generen, respectivament, els conjunts d'instruccions i d'expressions lògiques ben formades del llenguatge considerat. És clar que en aquestes condicions la seqüència

si C_1 **llavors si** C_2 **llavors** S_1 **si_no** S_2

admetria els dos arbres de derivació de la figura 2.4.

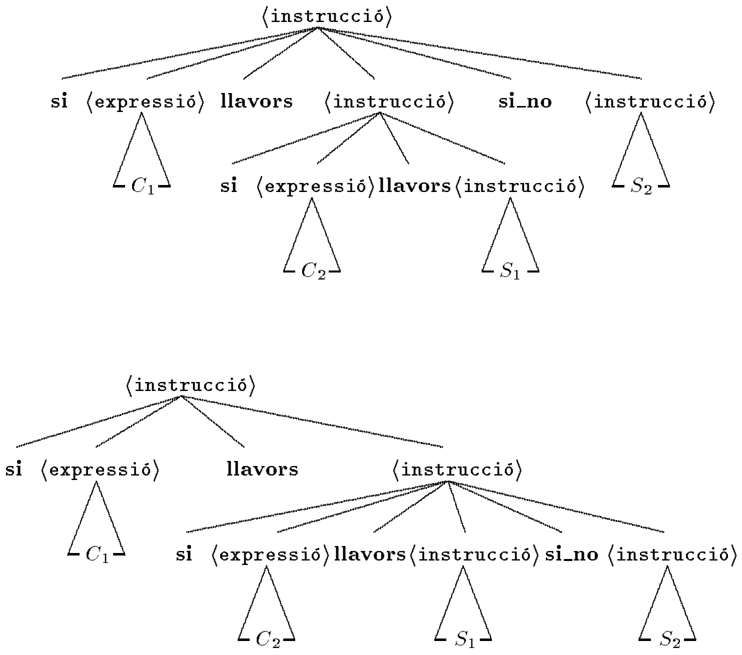

Fig. 2.4 *Ambigüitat en instruccions de control*

El criteri que s'adopta amb caràcter general en aquests casos és el de considerar que, en un procés d'esquerra a dreta, cada '**si_no**' s'ha d'aparellar amb el '**llavors**' més pròxim que el precedeixi i que no estigui encara aparellat. Així doncs, només la derivació que correspon al segon dels arbres dibuixats ha de ser considerada correcta. Una manera senzilla d'impedir una generació de la primera forma consisteix a introduir una segona variable del tipus ⟨instrucció⟩ entre cada '**si_no**' i el seu '**llavors**', a la qual no li és permès de generar la seqüència '**si** ⟨expressió⟩ **llavors** ⟨instrucció⟩'. Es tracta, doncs, de definir les regles

⟨instrucció 0⟩ → **si** ⟨expressió⟩ **llavors** ⟨instrucció 1⟩ **si_no** ⟨instrucció 0⟩ |
  si ⟨expressió⟩ **llavors** ⟨instrucció 0⟩ |
  ⟨instrucció 2⟩
⟨instrucció 1⟩ → **si** ⟨expressió⟩ **llavors** ⟨instrucció 1⟩ **si_no** ⟨instrucció 1⟩ |
  ⟨instrucció 2⟩,

on ara ⟨instrucció 2⟩ és una variable que genera la resta d'instruccions del llenguatge. Una solució equivalent, si bé no pas idèntica, pot trobar-se a la secció 4.3 de [ASU86].

Per posar de manifest que una gramàtica donada és ambigua, n'hi ha prou a mostrar, d'acord amb la definició d'ambigüitat, *un sol* cas d'existència de dos arbres de derivació diferents per a un mateix mot. En canvi, per poder afirmar que una gramàtica és inambigua, cal demostrar que *tot* mot del llenguatge admet un únic arbre de derivació. A la secció següent ens ocuparem del problema de demostrar que una gramàtica genera un cert llenguatge. Veurem que aquest procés demostratiu comporta una doble implicació. D'una banda, que tot mot generat per la gramàtica pertany al llenguatge; d'una altra banda, que tot mot del llenguatge és generat per la gramàtica. Demostrar la inambigüitat consisteix a imposar a aquesta segona implicació la condició addicional d'*unicitat* de la generació. Al capítol següent veurem una manera de caracteritzar les gramàtiques inambigües quan aquestes gramàtiques han estat depurades de símbols inútils.

DEFINICIÓ. Sempre que sigui possible, és útil substituir una gramàtica ambigua per una altra d'equivalent que no ho sigui. Ara bé, no sempre és possible fer això. Diem que un llenguatge és *inherentment ambigu*, o simplement *ambigu*, quan totes les gramàtiques que el generen són ambigües. Altrament diem que el llenguatge és *inambigu*. Si bé la demostració d'existència de llenguatges inherentment ambigus es posposa fins al capítol 7, convé que el lector sàpiga des d'ara que tals llenguatges existeixen. Un exemple de llenguatge inherentment ambigu és

$$\{a^i b^j c^k \mid i=j \vee j=k\}.$$

Tant el problema de reconèixer si una gramàtica és ambigua com el de reconèixer si un llenguatge és inherentment ambigu són problemes *indecidibles*.[2]

2.4 Verificació de gramàtiques

Entenem per *verificació* d'una gramàtica la demostració que aquesta gramàtica genera un llenguatge donat. Solen haver-hi diferents maneres de fer-ho, però hi ha un plantejament força general lligat al caràcter *recursiu* de les gramàtiques. Allò que dóna a les gramàtiques aquest caràcter recursiu és l'existència de derivacions de la forma $A \overset{*}{\Rightarrow} \alpha A \beta$. Donada una gramàtica de variables $V = \{X_1, X_2, \ldots, X_k\}$ i donat un llenguatge L, el plantejament general consisteix a establir k enunciats conjunts de la forma

$$\forall w \quad X_i \overset{*}{\Rightarrow} w \iff w \in L_i$$

on es disposa d'una definició de cada un dels llenguatges L_i per a $i = 1, 2, \ldots, k$. Un d'aquests llenguatges (el que correspon a la variable de sortida) ha de ser el llenguatge L donat. La demostració consisteix a associar a aquest conjunt d'enunciats una funció de fita (generalment la longitud de w), i procedir per inducció sobre aquesta funció numèrica. La recursivitat del conjunt de produccions permet un bon funcionament del mètode d'inducció.

Cal advertir, tanmateix, que no pot existir cap mètode absolutament general de verificació de gramàtiques. Fins i tot si suposem que la descripció dels llenguatges donats es fa mitjançant unes regles estàndards d'especificació (cosa que seria indispensable per poder aplicar qualsevol mètode), el problema és *indecidible*. La definició precisa del significat

[2] Sobre el significat d'aquest terme, vegeu el comentari que apareix al segon paràgraf de la secció següent.

del terme 'indecidible' escapa de l'abast d'aquest curs,[3] però convé diferenciar la mera inexistència d'un mètode (que podria ser deguda a la ignorància) de la impossibilitat d'aquesta existència (cosa que implica un coneixement positiu). També cal diferenciar el coneixement de la solució de problemes específics del coneixement d'un mètode general de solució. Que una categoria de problemes sigui indecidible no significa que no es pugui trobar una solució a cada especificació del problema. Significa que no és possible fer-ho seguint un mètode general.

Exemple 2.12 Demostrarem que la gramàtica

$$\begin{aligned} S &\to aBAS \mid aXS \mid bAAS \mid \lambda \\ A &\to bAAA \mid a \\ B &\to aXB \mid b \\ X &\to aXX \mid aB \end{aligned}$$

genera els mots sobre l'alfabet $\{a, b\}$ que tenen exactament el doble nombre de a's que de b's.

DEMOSTRACIÓ. Considerem la funció f que associa a cada mot w de $\{a, b\}^*$ el valor $\mathsf{f}(w) = 2 \cdot |w|_b - |w|_a$. Representarem per $\mathsf{P}(w)$ el conjunt de prefixos de w diferents del mateix w. Ens proposem demostrar conjuntament les equivalències següents:

$$\forall w \quad \begin{cases} S \overset{*}{\Rightarrow} w & \iff \mathsf{f}(w) = 0 \\ A \overset{*}{\Rightarrow} w & \iff \mathsf{f}(w) = -1 \ \wedge \ \forall x \in \mathsf{P}(w)\ \mathsf{f}(x) \geqslant 0 \\ B \overset{*}{\Rightarrow} w & \iff \mathsf{f}(w) = 2 \ \wedge \ \forall x \in \mathsf{P}(w)\ \mathsf{f}(x) \leqslant 0 \\ X \overset{*}{\Rightarrow} w & \iff \mathsf{f}(w) = 1 \ \wedge \ \forall x \in \mathsf{P}(w)\ \mathsf{f}(x) \leqslant 0. \end{cases}$$

Farem aquesta demostració per inducció sobre $n = |w|$.

Base. Quan $n = 0$, és $w = \lambda$. En tal cas, la primera equivalència se satisfà en ser certs els seus dos membres, i les altres tres se satisfan també en ser falsos els seus dos membres.

Pas inductiu. Quan $n > 0$, es poden donar dos casos: o bé $w = au$, o bé $w = bu$, amb $u \in \{a, b\}^*$ i $|u| < n$. Per a cada un d'aquests dos casos, estudiarem separadament cada una de les quatre equivalències, i per cada equivalència estudiarem primer la implicació d'esquerra a dreta i després la de dreta a esquerra.

Cas 1.1.1 $\quad S \overset{*}{\Rightarrow} au \implies \mathsf{f}(au) = 0.$

Si $S \overset{*}{\Rightarrow} au$, considerant el primer pas d'aquesta derivació, hi ha dues possibilitats:

$$\begin{cases} \exists x, y, z \quad u = xyz \ \wedge \ B \overset{*}{\Rightarrow} x \ \wedge \ A \overset{*}{\Rightarrow} y \ \wedge \ S \overset{*}{\Rightarrow} z \\ \exists x, y \quad u = xy \ \wedge \ X \overset{*}{\Rightarrow} x \ \wedge \ S \overset{*}{\Rightarrow} y. \end{cases}$$

En el primer cas, per hipòtesi d'inducció (h.i. d'ara endavant) es té $\mathsf{f}(x) = 2$, $\mathsf{f}(y) = -1$ i $\mathsf{f}(z) = 0$. En conseqüència, $\mathsf{f}(au) = -1 + 2 - 1 + 0 = 0$. En el segon cas es té $\mathsf{f}(x) = 1$ i $\mathsf{f}(y) = 0$. Per tant, $\mathsf{f}(au) = -1 + 1 + 0 = 0$.

Cas 1.1.2 $\quad \mathsf{f}(au) = 0 \implies S \overset{*}{\Rightarrow} au.$

Considerem el menor prefix v de u tal que $\mathsf{f}(v) = 1$. Sigui $u = vz$. Es té $\mathsf{f}(z) = 0$ i per h.i. $S \overset{*}{\Rightarrow} z$. Hi ha dues possibilitats: o bé $\forall x \in \mathsf{P}(v)$ es té $\mathsf{f}(x) \leqslant 0$; o bé v és de la forma $v = xy$ amb $\mathsf{f}(x) = 2$, $\mathsf{f}(y) = -1$ i amb $\mathsf{f}(t) \leqslant 0$ per a tot $t \in \mathsf{P}(x)$. En el primer cas, per

[3] Podeu trobar-la al capítol 7 de [SACL01].

h.i., $X \stackrel{*}{\Rightarrow} v$, i es té $S \Rightarrow aXS \stackrel{*}{\Rightarrow} au$. En el segon cas és forçós que $\forall s \in \mathsf{P}(y)\ \mathsf{f}(s) \geqslant 0$ i, en conseqüència, per h.i. $B \stackrel{*}{\Rightarrow} x$ i $A \stackrel{*}{\Rightarrow} y$, d'on resulta que $S \Rightarrow aBAS \stackrel{*}{\Rightarrow} au$.

Cas 1.2 $\quad A \stackrel{*}{\Rightarrow} au \iff \mathsf{f}(au) = -1 \ \wedge\ \forall x \in \mathsf{P}(au)\ \mathsf{f}(x) \geqslant 0.$

Aquesta equivalència no cal dividir-la en dues implicacions, ja que és immediat constatar que els dos membres equivalen a la condició $u = \lambda$.

Cas 1.3.1 $\quad B \stackrel{*}{\Rightarrow} au \implies \mathsf{f}(au) = 2 \ \wedge\ \forall x \in \mathsf{P}(au)\ \mathsf{f}(x) \leqslant 0.$

$B \stackrel{*}{\Rightarrow} au$ ha de ser de la forma $B \Rightarrow aXB \stackrel{*}{\Rightarrow} ayz$, amb $u = yz$, amb $X \stackrel{*}{\Rightarrow} y$ i amb $B \stackrel{*}{\Rightarrow} z$. Per h.i. es té que $\mathsf{f}(y) = 1 \ \wedge\ \forall s \in \mathsf{P}(y)\ \mathsf{f}(s) \leqslant 0$ i que $\mathsf{f}(z) = 2 \ \wedge\ \forall t \in \mathsf{P}(z)\ \mathsf{f}(t) \leqslant 0$. D'una banda, es té $\mathsf{f}(au) = \mathsf{f}(ayz) = -1 + 1 + 2 = 2$. D'altra banda, si $x \in \mathsf{P}(au)$ hi ha tres possibilitats:

$$\begin{cases} x = \lambda & \text{i aleshores } \mathsf{f}(x) = 0 \\ x = as \text{ amb } s \in \mathsf{P}(y) & \text{i aleshores } \mathsf{f}(x) = -1 + \mathsf{f}(s) \\ x = ayt \text{ amb } t \in \mathsf{P}(z) & \text{i aleshores } \mathsf{f}(x) = -1 + 1 + \mathsf{f}(t) \end{cases}$$

i en tots els casos resulta $\mathsf{f}(x) \leqslant 0$.

Cas 1.3.2 $\quad \mathsf{f}(au) = 2 \ \wedge\ \forall x \in \mathsf{P}(au)\ \mathsf{f}(x) \leqslant 0 \implies B \stackrel{*}{\Rightarrow} au.$

Sigui v el menor prefix de u tal que $\mathsf{f}(v) = 1$. Aquest prefix existeix i és l'únic que satisfà la condició de no tenir cap prefix propi t amb $\mathsf{f}(t) > 0$. Així doncs, per h.i. $X \stackrel{*}{\Rightarrow} v$. Sigui $u = vz$. De les condicions sobre v resulta que $\mathsf{f}(z) = 2$, i que si $s \in \mathsf{P}(z)$, com que $avs \in \mathsf{P}(au)$, sabem que $\mathsf{f}(avs) \leqslant 0$, és a dir, que $\mathsf{f}(s) \leqslant 0$. En conseqüència, per h.i. $B \stackrel{*}{\Rightarrow} z$. Per tant, podem formar la derivació $B \Rightarrow aXB \stackrel{*}{\Rightarrow} avz$.

Cas 1.4.1 $\quad X \stackrel{*}{\Rightarrow} au \implies \mathsf{f}(au) = 1 \ \wedge\ \forall x \in \mathsf{P}(au)\ \mathsf{f}(x) \leqslant 0.$

Si $X \stackrel{*}{\Rightarrow} au$, considerant el primer pas d'aquesta derivació, hi ha dues possibilitats:

$$\begin{cases} \exists y, z \quad u = yz \ \wedge\ X \stackrel{*}{\Rightarrow} y \ \wedge\ X \stackrel{*}{\Rightarrow} z \\ B \stackrel{*}{\Rightarrow} u. \end{cases}$$

En el primer cas, per h.i. es té que $\mathsf{f}(y) = \mathsf{f}(z) = 1$, que $\forall s \in \mathsf{P}(y)\ \mathsf{f}(s) \leqslant 0$ i que $\forall t \in \mathsf{P}(z)\ \mathsf{f}(t) \leqslant 0$. En conseqüència, $\mathsf{f}(ayz) = -1 + 1 + 1 = 1$. D'altra banda, si $x \in \mathsf{P}(au)$ hi ha tres possibilitats:

$$\begin{cases} x = \lambda & \text{i aleshores } \mathsf{f}(x) = 0 \\ x = as \text{ amb } s \in \mathsf{P}(y) & \text{i aleshores } \mathsf{f}(x) = -1 + \mathsf{f}(s) \\ x = ayt \text{ amb } t \in \mathsf{P}(z) & \text{i aleshores } \mathsf{f}(x) = -1 + 1 + \mathsf{f}(t) \end{cases}$$

i en tots els casos resulta $\mathsf{f}(x) \leqslant 0$.

En el segon cas, la h.i. dóna $\mathsf{f}(u) = 2$ i $\forall s \in \mathsf{P}(u)\ \mathsf{f}(s) \leqslant 0$. Per tant, $\mathsf{f}(au) = 1$. Quant als prefixos $x \in \mathsf{P}(au)$, poden ser o bé λ, o bé de la forma $x = as$ per a algun $s \in \mathsf{P}(u)$. En tots els casos resulta $\mathsf{f}(x) \leqslant 0$.

Cas 1.4.2 $\quad \mathsf{f}(au) = 1 \ \wedge\ \forall x \in \mathsf{P}(au)\ \mathsf{f}(x) \leqslant 0 \implies X \stackrel{*}{\Rightarrow} au.$

Hi ha dues possibilitats. O bé existeix un prefix y de u amb $\mathsf{f}(y) = 1$ (cas, per exemple, de $u = abaabab$ amb $y = ab$ o $y = abaab$). O bé no n'existeix cap (cas de $u = aabab$).

En el primer cas, considerem el mínim prefix y que satisfà $\mathsf{f}(y) = 1$. Aquest mot y satisfà, a més, la condició $\forall s \in \mathsf{P}(y)\ \mathsf{f}(s) \leqslant 0$, i és l'únic prefix de u que la satisfà. Si anomenem $u = yz$, resulta que z satisfà les dues condicions $\mathsf{f}(z) = 1$ i $\forall t \in \mathsf{P}(z)\ \mathsf{f}(t) \leqslant 0$. Ara, la h.i. permet establir que $X \stackrel{*}{\Rightarrow} y$ i que $X \stackrel{*}{\Rightarrow} z$. Per tant, es té $X \Rightarrow aXX \stackrel{*}{\Rightarrow} ayz$.

En el segon cas, es té que $\forall s \in \mathsf{P}(u)\ \mathsf{f}(s) \leqslant 0$ i com que $\mathsf{f}(u) = 2$ resulta, per h.i., que $B \stackrel{*}{\Rightarrow} u$ i, per tant, es té $X \Rightarrow aB \stackrel{*}{\Rightarrow} au$.

Cas 2.1.1 $S \stackrel{*}{\Rightarrow} bu \implies \mathsf{f}(bu) = 0$.

La derivació $S \stackrel{*}{\Rightarrow} bu$ ha de ser de la forma $S \Rightarrow bAAS \stackrel{*}{\Rightarrow} bxyz$, amb $u = xyz$, amb $A \stackrel{*}{\Rightarrow} x$, amb $A \stackrel{*}{\Rightarrow} y$ i amb $S \stackrel{*}{\Rightarrow} z$. Per h.i. tenim $\mathsf{f}(x) = \mathsf{f}(y) = -1$ i $\mathsf{f}(z) = 0$. Per tant, $\mathsf{f}(bu) = 2 - 1 - 1 + 0 = 0$.

Cas 2.1.2 $\mathsf{f}(bu) = 0 \implies S \stackrel{*}{\Rightarrow} bu$.

Partim de $\mathsf{f}(u) = -2$. Com que la funció f només pot decréixer d'unitat en unitat quan considerem prefixos creixents d'esquerra a dreta, podem dividir el mot u en tres factors $u = xyz$, de manera que x i y siguin els mínims factors inicials que tenen $\mathsf{f}(x) = \mathsf{f}(y) = -1$. El resultat és que aquesta és l'única factorització que satisfà les condicions $\forall s \in \mathsf{P}(x)\ \mathsf{f}(s) \geqslant 0$ i $\forall t \in \mathsf{P}(y)\ \mathsf{f}(t) \geqslant 0$. Per h.i. tenim $A \stackrel{*}{\Rightarrow} x$ i $A \stackrel{*}{\Rightarrow} y$. A més, com que $\mathsf{f}(z) = 0$, resulta $S \stackrel{*}{\Rightarrow} z$. Concloem que $S \Rightarrow bAAS \stackrel{*}{\Rightarrow} bxyz$.

Cas 2.2.1 $A \stackrel{*}{\Rightarrow} bu \implies \mathsf{f}(bu) = -1 \wedge \forall v \in \mathsf{P}(bu)\ \mathsf{f}(v) \geqslant 0$.

La derivació $A \stackrel{*}{\Rightarrow} bu$ ha de ser de la forma $A \Rightarrow bAAA \stackrel{*}{\Rightarrow} bxyz$, amb $u = xyz$, amb $A \stackrel{*}{\Rightarrow} x$, amb $A \stackrel{*}{\Rightarrow} y$ i amb $A \stackrel{*}{\Rightarrow} z$. Per h.i. tenim $\mathsf{f}(x) = \mathsf{f}(y) = \mathsf{f}(z) = -1$, i les condicions relatives als prefixos: $\forall r \in \mathsf{P}(x)\ \mathsf{f}(r) \geqslant 0$, $\forall s \in \mathsf{P}(y)\ \mathsf{f}(s) \geqslant 0$ i $\forall t \in \mathsf{P}(z)\ \mathsf{f}(t) \geqslant 0$. Així doncs, per una part tenim $\mathsf{f}(bxyz) = 2 - 1 - 1 - 1 = -1$, i per una altra, si $v \in \mathsf{P}(bu)$, els casos possibles són

$$\begin{cases} v = \lambda & & \text{i aleshores } \mathsf{f}(v) = 0 \\ v = br & \text{amb } r \in \mathsf{P}(x) & \text{i aleshores } \mathsf{f}(v) = 2 + \mathsf{f}(r) \\ v = bxs & \text{amb } s \in \mathsf{P}(y) & \text{i aleshores } \mathsf{f}(v) = 2 - 1 + \mathsf{f}(s) \\ v = bxyt & \text{amb } t \in \mathsf{P}(z) & \text{i aleshores } \mathsf{f}(v) = 2 - 1 - 1 + \mathsf{f}(t) \end{cases}$$

i en tots els casos resulta $\mathsf{f}(v) \geqslant 0$.

Cas 2.2.2 $\mathsf{f}(bu) = -1 \wedge \forall v \in \mathsf{P}(bu)\ \mathsf{f}(v) \geqslant 0 \implies A \stackrel{*}{\Rightarrow} bu$.

Tenim ara que $\mathsf{f}(u) = -3$. Per un raonament anàleg al del cas 2.1.2, podem dividir el mot u en tres factors $u = xyz$, de manera que x és el mínim prefix de u amb $\mathsf{f}(x) = -1$; i que del sufix restant, y n'és el mínim prefix amb $\mathsf{f}(y) = -1$. En aquestes condicions, podem assegurar que aquesta és l'única descomposició de u en tres factors, x, y i z, que satisfan les propietats $\mathsf{f}(x) = \mathsf{f}(y) = \mathsf{f}(z) = -1$, $\forall r \in \mathsf{P}(x)\ \mathsf{f}(r) \geqslant 0$, $\forall s \in \mathsf{P}(y)\ \mathsf{f}(s) \geqslant 0$ i $\forall t \in \mathsf{P}(z)\ \mathsf{f}(t) \geqslant 0$. La h.i. ens dóna $A \stackrel{*}{\Rightarrow} x$, $A \stackrel{*}{\Rightarrow} y$ i $A \stackrel{*}{\Rightarrow} z$. Existeix, per tant la derivació $A \Rightarrow bAAA \stackrel{*}{\Rightarrow} bxyz$.

Cas 2.3 $B \stackrel{*}{\Rightarrow} bu \iff \mathsf{f}(bu) = 2 \wedge \forall x \in \mathsf{P}(bu)\ \mathsf{f}(x) \leqslant 0$.

Anàlogament al cas 1.2, aquesta equivalència no cal dividir-la en dues implicacions, ja que és immediat constatar que els dos membres equivalen a la condició $u = \lambda$.

Cas 2.4 $X \stackrel{*}{\Rightarrow} bu \iff \mathsf{f}(bu) = 1 \wedge \forall x \in \mathsf{P}(bu)\ \mathsf{f}(x) \leqslant 0$.

Tampoc en aquest cas no cal separar les dues implicacions, ja que no hi ha cap valor de u que satisfaci cap dels dos membres. Així doncs, els dos membres són falsos (i l'equivalència, per tant, és certa).

Malgrat que el procediment seguit a l'exemple anterior és força general, hi ha casos en què es fa difícil aplicar el mètode d'inducció si es pren com a funció de fita la longitud dels mots. En tals casos, l'elecció d'una funció de fita diferent pot evitar que la demostració quedi encallada. Ho veurem amb l'exemple següent.

Exemple 2.13 Es tracta de demostrar que la gramàtica

$$S \rightarrow\ aSb \mid SS \mid \lambda$$

genera el llenguatge de mots ben construïts sobre un parell de "parèntesis" simbolitzats per a i b. És a dir, el llenguatge

$$L = \{w \in \{a,b\}^* \ \mid \ |w|_a = |w|_b \ \wedge \ \forall x,y \ \ w = xy \Rightarrow |x|_a \geqslant |x|_b\}.$$

INTENT DE DEMOSTRACIÓ. Considerem l'enunciat

$$\forall w \quad S \overset{*}{\Rightarrow} w \iff w \in L$$

i intentem demostrar-lo per inducció sobre $n = |w|$.

Base. Quan $n = 0$, és que $w = \lambda$ i, en tal cas, l'equivalència se satisfà en ser certs els seus dos membres.

Pas inductiu. Quan $n > 0$, w ha de ser de la forma $w = au$ o de la forma $w = bu$, amb $u \in \{a,b\}^*$ i $|u| < n$. Intentarem procedir com en l'exemple precedent, separant els dos casos i estudiant per a cada cas primer la implicació d'esquerra a dreta i després la de dreta a esquerra.

Cas 1.1 $\quad S \overset{*}{\Rightarrow} au \implies (|au|_a = |au|_b \ \wedge \ \forall x,y \ \ au = xy \Rightarrow |x|_a \geqslant |x|_b)$.

Si $S \overset{*}{\Rightarrow} au$, considerant el primer pas d'aquesta derivació, hi ha dues possibilitats:

$$\begin{cases} \exists v \quad u = vb \ \wedge \ S \overset{*}{\Rightarrow} v \\ \exists v,z \quad au = vz \ \wedge \ S \overset{*}{\Rightarrow} v \ \wedge \ S \overset{*}{\Rightarrow} z. \end{cases}$$

En el primer cas, per h.i. sabem que $v \in L$, i d'això resulta immediatament que $avb \in L$.

Però en el segon cas, entre les descomposicions possibles de au en dos factors v i z, hi ha les dues extremes,

$$\begin{cases} v = \lambda & \wedge & z = au \\ v = au & \wedge & z = \lambda \end{cases}$$

que ens retornen a la situació anterior, sense que hi hagi un decreixement en la funció de fita. Això ens impedeix de progressar de manera senzilla en el mètode d'inducció.

En aquests casos, pot ser útil prendre una funció de fita més adaptada a la generativitat de les gramàtiques que no pas la longitud dels mots generats. Fixem-nos que, en l'exemple anterior, el problema ha aparegut en el decurs de demostrar la implicació d'esquerra a dreta. Com veurem a continuació en la represa d'aquest exemple, no hi ha cap problema en la demostració de la implicació recíproca. És a dir, aquest segon tipus d'implicacions el continuarem expressant de la forma

$$\forall n \geqslant 0 \ \ \forall w \in \Sigma^n \quad w \in L_i \implies X_i \overset{*}{\Rightarrow} w$$

i farem inducció sobre n, la longitud del mot. En canvi, l'altre tipus d'implicacions és millor rescriure'l de la forma

$$\forall n \geqslant 0 \ \ \forall w \in \Sigma^* \quad X_i \overset{n}{\Rightarrow} w \implies w \in L_i$$

i fer inducció sobre el nombre de passos en la derivació dels mots. Ho veurem a l'exemple següent.

Exemple 2.14 Reprenem la gramàtica i el llenguatge L de l'exemple anterior, i fem ara la demostració de la implicació d'esquerra a dreta per inducció sobre el nombre n de passos de la derivació

$$S \overset{n}{\Rightarrow} w \implies w \in L.$$

En lloc de plantejar la inducció en termes de 'base' i 'pas inductiu', aquí és més convenient fer-ho seguint l'esquema alternatiu

$$\forall n\ P(n) \quad \Longleftrightarrow \quad \forall n\ \Big((\forall m < n\ P(m)) \ \Rightarrow\ P(n) \Big),$$

és a dir, la hipòtesi d'inducció consisteix a considerar que l'enunciat $P(n)$ se satisfà per a tots els valors menors que el n considerat. A partir d'això, s'ha de demostrar que també se satisfà per a n. En el nostre cas, la h.i. serà que $S \overset{m}{\Rightarrow} w \implies w \in L$ per a tot $m < n$.

Sigui $S \overset{n}{\Rightarrow} w$. Com que $w \neq S$, cal que sigui $n \geqslant 1$. Per tant, la derivació considerada ha de ser d'una d'aquestes tres formes:

$$S \overset{n}{\Rightarrow} w \quad \text{és} \quad \begin{cases} S \Rightarrow \lambda \\ S \Rightarrow SS \overset{n-1}{\Longrightarrow} w \\ S \Rightarrow aSb \overset{n-1}{\Longrightarrow} w. \end{cases}$$

- En el primer cas, resulta $w = \lambda$, que pertany a L.
- En el segon cas, w ha de ser de la forma $w = uv$ amb $S \overset{m_1}{\Longrightarrow} w$ i $S \overset{m_2}{\Longrightarrow} w$, on $m_1, m_2 < n$. Per h.i. tenim $u, v \in L$, i d'això fàcilment deduïm que $w \in L$.
- En el tercer cas, w ha de ser de la forma $w = aub$, amb $S \overset{n-1}{\Longrightarrow} u$. Per h.i. tenim $u \in L$, i d'això resulta $w \in L$.

La implicació recíproca la farem per inducció sobre $|w|$. Sigui $w \in L$. Si $w = \lambda$, tenim $S \Rightarrow w$. Si $w \neq \lambda$, sigui $w = uv$, on u és el menor prefix no buit de w que pertany a L. Aquest prefix existeix, ja que en el cas extrem és el mateix w. Poden donar-se dos casos:

- Si $u \neq w$, és fàcil veure que $v \in L$ i, per h.i., $S \overset{*}{\Rightarrow} u$ i $S \overset{*}{\Rightarrow} v$, d'on $S \Rightarrow SS \overset{*}{\Rightarrow} uv$.
- Si $u = w$, aleshores és fàcil veure que w és de la forma $w = avb$ amb $v \in L$. Per h.i. tenim $S \overset{*}{\Rightarrow} v$, i d'això $S \Rightarrow aSb \overset{*}{\Rightarrow} avb$.

De tota manera, al capítol següent veurem que la dificultat d'aplicar inducció sobre la longitud dels mots, dificultat que l'exemple anterior posa de manifest, desapareix quan la gramàtica està en forma *depurada*.

2.5 Operacions bàsiques amb gramàtiques

Considerarem a continuació un seguit d'operacions bàsiques sobre llenguatges, les quals tenen en comú que conserven la incontextualitat, és a dir, aplicades a llenguatges incontextuals donen com a resultat també llenguatges incontextuals. Les demostracions seguiran processos constructius: es partirà de les gramàtiques que generen els llenguatges operands i s'obtindrà la gramàtica que genera el llenguatge resultant.

Reunió de dos CFL

Siguin $G_1 = \langle V_1, \Sigma_1, P_1, S_1 \rangle$ i $G_2 = \langle V_2, \Sigma_2, P_2, S_2 \rangle$ dues gramàtiques que generen els llenguatges L_1 i L_2, respectivament. Podem considerar, sense pèrdua de generalitat, que els alfabets V_1 i V_2 són disjunts, ja que sempre podem obtenir una gramàtica equivalent a una de donada modificant les seves variables (p. ex., canviant A per A', B per B', etc.). Introduïm una nova variable, S, que no pertanyi ni a V_1 ni a V_2 i considerem la gramàtica $G = \langle V, \Sigma, P, S \rangle$, on $V = V_1 \cup V_2 \cup \{S\}$, on $\Sigma = \Sigma_1 \cup \Sigma_2$ i on P està formada per totes les produccions de P_1 i P_2 i, a més, les dues produccions $S \to S_1 \mid S_2$. Aquesta gramàtica genera el llenguatge $L_1 \cup L_2$. Ho demostrarem després de veure'n un exemple.

Exemple 2.15 Considerem el llenguatge $L = \{a^i b^j \mid i \neq j\}$ i volem construir una gramàtica que el generi. Podem escriure L de la forma

$$L = \{a^i b^j \mid i > j \ \vee \ i < j\} = \{a^i b^j \mid i > j\} \cup \{a^i b^j \mid i < j\},$$

on els llenguatges components tenen per gramàtiques $\{A \to aA \mid aAb \mid a\}$ i $\{B \to Bb \mid aBb \mid b\}$. Ara, la gramàtica $G = \langle \{S, A, B\}, \{a, b\}, P, S \rangle$, on P està format per les produccions

$$\begin{aligned} S &\to A \mid B \\ A &\to aA \mid aAb \mid a \\ B &\to Bb \mid aBb \mid b\,, \end{aligned}$$

genera el llenguatge L.

PROPOSICIÓ. $\mathsf{L}(G) = L_1 \cup L_2$.

DEMOSTRACIÓ. Per a cada mot w de L_1 existeix una derivació $S_1 \overset{*}{\Rightarrow} w$ en G_1. Com que totes les produccions de P_1 han estat incorporades a P, podem escriure amb produccions de P la derivació $S \Rightarrow S_1 \overset{*}{\Rightarrow} w$; així doncs, $w \in L$. El mateix raonament s'aplica als mots de L_2, i concloem que $L_1 \cup L_2 \subseteq \mathsf{L}(G)$.

Recíprocament, per a tot mot w de $\mathsf{L}(G)$ existeix una derivació de la forma $S \overset{*}{\Rightarrow} w$. Però aquesta derivació només pot ser d'un dels dos tipus, $S \Rightarrow S_1 \overset{*}{\Rightarrow} w$ o $S \Rightarrow S_2 \overset{*}{\Rightarrow} w$, ja que només hi ha dues produccions possibles per a S. Els mots que provenen de derivacions del primer tipus forçosament han de pertànyer a L_1, ja que tota la derivació $S_1 \overset{*}{\Rightarrow} w$ es fa amb produccions de P_1. Anàlogament, les derivacions del segon tipus donen lloc a mots de L_2. En conclusió, $\mathsf{L}(G) \subseteq L_1 \cup L_2$. □

Com a resultat de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de la reunió.*

Concatenació de dos CFL

Siguin, com abans, $G_1 = \langle V_1, \Sigma_1, P_1, S_1 \rangle$ i $G_2 = \langle V_2, \Sigma_2, P_2, S_2 \rangle$ dues gramàtiques que generen els llenguatges L_1 i L_2, respectivament, i tals que $V_1 \cap V_2 = \varnothing$. Considerem, també com abans, una gramàtica $G = \langle V, \Sigma, P, S \rangle$, on $S \notin V_1 \cup V_2$ torna a ser una variable nova, on $V = V_1 \cup V_2 \cup \{S\}$, on $\Sigma = \Sigma_1 \cup \Sigma_2$, però on ara P està formada per totes les produccions de P_1 i P_2, i per $S \to S_1 S_2$. Sigui L el llenguatge generat per G; es té $L = L_1 L_2$.

Exemple 2.16 Ens proposem de construir una gramàtica per al llenguatge
$$L = \{a^i b^j c^k \mid i, j, k \geqslant 0 \ \wedge \ j = i + k\}.$$
Podem expressar L com a concatenació de dos llenguatges. En efecte,
$$L = \{a^i b^i b^k c^k \mid i, k \geqslant 0\} = \{a^i b^i \mid i \geqslant 0\} \cdot \{b^k c^k \mid k \geqslant 0\},$$
on els llenguatges components són generats per les gramàtiques de produccions $\{X \to aXb \mid \lambda\}$ i $\{Y \to bYc \mid \lambda\}$. En conseqüència el llenguatge L és generat per la gramàtica de produccions
$$\begin{aligned} S &\to XY \\ X &\to aXb \mid \lambda \\ Y &\to bYc \mid \lambda. \end{aligned}$$

PROPOSICIÓ. $\mathsf{L}(G) = L_1 \cdot L_2$.

DEMOSTRACIÓ. Sigui $w \in L_1 L_2$ i siguin $x \in L_1$ i $y \in L_2$ tals que $w = xy$. Existeixen, doncs, sengles derivacions $S_1 \stackrel{*}{\Rightarrow} x$ i $S_2 \stackrel{*}{\Rightarrow} y$ en les gramàtiques G_1 i G_2, respectivament. Com que $P_1, P_2 \subset P$, podem construir a G la derivació $S \Rightarrow S_1 S_2 \stackrel{*}{\Rightarrow} xy$ i resulta $xy \in L$.

Recíprocament, tot mot $w \in L$ ha de tenir una derivació de la forma $S \Rightarrow S_1 S_2 \stackrel{*}{\Rightarrow} w$, ja que és forçosa la utilització de la producció $S \to S_1 S_2$. D'això resulta necessàriament que w és de la forma $w = xy$, amb $S_1 \stackrel{*}{\Rightarrow} x$ i $S_2 \stackrel{*}{\Rightarrow} y$. Com que no és possible que en la derivació de x intervinguin produccions de P_2, perquè partim de S_1, resulta $x \in L_1$. Anàlogament, tenim $y \in L_2$. Per tant $w \in L_1 L_2$. $\square$

Com a resultat de la proposició anterior podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de la concatenació.*

Tancament de Kleene i tancament positiu d'un CFL

Sigui $G_1 = \langle V_1, \Sigma, P_1, S_1 \rangle$ una gramàtica que genera el llenguatge L_1. Considerem la gramàtica $G = \langle V, \Sigma, P, S \rangle$ on $S \notin V_1$ és una variable nova, on $V = V_1 \cup \{S\}$ i on P s'ha obtingut afegint a P_1 les produccions $\{S \to S_1 S \mid \lambda\}$. La gramàtica G genera $L = L_1^*$.

PROPOSICIÓ. $\mathsf{L}(G) = L_1^*$.

DEMOSTRACIÓ. Tot mot de L_1^* o bé és λ, o bé és de la forma $w_1 \ldots w_n$, amb $w_1, \ldots, w_n \in L_1$. En el primer cas, la producció $S \to \lambda$ garanteix que $\lambda \in L$. En el segon cas, existeixen a G_1 derivacions de la forma $S_1 \stackrel{*}{\Rightarrow} w_i$ per a tot i entre 1 i n. Per tant, podem formar a G la derivació $S \Rightarrow S_1 S \Rightarrow \ldots \Rightarrow S_1^n S \Rightarrow S_1^n \stackrel{*}{\Rightarrow} w_1 \ldots w_n$ i concloem que $w_1 \ldots w_n \in L$.

Recíprocament, tota derivació de G o bé és $S \Rightarrow \lambda$, o bé es pot rescriure de manera que tingui la forma $S \stackrel{n}{\Rightarrow} S_1^n S \Rightarrow S_1^n \stackrel{*}{\Rightarrow} w$, ja que S no apareix a cap altra producció. Per tant, w és de la forma $w_1 \ldots w_n$ amb $w_1, \ldots, w_n \in L_1$ i resulta $w \in L_1^n \subseteq L_1^*$. $\square$

Quant al tancament positiu, la gramàtica que correspon a L_1^+ només es diferencia de la que correspon a L_1^* en la substitució de la producció $S \to \lambda$ per la producció $S \to S_1$. La demostració és anàloga a l'anterior. En conseqüència, obtenim el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de l'estrella de Kleene i del tancament positiu.*

Exemple 2.17 Sigui Σ un alfabet. Un subconjunt $C \subseteq \Sigma^*$ s'anomena *codi* si C genera lliurement C^*, és a dir, si tot mot de C^+ factoritza de manera única en mots de C. Un exemple molt elemental de codi sobre l'alfabet $\Sigma = \{a, b\}$ és el conjunt de mots de la forma ba^n per a $n \geqslant 0$, que admet la gramàtica $S_0 \to S_0 a \mid b$. Així doncs, les gramàtiques que generen C^* i C^+ són, respectivament,

$$C^* : \begin{cases} S_1 \to S_0 S_1 \mid \lambda \\ S_0 \to S_0 a \mid b \end{cases} \quad \text{i} \quad C^+ : \begin{cases} S_2 \to S_0 S_2 \mid S_0 \\ S_0 \to S_0 a \mid b\,. \end{cases}$$

Revessament d'un CFL

Sigui $G_1 = \langle V, \Sigma, P_1, S \rangle$ una gramàtica que genera el llenguatge L_1. Considerem la gramàtica $G = \langle V, \Sigma, P, S \rangle$ on $P = \{X \to \alpha^{\mathsf{R}} \mid (X \to \alpha) \in P_1\}$, és a dir, l'únic canvi consisteix a revessar els costats drets de totes les produccions de la gramàtica de partida. Aquesta gramàtica genera $L = L_1^{\mathsf{R}}$. En efecte, per cada derivació de G_1 podem escriure una derivació de G amb totes les cadenes intermèdies revessades, i recíprocament.

Exemple 2.18 La gramàtica que té per produccions

$$S \to +SS \mid \times SS \mid 0 \mid 1 \mid \cdots \mid 9$$

genera en notació *prefixa* les expressions aritmètiques en què només intervenen sumes i productes de dígits. La gramàtica que genera el mateix tipus d'expressions, però en notació *postfixa*, és, doncs,

$$S \to SS+ \mid SS\times \mid 0 \mid 1 \mid \cdots \mid 9.$$

Observeu que si considerem una expressió concreta, com ara $a+(b\times c)$, on a, b i c poden ser dígits qualssevol, i l'escrivim en notació prefixa, $+a\times bc$, la revessada d'aquesta, $cb\times a+$, ha quedat en notació postfixa, però ja no és l'expressió original. En efecte, l'expressió de partida, escrita en notació postfixa, és $abc\times+$. Aquest fet pot no ser rellevant en aquest exemple, perquè les operacions considerades són commutatives i les dues expressions obtingudes en notació postfixa tenen el mateix *valor*. Però això deixa de ser així quan considerem operadors no commutatius.

Morfisme d'un CFL

Sigui $G_1 = \langle V, \Sigma_1, P_1, S \rangle$ una gramàtica que genera el llenguatge L_1, i sigui $h\colon \Sigma_1^* \longrightarrow \Sigma_2^*$ un morfisme. Considerem la gramàtica $G_2 = \langle V, \Sigma_2, P_2, S \rangle$, on P_2 està format per les mateixes produccions que P_1 en les quals s'ha substituït cada aparició d'un símbol de Σ_1 per la seva imatge en el morfisme h. Podem formalitzar aquesta construcció de la manera següent.

Sigui $\tilde{h}$ l'extensió del morfisme h al domini $(\Sigma_1 \cup V)^*$ que deixa invariats el símbols de V, és a dir, per als símbols terminals $a \in \Sigma_1$ es té $\tilde{h}(a) = h(a)$, mentre que per a les variables $X \in V$ es té simplement $\tilde{h}(X) = X$. Amb aquesta notació, el conjunt de produccions de G_2 és

$$P_2 = \{A \to \tilde{h}(\alpha) \mid (A \to \alpha) \in P_1\}.$$

D'ara endavant identificarem $\tilde{h}$ i h. Abans de demostrar que G_2 genera $h(L_1)$, vegem un parell d'exemples.

Exemple 2.19 Considerem la gramàtica definida per les produccions

$$S \to aSb \mid c\,.$$

Es tracta clarament d'una gramàtica que genera el llenguatge

$$L = \{a^n cb^n \mid n \geqslant 0\}.$$

Considerem ara el morfisme definit per $h(a) = 0\,,\ h(b) = 00$ i $h(c) = \lambda$. És immediat constatar que $h(L) = \{000\}^*$. Una gramàtica que genera aquest llenguatge, obtinguda a partir de la construcció anterior, és

$$S \to 0S00 \mid \lambda\,.$$

Podem adonar-nos immediatament que aquesta no és probablement la gramàtica que hauríem escollit a l'hora de triar-ne alguna per generar aquest llenguatge. Possiblement hauríem preferit una gramàtica com ara

$$S \to 000S \mid \lambda\,.$$

Si, d'altra banda, considerem ara el morfisme invers del llenguatge $\{000\}^*$ obtenim

$$h^{-1}(\{000\}^*) = \big\{w \in \{a,b,c\}^* \mid |w|_a + 2|w|_b = \dot{3}\big\}\,,$$

llenguatge que inclou el llenguatge L de partida, però també altres mots. Així, per exemple, tenim que $bac \in h^{-1}(h(L))$, mentre que $bac \notin L$.

Exemple 2.20 Considerem la gramàtica següent:

$$G_0\colon \quad S \to aS \mid aSbS \mid c.$$

És fàcil de constatar que es tracta d'una gramàtica ambigua. Considerem ara les dues gramàtiques següents, que no ho són i que generen el mateix llenguatge que G_0:

$$G_1\colon \begin{cases} S \to A \mid B \\ A \to aAbA \mid c \\ B \to aS \mid aAbB \end{cases}$$

i

$$G_2\colon \begin{cases} S \to aS \mid aAbS \mid c \\ A \to aAbA \mid c\,. \end{cases}$$

Demostrar que aquestes gramàtiques són no ambigües no és gaire senzill. Cal fer una demostració que sigui de tipus estrictament sintàctic. Però el problema s'aclareix notablement si considerem el morfisme següent,

$$\begin{aligned} h(a) &= \textbf{si}\,\langle\texttt{expressió}\rangle\,\textbf{llavors} \\ h(b) &= \textbf{si\_no} \\ h(c) &= \langle\texttt{instrucció 2}\rangle\,. \end{aligned}$$

Això ens permet donar una *interpretació semàntica* del problema proposat, de manera que G_0 i G_2 donen lloc a les gramàtiques introduïdes a l'exemple 2.11, mentre que

G_1 dóna lloc a la gramàtica proposada per Aho, Sethi i Ullman [ASU86, secció 4.3]. Remarquem que, un cop interpretat el problema, la solució sintàctica és més fàcil de formular en termes de les gramàtiques G_0, G_1 i G_2 que en termes de les gramàtiques originals.

PROPOSICIÓ. $\mathsf{L}(G_2) = h(L_1)$.

DEMOSTRACIÓ. Farem, com de costum, la demostració de les dues inclusions per separat. Començarem per la inclusió $h(L_1) \subseteq \mathsf{L}(G_2)$ que és la més senzilla. Que per a tot mot $w \in L_1$ es tingui que la seva imatge $h(w)$ pertany a $\mathsf{L}(G_2)$ és un cas particular de l'enunciat següent, referit a cadenes intermèdies β qualssevol,

$$\forall \beta \quad S \overset{n}{\underset{G_1}{\Rightarrow}} \beta \implies S \overset{n}{\underset{G_2}{\Rightarrow}} h(\beta).$$

La demostració d'aquest enunciat per inducció sobre n es deixa com a exercici.

La inclusió recíproca, $\mathsf{L}(G_2) \subseteq h(L_1)$, s'obté de considerar un mot w qualsevol i fer $\beta' = w$ a l'enunciat següent:

$$\forall \beta' \quad S \overset{n}{\underset{G_2}{\Rightarrow}} \beta' \implies \exists \beta \;\; \beta' = h(\beta) \, \wedge \, S \overset{n}{\underset{G_1}{\Rightarrow}} \beta,$$

que demostrarem per inducció sobre n.

- *Base.* Quan $n = 0$, es té $\beta' = S$ i l'enunciat se satisfà fent $\beta = S$.
- *Pas inductiu.* Quan $n > 0$, podem separar l'últim pas de la derivació de β' dels $n-1$ anteriors. Sigui

$$S \overset{n-1}{\underset{G_2}{\Rightarrow}} \alpha' \underset{G_2}{\Rightarrow} \beta'.$$

a) Per hipòtesi d'inducció tenim $\exists \alpha \;\; \alpha' = h(\alpha) \, \wedge \, S \overset{n-1}{\underset{G_1}{\Rightarrow}} \alpha$.

b) La consideració de l'últim pas permet establir que

$$\exists \alpha_1', \alpha_2', \gamma \;\; \exists A \quad (\alpha' = \alpha_1' A \alpha_2') \wedge (\beta' = \alpha_1' h(\gamma) \alpha_2') \wedge (A \to \gamma \in P_1).^4$$

Així doncs, l'equació $\alpha_1' A \alpha_2' = h(\alpha)$ ens permet descompondre α de la forma $\alpha = \alpha_1 A \alpha_2$, amb $\alpha_1' = h(\alpha_1)$ i $\alpha_2' = h(\alpha_2)$. L'enunciat proposat es dedueix ara de forma immediata de prendre $\beta = \alpha_1 \gamma \alpha_2$. □

Com a conseqüència de la proposició anterior podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de l'aplicació de morfismes directes.*

Al capítol 8 veurem que la família dels CFL també és tancada respecte dels morfismes inversos. Però cal advertir que en cap cas no estem afirmant que, quan la imatge d'un llenguatge per un morfisme sigui un CFL, el llenguatge considerat també hagi de ser un CFL. Pot ser que ho sigui i pot ser que no.

[4] Pot ser que aquesta descomposició de α' i β' no sigui única, pel fet que $h(\gamma) = A$, en el qual cas resulta $\alpha' = \beta'$. Però el raonament és vàlid encara que no hi hagi tal unicitat.

Exemple 2.21 Considerem el llenguatge $L = \{a^n b^n c^n \mid n \geqslant 0\}$, al qual ens referirem a l'exemple 7.12 com a prototipus de llenguatge no incontextual. Considerem el morfisme $h(a) = a$, $h(b) = b$ i $h(c) = \lambda$. Es té $h(L) = \{a^n b^n \mid n \geqslant 0\}$, que sí que és incontextual, com s'ha vist a l'exemple 2.3.

Substitució incontextual d'un CFL

Sigui $G' = \langle V', \Sigma', P', S\rangle$ una gramàtica que genera el llenguatge L', i sigui $\sigma\colon \Sigma'^* \longrightarrow \mathsf{CFL}$ una substitució on la imatge de cada mot és un llenguatge incontextual. Direm que σ és una *substitució incontextual*. Considerem que aquesta substitució ve definida per les imatges dels símbols de Σ'. Sigui $\Sigma' = \{a_1, \ldots, a_r\}$, i sigui $L_i = \sigma(a_i)$ per a cada i entre 1 i r. Per tractar-se de CFL, a cada L_i podem associar-li una gramàtica $G_i = \langle V_i, \Sigma_i, P_i, S_i\rangle$ que el genera.

Construïm una gramàtica $G = \langle V, \Sigma, P, S\rangle$ fent

$$\begin{aligned} V =& V' \cup V_1 \cup \cdots \cup V_r \\ \Sigma =& \Sigma_1 \cup \cdots \cup \Sigma_r \end{aligned}$$

i on P està format per totes les produccions de $P_1 \cup \cdots \cup P_r$, a les quals afegim les produccions obtingudes a partir de les de P' en les quals s'ha substituït cada aparició d'un símbol a_i de Σ' per la variable inicial S_i que li correspon. Podem formalitzar aquesta construcció de la manera següent. Sigui

$$f\colon (V' \cup \Sigma')^* \longrightarrow (V' \cup \{S_1, \ldots, S_r\})^*$$

el morfisme definit així

$$\begin{cases} f(X) = X & \text{si } X \in V' \\ f(a_i) = S_i & \text{si } a_i \in \Sigma'. \end{cases}$$

El que fem és $P = \{A \to f(\alpha) \mid A \to \alpha \in P'\} \cup P_1 \cup \cdots \cup P_r$.

Proposició. $\mathsf{L}(G) = \sigma(L')$.

Demostració. La demostració, que és força immediata i que segueix els passos de la corresponent als morfismes, es deixa com a exercici. □

Exemple 2.22 Reprenem aquí la substitució introduïda a l'exemple 1.11. Partim de la gramàtica següent que genera els arbres binaris

$$S \to 2SS \mid 0.$$

Considerem la substitució σ definida per

$$\sigma(0) = L_0 = 1^*0 \quad \text{i} \quad \sigma(2) = L_2 = 1^*2.$$

Una gramàtica per a L_0 és $X_0 \to 1X_0 \mid 0$, mentre que una gramàtica per a L_2 és $X_2 \to 1X_2 \mid 2$. Així doncs, una gramàtica per als arbres unari-binaris és

$$\begin{aligned} S &\to X_2SS \mid X_0 \\ X_0 &\to 1X_0 \mid 0 \\ X_2 &\to 1X_2 \mid 2. \end{aligned}$$

Deixem com a exercici per al lector demostrar que aquesta gramàtica és equivalent a la presentada a l'exemple 1.11, és a dir,

$$S \to 2SS \mid 1S \mid 0 .$$

Com a conseqüència de la proposició anterior podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de les substitucions incontextuals.*

És curiós d'observar que les propietats de tancament dels llenguatges incontextuals respecte de les operacions de reunió, concatenació i estrella s'haurien pogut deduir del tancament respecte de les substitucions. En efecte, donats dos llenguatges incontextuals, L_1 i L_2, només cal considerar un alfabet auxiliar de dos símbols, $\Sigma = \{a_1, a_2\}$, i fer la substitució $\sigma(a_1) = L_1$ i $\sigma(a_2) = L_2$. Resulta que $L_1 \cup L_2 = \sigma(\{a_1, a_2\})$, $L_1 \cdot L_2 = \sigma(\{a_1 a_2\})$ i $L_1^* = \sigma(a_1^*)$. Les propietats de tancament deriven del fet que $\{a_1, a_2\}$, $\{a_1 a_2\}$ i a_1^* són CFL.

2.6 La intersecció de dos CFL pot no ser CFL

A diferència del que hem vist amb l'operació de reunió, que conserva la incontextualitat, les operacions de complementació i d'intersecció no la conserven. Pot ser que la intersecció de dos CFL sigui també un CFL, però pot no ser-ho. És evident que la intersecció de dos CFL pot ser un CFL. N'hi ha prou de considerar el cas de dos CFL, L_1 i L_2, tals que $L_1 \subseteq L_2$, en què la intersecció és el mateix L_1. O el cas en què L_1 i L_2 són disjunts, ja que el conjunt buit és CFL (una gramàtica que tingui un conjunt buit de produccions genera el llenguatge buit). Però també hi ha casos en què la intersecció de dos CFL no és un CFL. L'exemple següent ho posa de manifest.

Exemple 2.23 Considerem els llenguatges:

$$L_1 = \{a^n b^n \mid n \geqslant 0\} c^* \quad \text{i} \quad L_2 = a^* \{b^n c^n \mid n \geqslant 0\}.$$

Es tracta de dos llenguatges incontextuals. L_1 és la concatenació del llenguatge definit a l'exemple 2.3 amb el llenguatge c^*, que té per gramàtica simplement $S \to cS \mid \lambda$. L_2 és la concatenació, en sentit invers, de dos llenguatges isomorfs als que componen L_1. Si ara considerem la intersecció d'aquests dos llenguatges obtenim

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geqslant 0\}.$$

Com veurem més endavant, en estudiar el lema d'Ogden al capítol 7, aquest llenguatge no és incontextual.

Així doncs, en aquest cas només podem enunciar el resultat negatiu següent.

TEOREMA. *La família dels llenguatges incontextuals no és tancada respecte de la intersecció.*

Tenint en compte que la intersecció és el complementari de la reunió de complementaris, i que la família dels CFL sí que és tancada respecte de la reunió, podem enunciar el corol·lari següent.

COROL·LARI. *La família dels llenguatges incontextuals no és tancada respecte de la complementació.*

Exemple 2.24 Al capítol 7 veurem que el llenguatge dels *quadrats* dels mots sobre l'alfabet $\{a, b\}$, és a dir, el llenguatge $L = \{ww \mid w \in \{a, b\}^*\}$ no és un CFL. En canvi, el seu complementari és generat per la gramàtica

$$\begin{aligned} S &\to AB \mid BA \mid A \mid B \\ A &\to TAT \mid a \\ B &\to TBT \mid b \\ T &\to a \mid b\,. \end{aligned}$$

En efecte, observeu que la variable A genera tots els mots de longitud senar que tenen una a al punt mig. Semblantment, la variable B genera mots amb una b central. Ara bé, els mots que no són de la forma ww o bé tenen longitud senar (i aleshores són generats a partir d'una producció inicial $S \to A \mid B$), o bé tenen una de les dues formes següents, $\{a, b\}^n a \{a, b\}^{n+m} b \{a, b\}^m$ o $\{a, b\}^n b \{a, b\}^{n+m} a \{a, b\}^m$, i aleshores són generats a partir de $S \to AB$ o de $S \to BA$, respectivament.

Cal remarcar, no obstant això, que hi ha CFL, fins i tot *no deterministes* (la definició de CFL determinista es donarà al capítol 8), que tenen per complementari també un CFL, com posa de manifest l'exemple que donem a continuació.

Exemple 2.25 Sigui $L_1 = \{w \in \{a, b\}^* \mid w = w^{\mathrm{R}}\}$ i considerem el seu complementari. Es tracta del llenguatge format pels mots que *no* són palíndroms. El representarem per L_2 i és el següent

$$L_2 = \{xa_1ya_2z \mid x, y, z \in \{a, b\}^* \wedge a_1, a_2 \in \{a, b\} \wedge |x| = |z| \wedge a_1 \neq a_2\}.$$

Tant L_1 com L_2 tenen gramàtiques que els generen. Són les següents:

$$L_1 : S \to aSa \mid bSb \mid a \mid b \mid \lambda$$

$$L_2 : \begin{cases} S \to aSa \mid bSb \mid aXb \mid bXa \\ X \to aX \mid bX \mid \lambda\,. \end{cases}$$

Exercicis

2.1 Trobeu gramàtiques incontextuals que generin els llenguatges següents:

1. $\{xcy \mid x, y \in \{a, b\}^* \wedge x^{\mathrm{R}} \text{ és submot de } y\}$
2. $\{w \in \{a, b\}^* \mid 2|w|_a = 3|w|_b + 2\}$
3. $\{w \in \{a, b\}^* \mid |w|_a < 2|w|_b + 1\}$

2.2 Donada la gramàtica $S \to aS \mid aSbS \mid \lambda$

1. Demostreu que genera els mots tals que qualsevol prefix té almenys tantes a's com b's.

2. Demostreu que és ambigua.
3. Trobeu una gramàtica equivalent no ambigua.

2.3 Donada la gramàtica de produccions

$$\begin{aligned} S &\to aBS \mid bAS \mid \lambda \\ A &\to bAA \mid a \\ B &\to aBB \mid b\,, \end{aligned}$$

1. Demostreu que genera el conjunt de mots sobre l'alfabet $\{a\,, b\}$ que tenen tantes a's com b's.
2. Demostreu que és inambigua.

2.4 Demostreu que les dues gramàtiques següents generen el mateix llenguatge:

$$G_1\colon \begin{cases} S \to A \mid B \\ A \to aAbA \mid c \\ B \to aS \mid aAbB \end{cases} \qquad \text{i} \qquad G_2\colon \begin{cases} S \to aS \mid aAbS \mid c \\ A \to aAbA \mid c\,. \end{cases}$$

2.5 Considereu l'equació $L = a^+ (\{\lambda\} \cup Lb)$. Trobeu una gramàtica per a l'únic llenguatge L que la satisfà. Quin és aquest llenguatge?

2.6 Una gramàtica s'anomena *lineal* quan les seves produccions són de la forma $X \to uYv$ o bé $X \to w$, on X i Y són variables i u, v i w són mots terminals. Un llenguatge s'anomena lineal quan hi ha alguna gramàtica lineal que el genera. Trobeu gramàtiques lineals per als llenguatges següents:

1. $\{a^i b^j \mid i < j\}$
2. $\{a^i b^j \mid i < 2j\}$
3. $\{a^i b^j \mid i > 2j\}$
4. $\{a^i b^j \mid j < i < 2j\}$
5. $\{a^i b^j \mid i = j \ \vee \ i = 2j\}$
6. $\{a^i b^j \mid i \neq j \ \wedge \ i \neq 2j\}$
7. $\{a^i b^j c^k \mid i = j \ \vee \ j = k\}$
8. $\{a^i b^j c^k \mid i \neq j \ \vee \ j \neq k\}$
9. $\{a^i b^j c^k \mid k = i + j\}$
10. El complementari sobre l'alfabet $\{a, b, c\}$ del conjunt $\{a^n b^n c^n \mid n > 0\}$
11. $\{ww^{\mathsf{R}} \mid w \in \{a\,, b\}^*\}$
12. $\{a^i b^j 1 c^i \mid i, j \geqslant 0\} \cup \{a^i b^j 2 c^j \mid i, j \geqslant 0\}$
13. $\{a^i b^j c^k \mid k = i \ \vee \ k = j\}$
14. $\{a^i b^j c^k d^l \mid k = i \ \vee \ l = j\}$

2.7 Una gramàtica s'anomena *metalineal* quan les seves produccions són d'una de les formes següents:

$$\begin{aligned} &S \to A_1 \ldots A_m \\ &A \to uBv \\ &A \to u\,, \end{aligned}$$

on A és una variable qualsevol, A_i i B poden ser qualsevol variable que no sigui S, i u i v són mots terminals. Un llenguatge s'anomena metalineal quan hi ha alguna gramàtica metalineal que el genera. Demostreu que un llenguatge és metalineal si i sols si és reunió finita de concatenacions de llenguatges lineals.

2.8 Trobeu gramàtiques metalineals per als llenguatges següents:

1. $\{a^i b^i c^j d^j \mid i, j \geqslant 0\}$
2. $\{a^i b^j c^k d^l \mid (i = j \wedge k = l) \ \vee \ (i = l \wedge j = k)\}$

3. $\{xy \mid x, y \in \{a, b\}^* \wedge |x| = |y| \wedge x \neq y\}$

2.9 Els llenguatges següents són CFL no metalineals. Trobeu gramàtiques que els generin.

1. $\{a^i b^j c^j b^k c^k a^i \mid i, j, k \geqslant 0\}$
2. $\{a^n b^n \mid n \geqslant 0\}^*$
3. $\{wcw^{\mathsf{R}}c \mid w \in \{a, b\}^+\}^*$
4. $\{a^i b^j c^k d^l \mid i + k = j + l\}$

2.10 Construïu gramàtiques que generin els llenguatges definits a continuació.

1. $\{w \in \{a, b\}^* \mid w = w^{\mathsf{R}} \wedge \exists x, y \; w = xabay\}$
2. $\{w \in a^+b^+c^+ \mid |w| = \dot{2} \wedge |w|_a \geqslant |w|_b\}$
3. $\{w \in \{a, b\}^* \mid w = w^{\mathsf{R}} \wedge |w| = \dot{4}\}$
4. $\{a^i b^j c^k \mid j = i + k \wedge j \neq \dot{3}\}$

2.11 Trobeu gramàtiques que generin els llenguatges següents:

1. $\{ab^{i_1}ab^{i_2}\ldots ab^{i_n} \mid n \geqslant 1 \wedge \exists j \; i_j = j\}$
2. $\{ab^{i_1}ab^{i_2}\ldots ab^{i_n} \mid n \geqslant 1 \wedge i_n = n\}$
3. $\{b^n ab^{i_1}ab^{i_2}\ldots ab^{i_n} \mid n \geqslant 0\}$
4. $\{ab^{i_1}ab^{i_2}\ldots ab^{i_n} \mid n \geqslant 1 \wedge \exists j \; i_j = n\}$

2.12 Siguin L_1 i L_2 dos CFL. Demostreu que

$$\bigcup_{n \geqslant 1} (L_1)^n (L_2)^n$$

és també un CFL.

2.13 Demostreu que tot CFL infinit és generat per alguna gramàtica en què totes les variables generen llenguatges infinits. És a dir que existeix una gramàtica $\langle V, \Sigma, P, S\rangle$ que genera el llenguatge i tal que

$$\forall X \in V \quad \{w \in \Sigma^* \mid X \stackrel{*}{\Rightarrow} w\} \text{ és infinit.}$$

2.14 Demostreu que la reunió de dos CFL no ambigus que siguin disjunts és un CFL no ambigu.

2.15 Siguin $A_n = \{a_1, \ldots, a_n\}$ i $B_n = \{b_1, \ldots, b_n\}$ dos alfabets disjunts del mateix nombre n de símbols. Anomenem *llenguatge de Dyck* de n parells de símbols, i el representem per D_n^*, el conjunt de mots generats per la gramàtica que té per produccions

$$S \to \lambda \mid SS \mid a_i S b_i \mid b_i S a_i \quad \text{per a cada } i \;\; 1 \leqslant i \leqslant n.$$

Es tracta de mots que presenten ben aparionats els símbols dels alfabets A_n i B_n, de manera que no hi ha imbricacions encavalcades; així, el mot $b_1a_2b_2a_1$ és un mot ben construït; en canvi, el mot $a_1a_2b_1b_2$ no ho és.

1. Trobeu una gramàtica no ambigua que generi D_2^*. (Observeu que una gramàtica no ambigua per a D_1^* és la definida a l'exercici 2.3.)

Anomenem *llenguatge de Dyck restringit* de n parells de símbols, i el representem per ${\mathsf{D}'_n}^*$, el conjunt de mots generats per la gramàtica que té per produccions

$$S \to \lambda \mid SS \mid a_i S b_i \quad \text{per a cada } i \; 1 \leqslant i \leqslant n.$$

Es tracta d'interpretar els n parells (a_i, b_i) com a diferents parells de parèntesis en correspondència, i prendre en consideració els mots ben parentitzats.

2. Trobeu gramàtiques no ambigües que generin ${\mathsf{D}'_n}^*$ per a cada n.

Representem per D_n i D'_n els conjunts de factors primers[5] de D_n^* i ${\mathsf{D}'_n}^*$, respectivament.

[5]Podeu trobar la definició de factors primers a l'exercici 1.18.

3. Demostreu que D_n i D'_n són codis.[6]
4. Trobeu gramàtiques que generin D_1 i D'_1.
5. Demostreu que, per a qualsevol valor de n, existeix un morfisme h_n de la forma $h_n\colon (A_n \cup B_n)^* \longrightarrow (A_2 \cup B_2)^*$ que permet obtenir els diferents llenguatges de Dyck de n parells de símbols com a antiimatges dels corresponents de dos parells de símbols, és a dir,
$$\mathsf{D}_n^* = h_n^{-1}(\mathsf{D}_2^*), \quad {\mathsf{D}'_n}^* = h_n^{-1}({\mathsf{D}'_2}^*), \quad \mathsf{D}_n = h_n^{-1}(\mathsf{D}_2) \quad \text{i} \quad \mathsf{D}'_n = h_n^{-1}(\mathsf{D}'_2).$$

[6]Podeu trobar la definició de codi a l'exercici 1.20.

Capítol 3 Normalització de gramàtiques

3.1 Introducció

La definició de les gramàtiques incontextuals, tal com ha estat efectuada al capítol precedent, ofereix a la vegada uns avantatges i uns inconvenients que convé considerar. La definició donada no posa cap restricció a la forma que han de tenir les produccions de les gramàtiques. Qualsevol cadena sobre l'alfabet $\Sigma \cup V$, on Σ i V siguin, respectivament, els alfabets de terminals i variables, pot figurar al costat dret d'una producció. L'avantatge principal d'aquesta formulació és l'adaptabilitat: es facilita la tasca de trobar una gramàtica que generi un llenguatge donat sense trobar-se condicionat per altres requeriments que els posats pel problema considerat. Gràcies a això, és possible construir, en molts casos, gramàtiques que expressen de manera clara i directa les característiques del llenguatge que generen.

Però al costat d'aquests avantatges, la falta de restriccions en les gramàtiques és origen de nombrosos problemes en la seva utilització. Així, per exemple, l'admissió de produccions *unàries* en les gramàtiques (és a dir, produccions de la forma $X \to Y$, on X i Y són variables), admissió que, d'altra banda, és força útil a l'hora de construir una gramàtica, pot originar derivacions *cícliques*, és a dir, de la forma $X \Rightarrow \cdots \Rightarrow X$, que donen lloc a problemes amb els algorismes que han de processar les gramàtiques. Una reflexió similar mereixen les produccions *nul·les* o *λ-produccions* (produccions de la forma $X \to \lambda$).

Diem que una gramàtica està en alguna forma *normalitzada* quan les seves variables i produccions estan subjectes a normes addicionals sense que aquests requeriments limitin la família dels llenguatges generats. Al llarg d'aquest capítol, veurem diferents formes de normalitzar les gramàtiques i, en cada cas, explicarem els objectius que es persegueixen amb la normalització considerada. Abans veurem, tanmateix, un exemple que il·lustra el que estem dient.

Exemple 3.1 Considerem el llenguatge format pel conjunt dels mots sobre l'alfabet $\{a, b\}$ que tenen tantes a's com b's. Una de les gramàtiques més fàcils de plantejar entre les que generen aquest llenguatge és

$$S \to aSb \mid bSa \mid SS \mid \lambda.$$

En aquest capítol definirem la *forma normal de Chomsky*, que té la propietat que tots els seus arbres de derivació són binaris. Doncs bé, la conversió de la gramàtica anterior a una en forma normal de Chomsky dóna el resultat següent:

$$\begin{aligned}
&S \to AX \mid BY \mid ZZ \mid AB \mid BA \mid \lambda\\
&Z \to AX \mid BY \mid ZZ \mid AB \mid BA\\
&X \to ZB\\
&Y \to ZA\\
&A \to a\\
&B \to b.
\end{aligned}$$

Es tracta, com és palès, d'una gramàtica molt menys comprensible, tot i els avantatges que presenta amb vista a la seva utilització.

3.2 Eliminació de produccions nul·les

DEFINICIÓ. S'anomenen *produccions nul·les* o *λ-produccions* les que són de la forma $X \to \lambda$.

Les λ-produccions constitueixen un recurs molt útil amb vista a facilitar la concepció d'una gramàtica. Més endavant, al capítol 6, veurem que el seu ús permet fer més immediata la relació entre *gramàtiques regulars* i *autòmats finits*. Però també hem pogut veure, a l'exemple 2.13 del capítol anterior, que l'existència de λ-produccions posa dificultats a la verificació d'una gramàtica. El fet és que, si una gramàtica està exempta de λ-produccions, la successió de longituds de les cadenes intermèdies de qualsevol derivació és *no decreixent*. És a dir, si α i β són dues cadenes intermèdies i $\alpha \stackrel{*}{\Rightarrow} \beta$, llavors $|\alpha| \leqslant |\beta|$. Això sol ser essencial en qualsevol demostració per inducció.

És clar que no sempre és possible construir una CFG sense λ-produccions que sigui equivalent a una gramàtica donada, perquè una CFG sense λ-produccions no pot generar el mot λ. Però com veurem de seguida, aquesta és l'única diferència possible. És a dir que per a tot llenguatge incontextual L existeix una gramàtica G' sense λ-produccions que genera $L - \{\lambda\}$. La construcció d'una gramàtica sense λ-produccions G' a partir d'una gramàtica G qualsevol, de manera que es tingui $\mathsf{L}(G') = \mathsf{L}(G) - \{\lambda\}$, es basa en la determinació prèvia del que anomenem variables *anul·lables* de G.

DEFINICIÓ. Una variable X s'anomena *anul·lable* quan existeix una derivació de la forma $X \stackrel{*}{\Rightarrow} \lambda$.

L'algorisme de la figura 3.1 té per entrades els alfabets V i Σ de variables i terminals, respectivament, i el conjunt P de produccions d'una gramàtica qualsevol. Demostrarem que dóna com a sortida el subconjunt N de variables anul·lables de V. L'algorisme fa un càlcul progressiu d'aquest conjunt N, que s'inicia amb les variables X que tenen produccions $X \to \lambda$ i que va incrementant a cada pas el nombre dels seus elements, fins que ja no hi ha nous elements a afegir.

PROPOSICIÓ. *Els dos enunciats següents són equivalents:*

1. *Hi ha una derivació $X \stackrel{*}{\Rightarrow} \lambda$ l'arbre de la qual té alçària*[1] $n+1$.

[1] Entenem per *alçària* d'un arbre la longitud (definida pel nombre d'arestes components) màxima dels camins que van de l'arrel de l'arbre a cada una de les fulles. Convé tenir present que alguns autors compten el nombre de vèrtexs i no el d'arestes com a definidor d'aquesta longitud.

```
funció anul·lables ( V: conjunt de caràcters;
                     Σ: conjunt de caràcters;
                     P: conjunt de produccions)
                     retorna (N: conjunt de caràcters)
    variables N_ant : conjunt de caràcters fvars
    N_ant := ∅;
    N := {X ∈ V | X → λ ∈ P};
    mentre N_ant ≠ N fer
        N_ant := N;
        N := N_ant ∪ {X ∈ V | ∃α ∈ N*_ant  X → α ∈ P}
    fmentre;
    retorna N
ffunció.
```

Fig. 3.1 *Càlcul del conjunt de variables anul·lables*

2. *A partir de l'n-èsima iteració del bucle* **mentre** *de l'algorisme* 'anul·lables', *es té* $X \in N$.

DEMOSTRACIÓ. La demostració és una senzilla inducció sobre n. □

Observeu que X és anul·lable si i sols si existeix algun n que satisfà la condició 1. I que $X \in N$ si i sols si existeix algun n que satisfà la condició 2. Així doncs, la proposició anterior, en establir l'equivalència entre les condicions 1 i 2, demostra que N és efectivament el conjunt de variables anul·lables de la gramàtica considerada.

Un procediment d'eliminació de produccions nul·les

Estem ara en condicions de construir una gramàtica sense λ-produccions a partir d'una gramàtica donada. Sigui $G = \langle V, \Sigma, P, S \rangle$ una gramàtica qualsevol. La nova gramàtica és de la forma $G' = \langle V, \Sigma, P', S \rangle$, amb les mateixes variables i terminals que G i on el conjunt P' s'obté a partir de P en eliminar totes les λ-produccions i afegir-hi a continuació noves produccions, seguint el procediment següent. Per cada producció $A \to \chi_1 \dots \chi_n \in P$, on les χ_i representen indistintament variables o terminals, formem a partir d'aquesta producció totes les produccions possibles de la forma

$$A \to \alpha_1 \dots \alpha_n$$

tals que

1. si χ_i no és una variable anul·lable, $\alpha_i = \chi_i$,
2. si χ_i és una variable anul·lable, α_i pot ser χ_i o λ,
3. no totes les α_i són λ al mateix temps.

Exemple 3.2 Considerem de nou el llenguatge format pels mots sobre l'alfabet $\{a, b\}$

que tenen tantes a's com b's. La següent és una gramàtica *no ambigua* que genera aquest llenguatge.

$$\begin{aligned} S &\to aXbS \mid bYaS \mid \lambda \\ X &\to aXbX \mid \lambda \\ Y &\to bYaY \mid \lambda. \end{aligned}$$

El conjunt de variables anul·lables d'aquesta gramàtica és $\{S, X, Y\}$. La gramàtica que s'obté a partir de la donada quan apliquem el procediment anterior d'eliminació de λ-produccions és aquesta

$$\begin{aligned} S &\to aXbS \mid aXb \mid abS \mid ab \mid bYaS \mid bYa \mid baS \mid ba \\ X &\to aXbX \mid aXb \mid abX \mid ab \\ Y &\to bYaY \mid bYa \mid baY \mid ba. \end{aligned}$$

Proposició. $\mathsf{L}(G') = \mathsf{L}(G) - \{\lambda\}$.

Demostració. Es tracta de demostrar l'enunciat següent, del qual la proposició n'és un cas particular quan l'apliquem a la variable $A := S$.

$$\forall A \in V \ \ \forall w \in \Sigma^* \quad \left(A \underset{G'}{\overset{*}{\Rightarrow}} w\right) \iff \left(w \neq \lambda \ \wedge \ A \underset{G}{\overset{*}{\Rightarrow}} w\right).$$

$(\Rightarrow)$ En primer lloc, és clar que w no pot ser λ, ja que la gramàtica G' no conté cap λ-producció. L'altra implicació la veurem per inducció sobre el nombre i de passos de la derivació de la gramàtica G'

$$\left(A \underset{G'}{\overset{i}{\Rightarrow}} w\right) \implies \left(A \underset{G}{\overset{*}{\Rightarrow}} w\right).$$

- *Base:* $i = 1$. Si tenim $A \underset{G'}{\Rightarrow} w$, hem de tenir una producció $A \to w$ en P'. Si aquesta producció és a P, ja hem acabat; altrament, n'hi ha una altra $A \to \alpha$ on α és w amb algunes variables anul·lables barrejades entre les seves lletres. És obvi ara com derivar w de A en G.
- *Pas inductiu:* $i > 1$. Sigui $A \underset{G'}{\overset{i}{\Rightarrow}} w$ amb derivació

$$A \underset{G'}{\Rightarrow} \alpha_1 \cdots \alpha_r \underset{G'}{\overset{i-1}{\Rightarrow}} w.$$

Sigui w_k el submot de w derivat de α_k per a $k \leqslant r$. Per la hipòtesi d'inducció, tindrem que $\forall k \leqslant r \ \ \alpha_k \underset{G}{\overset{*}{\Rightarrow}} w_k$. Si la producció $A \to \alpha_1 \ldots \alpha_r$ pertany a P, ja podem formar una derivació de w en G.

Ara bé, si la producció $A \to \alpha_1 \ldots \alpha_r$ no pertany a P, aleshores existeix una producció en P de la forma $A \to \chi_1 \ldots \chi_s$, amb $s > r$, i un conjunt $J = \{j_1 < \cdots < j_r\} \subsetneq \{1, \ldots, s\}$ tal que $\chi_{j_k} = \alpha_k$, i si $j \notin J$, χ_j és anul·lable. Considerant finalment la derivació

$$A \underset{G}{\Rightarrow} \chi_1 \ldots \chi_s \underset{G}{\overset{*}{\Rightarrow}} w_1 \ldots w_r = w$$

on les χ_j amb $j \notin J$ han derivat el mot buit, es prova el que volíem.

($\Leftarrow$) Fem la demostració també per inducció sobre el nombre i de passos de la derivació de la gramàtica G

$$\left(w \neq \lambda \ \wedge \ A \underset{G}{\overset{i}{\Rightarrow}} w\right) \implies \left(A \underset{G'}{\overset{*}{\Rightarrow}} w\right).$$

- *Base:* $i = 1$.

$$\left.\begin{array}{c} A \underset{G}{\Rightarrow} w \\ w \neq \lambda \end{array}\right\} \implies \left\{\begin{array}{c} A \to w \in P \\ w \neq \lambda \end{array}\right\} \implies A \to w \in P' \implies A \underset{G'}{\overset{*}{\Rightarrow}} w.$$

- *Pas inductiu:* $i > 1$. Sigui $A \underset{G}{\overset{i}{\Rightarrow}} w$ amb $w \neq \lambda$. Explicitem el primer pas d'aquesta derivació per separat dels $i - 1$ restants. Sigui

$$A \underset{G}{\Rightarrow} \chi_1 \dots \chi_r \underset{G}{\overset{i-1}{\Rightarrow}} w.$$

Siguin $w_1, \dots, w_r$ tals que

$$\begin{cases} w = w_1 \dots w_r \\ \forall k \quad \chi_k \underset{G}{\overset{*}{\Rightarrow}} w_k \quad \text{(en menys de } i \text{ passos).} \end{cases}$$

- Si $w_k \neq \lambda$ i χ_k és una variable, tenim, per h.i., $\chi_k \underset{G'}{\overset{*}{\Rightarrow}} w_k$.
- Si $w_k = \lambda$, és que χ_k és una variable anul·lable.

Així doncs, i com que no totes les w_k poden ser λ perquè $w \neq \lambda$, ha d'haver-hi a P', per construcció, una producció $A \to \alpha_1 \dots \alpha_r$ on

$$\begin{cases} \alpha_k = \chi_k & \text{si } w_k \neq \lambda \\ \alpha_k = \lambda & \text{si } w_k = \lambda. \end{cases}$$

Per tant, a G' hi ha una derivació $A \underset{G'}{\Rightarrow} \alpha_1 \dots \alpha_r \underset{G'}{\overset{*}{\Rightarrow}} w_1 \dots w_r$. $\square$

Complexitat de l'eliminació de produccions nul·les

El procediment que acabem de descriure d'eliminació de les λ-produccions pot augmentar exponencialment la magnitud de la gramàtica. Ho veurem immediatament amb un exemple. Primer, però, necessitem partir d'una definició concreta de *grandària* d'una gramàtica. Podem considerar la següent, on P representa el conjunt de produccions d'una gramàtica G,

$$\sum_{(A \to \alpha) \in P} |A\alpha|.$$

Es tracta de prendre com a grandària el nombre total de símbols que apareixen a esquerra i dreta de totes les produccions de la gramàtica.

Considerem ara un exemple senzill de gramàtica que servirà per posar de manifest que l'eliminació de λ-produccions, quan es fa seguint el procediment donat a la subsecció precedent, pot conduir a un augment exponencial de la grandària de la gramàtica resultant. Sigui la gramàtica formada per les $k + 1$ produccions següents

$$\begin{cases} S \to A_1 \dots A_k \\ A_i \to a_i \mid \lambda \qquad 1 \leqslant i \leqslant k\,. \end{cases}$$

La grandària d'aquesta gramàtica és $4k+1$. L'eliminació de λ-produccions manté les k produccions $A_i \rightarrow a_i$ i transforma la primera producció en

- $\binom{k}{0}$ produccions amb k variables a la dreta,
- $\binom{k}{1}$ produccions amb $k-1$ variables a la dreta,

$\vdots$

- $\binom{k}{k-1}$ produccions amb 1 variable a la dreta.

La longitud total dels costats drets d'aquestes produccions és

$$\sum_{i=0}^{k-1} \binom{k}{i}(k-i) = k \cdot 2^{k-1},$$

que, afegida a les $2^k - 1$ vegades que hem de comptar el símbol S i als $2k$ símbols que aporten les k produccions que es mantenen, dóna com a grandària de la gramàtica resultant la següent:

$$(k+2) \cdot 2^{k-1} + 2k - 1.$$

Observeu que aquest resultat implica una complexitat exponencial del procediment proposat, ja que la mera escriptura del resultat ja comporta un nombre exponencial de passos respecte de la longitud de l'entrada.

Podem millorar substancialment aquesta complexitat amb una lleugera modificació de l'algorisme. Es tracta d'aplicar una idea que tornarà a aparèixer més endavant, en aquest mateix capítol, en estudiar la *forma normal de Chomsky*. Consisteix a transformar les produccions de la gramàtica donada de manera que tinguin *arietat* limitada, abans d'aplicar el procediment d'eliminació de λ-produccions. Entenem aquí per arietat d'una producció la longitud del seu costat dret. En el cas concret de fer una transformació a arietat menor o igual que dos, i referint-nos a la mateixa gramàtica que ha servit de base a aquesta anàlisi, es tracta de substituir la producció $S \rightarrow A_1 \dots A_k$ pel conjunt de $k-1$ produccions

$$\begin{aligned} S &\rightarrow A_1 Z_1 \\ Z_1 &\rightarrow A_2 Z_2 \\ &\vdots \\ Z_{k-3} &\rightarrow A_{k-2} Z_{k-2} \\ Z_{k-2} &\rightarrow A_{k-1} A_k \end{aligned}$$

i aplicar a continuació el procediment d'eliminació de produccions nul·les. El resultat és la gramàtica següent

$$\begin{array}{rlcrl} S \rightarrow & A_1 Z_1 \mid A_1 \mid Z_1 & \qquad & A_1 \rightarrow & a_1 \\ Z_1 \rightarrow & A_2 Z_2 \mid A_2 \mid Z_2 & & A_2 \rightarrow & a_2 \\ \vdots & & & \vdots & \\ Z_{k-2} \rightarrow & A_{k-1} A_k \mid A_{k-1} \mid A_k & & A_k \rightarrow & a_k\,, \end{array}$$

gramàtica que té una grandària de $9k-7$. És fàcil de demostrar que aquest algorisme es pot aplicar a qualsevol gramàtica i que la seva complexitat és lineal.

Gramàtiques quasi-λ-exemptes

L'eliminació de les produccions nul·les dóna lloc a gramàtiques que exclouen el mot buit del llenguatge generat. En aquesta secció definirem una forma normal de gramàtica que permetrà superar aquest inconvenient.

DEFINICIÓ. Diem que una gramàtica és *quasi-λ-exempta* quan o bé no conté cap λ-producció, o bé l'única que conté és $S \to \lambda$ —essent S el símbol inicial— en el qual cas S no apareix al costat dret de cap producció.

TEOREMA. *Tot llenguatge incontextual pot ser generat per una gramàtica quasi-λ-exempta.*

DEMOSTRACIÓ. Sigui L un CFL qualsevol. Si $\lambda \notin L$, l'aplicació de l'algorisme d'eliminació de les λ-produccions a qualsevol gramàtica que generi L dóna una gramàtica λ-exempta que genera el mateix llenguatge L.

Si $\lambda \in L$, considerem una CFG qualsevol que generi L. Sigui $G = \langle V, \Sigma, P, S\rangle$ la gramàtica que genera $L - \{\lambda\}$, construïda a partir d'aquella mitjançant l'algorisme d'eliminació de les produccions nul·les. Definim a partir de G la gramàtica G' següent

$$G' = \langle V \cup \{S'\}, \Sigma, P', S'\rangle$$

on $S' \notin V$ i on P' s'ha format afegint a P les dues produccions $S' \to \lambda$ i $S' \to S$. És clar que $\mathsf{L}(G') = L$ i que G' és quasi-λ-exempta. □

Exemple 3.3 Considerem la gramàtica que té per símbol inicial S i per produccions

$$\begin{aligned} S &\to aSa \mid bSb \mid D \mid AB \mid \lambda \\ A &\to aaA \mid aa \\ B &\to Bbb \mid b \\ D &\to bD \mid Db \\ E &\to cDc \mid FS \\ F &\to DF \mid cE \mid \lambda. \end{aligned}$$

L'algorisme 'anul·lables' troba S i F abans de passar pel bucle, E a la primera passada i cap altra variable a la segona. Així doncs, el conjunt de variables anul·lables és $\{S, E, F\}$.

L'eliminació de les λ-produccions, la seva substitució per les produccions noves i l'afegiment d'una nova variable inicial S' que generi altra vegada λ dóna lloc al conjunt de produccions següent:

$$\begin{aligned} S' &\to S \mid \lambda \\ S &\to aSa \mid aa \mid bSb \mid bb \mid D \mid AB \\ A &\to aaA \mid aa \\ B &\to Bbb \mid b \\ D &\to bD \mid Db \\ E &\to cDc \mid FS \mid F \mid S \\ F &\to DF \mid D \mid cE \mid c. \end{aligned}$$

3.3 Eliminació de produccions unàries

DEFINICIÓ. S'anomenen *produccions unàries*[2] les que són de la forma $A \to B$, on A i B són variables.

Amb l'ús de les produccions unàries passa una cosa semblant al que passa amb l'ús de les produccions nul·les. D'una banda, són molt útils a l'hora de construir gramàtiques senzilles per a llenguatges donats. Pot observar-se aquest fet en els exemples 2.1, 2.10, 2.11, 2.15, 2.17, 2.20, 2.22 i 2.24 del capítol anterior, on se n'ha fet un ús extens. També en les construccions associades a les operacions de reunió i de tancament positiu. Però, d'altra banda, compliquen els processos demostratius. Un cas extrem és la possibilitat que en una gramàtica es puguin donar simultàniament dues derivacions simètriques de la forma $A \overset{*}{\Rightarrow} B$ i $B \overset{*}{\Rightarrow} A$ entre variables A i B diferents. La inexistència de produccions unàries impedeix que això succeeixi quan la gramàtica és λ-exempta. En una gramàtica sense produccions nul·les ni produccions unàries, es pot associar una relació d'ordre estricte a cada pas d'una derivació, ja que si $\alpha \Rightarrow \beta$, llavors o bé $|\alpha| < |\beta|$ o bé, si $|\alpha| = |\beta|$, el nombre de terminals de β és superior al de α.

Donarem la manera de construir una gramàtica sense produccions unàries que generi un CFL L donat. En virtut de l'últim teorema de la secció anterior, podem partir d'una CFG quasi-λ-exempta per a aquest llenguatge L. Sigui $G = \langle V, \Sigma, P, S \rangle$ aquesta gramàtica.

Per a cada $A \in V$, considerem el conjunt de variables que es deriven de A, és a dir,

$$\mathsf{unaris}(A) = \{B \in V \mid A \underset{G}{\overset{*}{\Rightarrow}} B\}.$$

Aquest conjunt es pot construir de manera efectiva per a cada $A \in V$, ja que si $A \overset{+}{\Rightarrow} B$ aleshores existeix una derivació de B a partir de A formada per una seqüència de produccions unàries en què no es repeteix cap variable. La inexistència de λ-produccions —tret, eventualment, de $(S \to \lambda)$— força que totes les produccions que intervenen a la seqüència hagin de ser unàries, altrament la cadena derivada seria de longitud més gran que 1. A més, la llargada d'aquesta seqüència ha de ser inferior o igual a $\|V\|$, altrament hi hauria alguna variable que es repetiria i és clar que podríem escurçar aquesta derivació eliminant els passos intermedis entre dues aparicions consecutives d'aquesta variable.

Sigui $G' = \langle V, \Sigma, P', S \rangle$ on P' està formada, per a cada $A \in V$ i per a cada $B \in \mathsf{unaris}(A)$, per totes les produccions $A \to \alpha$ tals que $B \to \alpha$ sigui una producció no unària de P, incloent-hi, eventualment, $S \to \lambda$. Formalment,

$$P' = \bigcup_{A \in V} \{A \to \alpha \mid \exists B \in \mathsf{unaris}(A) \quad (B \to \alpha) \in P \ \wedge \ \alpha \notin V\}.$$

És clar que G' es manté quasi-λ-exempta i que no té produccions unàries. De seguida veurem que genera el mateix llenguatge, però abans il·lustrem amb un exemple la construcció donada.

Exemple 3.4 Partim de la gramàtica quasi-λ-exempta trobada a l'exemple 3.3. El seu conjunt de produccions era

[2] En anglès, *unit productions* o també *chain rules*.

$$\begin{aligned}
S' &\to S \mid \lambda \\
S &\to aSa \mid aa \mid bSb \mid bb \mid D \mid AB \\
A &\to aaA \mid aa \\
B &\to Bbb \mid b \\
D &\to bD \mid Db \\
E &\to cDc \mid FS \mid F \mid S \\
F &\to DF \mid D \mid cE \mid c\,.
\end{aligned}$$

Per a aquesta gramàtica podem calcular els conjunts següents:

$$\begin{array}{ll}
\mathsf{unaris}\,(S') = \{S', S, D\} & \mathsf{unaris}\,(D) = \{D\} \\
\mathsf{unaris}\,(S) = \{S, D\} & \mathsf{unaris}\,(E) = \{E, F, S, D\} \\
\mathsf{unaris}\,(A) = \{A\} & \mathsf{unaris}\,(F) = \{D, F\}\,. \\
\mathsf{unaris}\,(B) = \{B\} &
\end{array}$$

Aplicant ara el mètode exposat anteriorment, trobem les produccions

$$\begin{aligned}
S' &\to \lambda \mid aSa \mid aa \mid bSb \mid bb \mid AB \mid bD \mid Db \\
S &\to aSa \mid aa \mid bSb \mid bb \mid AB \mid bD \mid Db \\
A &\to aaA \mid aa \\
B &\to Bbb \mid b \\
D &\to bD \mid Db \\
E &\to cDc \mid FS \mid DF \mid cE \mid c \mid aSa \mid aa \mid bSb \mid bb \mid AB \mid bD \mid Db \\
F &\to DF \mid cE \mid c \mid bD \mid Db
\end{aligned}$$

que constitueixen una gramàtica quasi-λ-exempta i sense produccions unàries, equivalent a la de partida.

Proposició. $\mathsf{L}(G') = \mathsf{L}(G)$.

Demostració.

1. $\mathsf{L}(G') \subseteq \mathsf{L}(G)$.

Per cada producció $A \to \alpha$ de $P' - P$, tenim, per la construcció de P', una derivació $A \overset{*}{\Rightarrow} \alpha$ en G.

2. $\mathsf{L}(G) \subseteq \mathsf{L}(G')$.

Sigui $w \in \mathsf{L}(G)$ i considerem una derivació de w

$$S = \alpha_0 \underset{G}{\Rightarrow} \alpha_1 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} \alpha_n = w,$$

tal que en cada derivació directa la variable a la qual ha estat aplicada la producció és la de més a l'esquerra possible. Direm que una derivació directa és *unària* quan la producció que hi intervé és unària. Si, per a tot i tal que $0 \leqslant i < n$, la derivació directa $\alpha_i \Rightarrow \alpha_{i+1}$ és no unària, aleshores tota la derivació és vàlida en G'.

Si, al contrari, $\alpha_i \Rightarrow \alpha_{i+1}$ és una derivació unària de G, podem suposar que $\alpha_i \Rightarrow \alpha_{i+1} \Rightarrow \cdots \Rightarrow \alpha_j$ és una successió de derivacions unàries, i que $\alpha_{i-1} \Rightarrow \alpha_i$ i $\alpha_j \Rightarrow \alpha_{j+1}$ no ho són pas. En aquest cas, hem de tenir

$$\exists x \in \Sigma^* \ \ \exists \beta \in (V \cup \Sigma)^* \ \ \forall k \text{ tal que } i \leqslant k \leqslant j-1$$
$$\alpha_k = x X_k \beta \ \wedge \ \alpha_{k+1} = x X_{k+1} \beta \ \wedge \ (X_k \to X_{k+1}) \in P.$$

Sigui $X_j \to \gamma$ la producció utilitzada en $\alpha_j \Rightarrow \alpha_{j+1}$. Com que $X_j \in \mathsf{unaris}(X_i)$, existeix una producció $X_i \to \gamma$ en P' amb la qual obtenim la derivació

$$S = \alpha_0 \underset{G'}{\overset{*}{\Rightarrow}} \alpha_i \underset{G'}{\Rightarrow} \alpha_{j+1}.$$

Continuant d'aquesta manera, obtenim una derivació de w en G'. □

Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *Tot llenguatge incontextual pot ser generat per una gramàtica quasi-λ-exempta i sense produccions unàries.*

3.4 Eliminació de símbols inútils

DEFINICIÓ. Donada una gramàtica $G = \langle V, \Sigma, P, S \rangle$ i un símbol $\chi \in V \cup \Sigma$, direm que χ és *útil* si existeix un mot $w \in \Sigma^*$ i una derivació de la forma $S \overset{*}{\Rightarrow} \alpha\chi\beta \overset{*}{\Rightarrow} w$, on $\alpha, \beta \in (V \cup \Sigma)^*$. Altrament direm que χ és *inútil*.

De la definició es dedueix que perquè χ sigui útil és necessari que existeixin dues derivacions

$$S \overset{*}{\Rightarrow} \alpha\chi\beta \qquad \text{i} \qquad \chi \overset{*}{\Rightarrow} x \quad \text{amb } x \in \Sigma^*.$$

Aquesta condició, però, no és suficient, com posa de manifest l'exemple següent:

Exemple 3.5 En la gramàtica

$$\begin{aligned} &S \to AB \mid CD \\ &A \to AS \\ &B \to b \\ &C \to Cc \mid \lambda \\ &D \to d\,, \end{aligned}$$

la variable B satisfà totes dues condicions, ja que es té $S \overset{*}{\Rightarrow} AB$ i $B \overset{*}{\Rightarrow} b$. En canvi es tracta d'un símbol inútil.

DEFINICIÓ. Anomenem *fecunds* els símbols per als quals existeix la segona de les dues derivacions, i *accessibles* aquells per als quals n'existeix la primera.

Eliminació de símbols no fecunds

L'algorisme de la figura 3.2 té com a entrades els alfabets V i Σ de variables i terminals, respectivament, i el conjunt P de produccions d'una gramàtica qualsevol. Pot observar-se que la seva estructura és similar a la de l'algorisme 'anul·lables' de la figura 3.1. Demostrarem que dóna com a sortida el subconjunt F de símbols fecunds de G. Remarquem, de passada, que aquest mateix algorisme pot utilitzar-se per resoldre el *problema de la*

funció fecunds (V: conjunt de caràcters;
Σ: conjunt de caràcters;
P: conjunt de produccions)
retorna (F: conjunt de caràcters)
variables F_{ant} : conjunt de caràcters **fvars**
$F_{\mathrm{ant}} := \varnothing$;
$F := \Sigma$;
mentre $F \neq F_{\mathrm{ant}}$ **fer**
$F_{\mathrm{ant}} := F$;
$F := F_{\mathrm{ant}} \cup \{X \in V \mid \exists \alpha \in F^*_{\mathrm{ant}} \quad X \to \alpha \in P\}$
fmentre;
retorna F
ffunció.

Fig. 3.2 *Càlcul del conjunt de símbols fecunds*

buidor[3] referit a CFG, és a dir, el problema de determinar si el llenguatge generat per una gramàtica donada és o no és el conjunt buit. En efecte, el llenguatge generat per una gramàtica és buit si i sols si el seu símbol inicial no és fecund.

PROPOSICIÓ. *Els dos enunciats següents són equivalents:*

1. *Hi ha una derivació* $X \overset{*}{\Rightarrow} w$ *amb* $w \in \Sigma^*$ *l'arbre de la qual té alçària* n.
2. *A partir de l'*n*-èsima iteració del bucle* **mentre** *de l'algorisme* 'fecunds' *es té* $X \in F$.

DEMOSTRACIÓ. La demostració, idèntica a la de l'algorisme de variables anul·lables, es redueix a una inducció sobre n. □

Donada una CFG $G = \langle V, \Sigma, P, S\rangle$, amb $\mathsf{L}(G) \neq \varnothing$, podem construir-ne una altra d'equivalent a G amb tots els seus símbols —variables i terminals— fecunds. En efecte, sigui F el conjunt de símbols fecunds de G obtingut per aplicació de l'algorisme anterior. Fem $G' = \langle V', \Sigma, P', S\rangle$, on $V' = V \cap F$, i on P' és el subconjunt de P format per les produccions en què només apareixen variables de V'.

Exemple 3.6 Reprenem la gramàtica obtinguda a l'exemple 3.4, que era

$$G = \langle \{S', S, A, B, D, E, F\}, \{a, b, c\}, P, S'\rangle,$$

amb produccions

[3] En anglès, *emptiness problem*.

$$
\begin{aligned}
S' &\to \lambda \mid aSa \mid aa \mid bSb \mid bb \mid AB \mid bD \mid Db\\
S &\to aSa \mid aa \mid bSb \mid bb \mid AB \mid bD \mid Db\\
A &\to aaA \mid aa\\
B &\to Bbb \mid b\\
D &\to bD \mid Db\\
E &\to cDc \mid FS \mid DF \mid cE \mid c \mid aSa \mid aa \mid bSb \mid bb \mid AB \mid bD \mid Db\\
F &\to DF \mid cE \mid c \mid bD \mid Db.
\end{aligned}
$$

L'algorisme 'fecunds' parteix del conjunt $\{a, b, c\}$ i hi incorpora les variables S', S, A, B, E i F en el primer pas del bucle. En el segon pas, no n'afegeix cap més. Així doncs, només la variable D és no fecunda. En eliminar totes les produccions que contenen D ens quedem amb

$$
\begin{aligned}
S' &\to \lambda \mid aSa \mid aa \mid bSb \mid bb \mid AB\\
S &\to aSa \mid aa \mid bSb \mid bb \mid AB\\
A &\to aaA \mid aa\\
B &\to Bbb \mid b\\
E &\to FS \mid cE \mid c \mid aSa \mid aa \mid bSb \mid bb \mid AB\\
F &\to cE \mid c.
\end{aligned}
$$

Proposició. $\mathsf{L}(G') = \mathsf{L}(G)$.

Demostració.

1. $\mathsf{L}(G') \subseteq \mathsf{L}(G)$.

Trivial, ja que tota derivació en G' és també una derivació en G.

2. $\mathsf{L}(G) \subseteq \mathsf{L}(G')$.

Sigui F el conjunt de símbols fecunds de G. Si $w \in \mathsf{L}(G)$, considerem una derivació de w en G. Sigui $S \overset{*}{\underset{G}{\Rightarrow}} w$. Per a qualsevol variable A que aparegui en aquesta derivació es tindrà que es pot derivar de A un mot de Σ^*: el submot de w que "penja" de A en l'arbre de derivació que correspon a $S \overset{*}{\underset{G}{\Rightarrow}} w$. Així doncs, $A \in F$. Per tant, la derivació completa és també una derivació en G'. □

Eliminació de símbols no accessibles

L'algorisme de la figura 3.3 té com a entrades els alfabets V i Σ de variables i terminals, respectivament, el conjunt P de produccions i el símbol inicial S d'una gramàtica G qualsevol. La seva estructura és similar a la dels algorismes anteriors, amb la diferència que el càlcul progressa ara "cap endavant", des del símbol inicial cap als terminals. Demostrarem que dóna com a sortida el subconjunt A de símbols accessibles de G.

Proposició. *Els dos enunciats següents són equivalents:*

1. *Hi ha una derivació $S \overset{*}{\Rightarrow} \alpha\chi\beta$, on $\chi \in V \cup \Sigma$, l'arbre de la qual té alçària n.*
2. *A partir de l'n-èsima iteració del bucle* **mentre** *de l'algorisme* 'accessibles' *es té $\chi \in A$.*

funció accessibles (V: conjunt de caràcters;
Σ: conjunt de caràcters;
P: conjunt de produccions;
S: caràcter)
retorna (A: conjunt de caràcters)
variables A_{ant} : conjunt de caràcters **fvars**
$A_{\text{ant}} := \varnothing$;
$A := \{S\}$;
mentre $A_{\text{ant}} \neq A$ **fer**
$A_{\text{ant}} := A$;
$A := A_{\text{ant}} \cup \{\chi \in V \cup \Sigma \mid \exists X \in A_{\text{ant}}\ \exists \alpha, \beta \in (V \cup \Sigma)^* \ \ X \to \alpha\chi\beta \in P\}$
fmentre;
retorna A
ffunció.

Fig. 3.3 *Càlcul del conjunt de símbols accessibles*

DEMOSTRACIÓ. La demostració, idèntica a la feta en el cas dels algorismes 'anul·lables' i 'fecunds', torna a fer-se per inducció sobre n. □

Donada una CFG $G = \langle V, \Sigma, P, S\rangle$ qualsevol, podem construir-ne una altra d'equivalent a G amb tots els seus símbols —variables i terminals— accessibles. En efecte, sigui A el conjunt de símbols accessibles de G obtingut per aplicació de l'algorisme anterior. Fem $G' = \langle V', \Sigma', P', S\rangle$, on $V' = V \cap A$, $\Sigma' = \Sigma \cap A$ i on P' és el subconjunt de P format per les produccions en què només apareixen símbols de A.

Exemple 3.7 Ens proposem construir una gramàtica equivalent a la de l'exemple 3.6, però que tingui tots els símbols accessibles. Recordem que la gramàtica de què parlem, que era quasi-λ-exempta i que no tenia produccions unàries ni símbols no fecunds, era

$$
\begin{aligned}
S' &\to \lambda \mid aSa \mid aa \mid bSb \mid bb \mid AB \\
S &\to aSa \mid aa \mid bSb \mid bb \mid AB \\
A &\to aaA \mid aa \\
B &\to Bbb \mid b \\
E &\to FS \mid cE \mid c \mid aSa \mid aa \mid bSb \mid bb \mid AB \\
F &\to cE \mid c\,.
\end{aligned}
$$

L'aplicació de l'algorisme 'accessibles' en aquesta gramàtica parteix del símbol S' en la inicialització del conjunt de símbols accessibles. Hi incorpora els símbols a, b, S, A i B en el primer pas pel bucle i cap símbol nou en el segon. Així, c, E i F són inaccessibles. En suprimir les produccions que contenen algun d'aquests símbols, ens quedem amb el conjunt P' de produccions

$$\begin{aligned} S' &\rightarrow \lambda \mid aSa \mid aa \mid bSb \mid bb \mid AB \\ S &\rightarrow aSa \mid aa \mid bSb \mid bb \mid AB \\ A &\rightarrow aaA \mid aa \\ B &\rightarrow Bbb \mid b\,, \end{aligned}$$

la qual cosa ens dóna finalment la gramàtica depurada

$$G' = \langle \{S', S, A, B\}, \{a, b\}, P', S' \rangle.$$

PROPOSICIÓ. $\mathsf{L}(G') = \mathsf{L}(G)$.

DEMOSTRACIÓ.

1. $\mathsf{L}(G') \subseteq \mathsf{L}(G)$.

Trivial, ja que tota derivació en G' és també una derivació en G.

2. $\mathsf{L}(G) \subseteq \mathsf{L}(G')$.

Sigui A el conjunt de símbols accessibles de G. Si $w \in \mathsf{L}(G)$, considerem una derivació de w en G. Sigui $S \underset{G}{\overset{*}{\Rightarrow}} w$. Per a tots els símbols $\chi \in V \cup \Sigma$ que apareixen en aquesta derivació es té que χ és accessible des de S. Així doncs, $\chi \in A$, i la derivació, tota sencera, és també una derivació en G'. $\square$

Eliminació de símbols inútils

L'eliminació consecutiva primer dels símbols no fecunds i a continuació dels símbols no accessibles dóna com a resultat, quan es fa en l'ordre indicat, una gramàtica sense símbols inútils, com demostrem a continuació. Però cal tenir en compte que és essencial fer-ho en aquest ordre i no a l'inrevés, com posa de manifest l'exemple següent.

Exemple 3.8 Considerem la gramàtica que té per variables $V = \{S, A, B\}$, per terminals $\Sigma = \{a\,, b\}$ i per produccions P les següents:

$$\begin{aligned} S &\rightarrow AB \mid a \\ A &\rightarrow BA \\ B &\rightarrow b. \end{aligned}$$

Primer pas: eliminació de símbols no fecunds. Obtenim $V' = \{S, B\}$, $\Sigma = \{a\,, b\}$ i P' igual a

$$\begin{aligned} S &\rightarrow a \\ B &\rightarrow b. \end{aligned}$$

Segon pas: eliminació de símbols no accessibles. Obtenim $V'' = \{S\}$, $\Sigma'' = \{a\}$ i P'' igual a

$$S \rightarrow a.$$

En canvi, si s'hagués fet el segon pas abans del primer, el resultat hauria estat el següent.

Primer pas: eliminació de símbols no accessibles. En aquest cas, no obtindríem cap canvi en la gramàtica de partida ja que el resultat seria $V' = \{S, A, B\}$, $\Sigma' = \{a\,, b\}$ i P' estaria format per

$$
\begin{aligned}
S &\to AB \mid a \\
A &\to BA \\
B &\to b.
\end{aligned}
$$

Segon pas: eliminació de símbols no fecunds. Obtindríem $V'' = \{S, B\}$, $\Sigma'' = \{a, b\}$ i P'' igual a

$$
\begin{aligned}
S &\to a \\
B &\to b.
\end{aligned}
$$

TEOREMA. *Tot llenguatge incontextual no buit pot ser generat per una gramàtica sense símbols inútils.*

DEMOSTRACIÓ. Sigui $G = \langle V, \Sigma, P, S\rangle$ la gramàtica que genera el llenguatge no buit donat. Sigui $G' = \langle V', \Sigma, P', S\rangle$ la gramàtica que s'obté a partir de G per eliminació dels símbols no fecunds. Sigui $G'' = \langle V'', \Sigma'', P'', S\rangle$ la gramàtica que s'obté a partir de G' per eliminació dels símbols no accessibles. En virtut de les proposicions d'aquesta mateixa secció que demostren la conservació dels llenguatges afectats, tenim successivament $\mathsf{L}(G') = \mathsf{L}(G)$ i $\mathsf{L}(G'') = \mathsf{L}(G')$. En conseqüència, tenim

$$\mathsf{L}(G'') = \mathsf{L}(G).$$

Només ens queda veure que G'' no té símbols inútils. Per a això, considerem un símbol $\chi \in V'' \cup \Sigma''$ de G''. Com que G'' és el resultat d'eliminar els símbols no accessibles de G', χ és accessible en G''. Per tant, existeix una derivació de la forma

$$S \underset{G''}{\overset{*}{\Rightarrow}} \alpha\chi\beta$$

amb $\alpha, \beta \in (V'' \cup \Sigma'')^*$.

Per construcció de G'', aquesta derivació ho és també en G'. Però en G', tots els símbols són fecunds. Podem, doncs, derivar mots terminals de les variables de $\alpha\chi\beta$ i obtenir la derivació

$$S \underset{G'}{\overset{*}{\Rightarrow}} \alpha\chi\beta \underset{G'}{\overset{*}{\Rightarrow}} w$$

on w és un mot terminal.

Finalment, tots els símbols que apareixen en totes les cadenes que intervenen en aquesta derivació són accessibles (ja que la derivació comença amb S), per tant, la construcció feta per obtenir G'' els preservarà, i tota la derivació és vàlida en G'', és a dir, tenim

$$S \underset{G''}{\overset{*}{\Rightarrow}} \alpha\chi\beta \underset{G''}{\overset{*}{\Rightarrow}} w.$$

Això demostra que χ no és inútil. $\square$

3.5 Gramàtiques depurades

TEOREMA. *Tot llenguatge incontextual no buit pot ser generat per una gramàtica quasi-λ-exempta, sense produccions unàries i sense símbols inútils.*

DEMOSTRACIÓ. Es tracta de considerar l'aplicació successiva, i en aquest ordre precís, dels processos de transformació d'una gramàtica donada en una altra quasi-λ-exempta; d'aquesta en una altra sense produccions unàries; i d'aquesta en una altra sense símbols inútils. Els únics punts a remarcar són els següents:

1. L'eliminació de produccions unàries pot introduir símbols inútils, però en cap cas no pot introduir λ-produccions.
2. L'eliminació dels símbols inútils no pot introduir en cap cas ni λ-produccions ni produccions unàries.

Observeu, incidentalment, que l'eliminació de λ-produccions pot introduir produccions unàries, raó per la qual ha de ser el primer dels processos a efectuar.

Com que ja ha estat demostrat que totes les transformacions considerades produeixen gramàtiques equivalents, la gramàtica obtinguda en l'últim pas satisfà els requeriments del teorema. □

Una caracterització de la no-ambigüitat en CFG

El fet de partir d'una gramàtica depurada ens posa en millors condicions a l'hora de determinar si la gramàtica considerada és ambigua. El teorema següent estableix unes condicions que caracteritzen la no-ambigüitat.

TEOREMA. *Una gramàtica $G = \langle V, \Sigma, P, S\rangle$ sense símbols inútils és inambigua si i només si se satisfan les condicions següents:*

1. *Per a tot parell de produccions $X \to \alpha_1$ i $X \to \alpha_2$ de P diferents, es té $\mathsf{L}(\alpha_1) \cap \mathsf{L}(\alpha_2) = \varnothing$.*
2. *Per a tota producció $X \to \chi_1 \ldots \chi_n$ de P, on els $\chi_i \in V \cup \Sigma$, si existeixen mots $w_1, w_1' \in \mathsf{L}(\chi_1)$, ... , $w_n, w_n' \in \mathsf{L}(\chi_n)$ tals que $w_1 \ldots w_n = w_1' \ldots w_n'$, llavors necessàriament $w_1 = w_1'$, ... , $w_n = w_n'$.*

DEMOSTRACIÓ. És obvi que les dues condicions són necessàries, ja que de no complir-se una d'elles, com que la variable X afectada és útil, obtindríem directament un cas d'ambigüitat.

Suposem que les dues condicions es compleixen i mirem de demostrar que G és no ambigua; això ho fem per mínim contraexemple (equivalent a inducció).

Suposem que G és ambigua. Ha d'existir una varible X i dos arbres de derivació diferents d'un mateix mot $w \in \Sigma^*$ a partir de X. Considerem un contraexemple format per un parell d'arbres (t, t') que tenen entre tots dos un nombre mínim de vèrtexs. Sigui X l'arrel de t i de t' (que no té per què ser S). Per la condició 1, el primer pas de la derivació en t i en t' coincideix; i per tant és de la forma $X \to \chi_1 \ldots \chi_n$, amb els $\chi_i \in V \cup \Sigma$. Llavors t i t' tenen n fills directes $t_1, \ldots, t_n$ i $t_1', \ldots, t_n'$, respectivament, on cada χ_i és el símbol de l'arrel dels corresponents t_i i t_i'. Si w_i i w_i' són, respectivament, els mots generats pels corresponents t_i i t_i', llavors $w = w_1 \ldots w_n = w_1' \ldots w_n'$ i, per la condició 2, tenim que cada w_i coincideix amb w_i'. Donat que t i t' són arbres diferents, existeix algun j entre 1 i n tal que t_j i t_j' són arbres diferents. Però com que $w_j = w_j'$, tenim que χ_j és una variable i llavors (t_j, t_j') és un contraexemple menor que (t, t'), contradicció. □

Exemple 3.9 Considerem la gramàtica $S \to aSbS \mid \lambda$. Suposem que ja hem demostrat que aquesta gramàtica genera el llenguatge

$$\{w \in \{a, b\}^* \mid (|w|_a = |w|_b) \wedge (\forall x, y\ w = xy \implies |x|_a \geqslant |x|_b)\}.$$

Ara volem veure que la gramàtica no és ambigua. Clarament satisfà la condició 1. La condició 2 se satisfà trivialment per a la producció $S \to \lambda$. Considerem doncs la producció $S \to aSbS$; siguin $w_1, w_1', w_2, w_2', w \in \mathsf{L}(S)$ tals que $aw_1bw_2 = aw_1'bw_2' = w$. Considerem el primer prefix de w diferent de λ i que té tantes a's com b's; aquest prefix existeix, ja que en particular w té tantes a's com b's. És fàcil justificar, per les propietats del llenguatge $\mathsf{L}(S)$, que aquest prefix és tant aw_1b com $aw_1'b$; per tant $aw_1b = aw_1'b$ i llavors $w_1 = w_1'$ i $w_2 = w_2'$.

El problema de la pertinença en CFL

S'anomena *problema de la pertinença*,[4] referit a una certa família de llenguatges, el problema de determinar, donats un mot qualsevol i un llenguatge qualsevol de la família en qüestió, si el mot pertany o no al llenguatge.

En el cas de la família dels CFL, la possibilitat de construir gramàtiques depurades per a qualsevol llenguatge de la família permet donar un algorisme per resoldre el problema de la pertinença. Aquest algorisme es basa en el fet que podem fitar superiorment el nombre de passos que requereix qualsevol derivació en una gramàtica depurada. La proposició següent permet concretar aquesta idea de forma precisa.

PROPOSICIÓ. *Sigui $G = \langle V, \Sigma, P, S \rangle$ una* CFG *quasi-λ-exempta i sense produccions unàries. Sigui $w \in \mathsf{L}(G) - \{\lambda\}$. Aleshores, tota derivació de w en G té una llargària de $2|w| - 1$ passos, com a màxim.*

DEMOSTRACIÓ. Farem la demostració en general per a una variable X qualsevol. És a dir, demostrarem que tota derivació de la forma $X \overset{*}{\Rightarrow} w$ està composta per $2|w| - 1$ passos, com a màxim. Ho farem per inducció sobre $n = |w|$.

Base: $n = 1$
En aquest cas, la derivació $X \overset{*}{\Rightarrow} w$ ha de ser de la forma $X \Rightarrow a$, d'un sol pas, amb $a \in \Sigma$. Es compleix exactament la fita.

Pas inductiu: $n > 1$
Sigui

$$X \Rightarrow \chi_1 \ldots \chi_r \overset{*}{\Rightarrow} w$$

una derivació de w. Es té que r ha de ser més gran que 1. Altrament tindríem $X \Rightarrow \chi_1 \overset{*}{\Rightarrow} w$, i χ_1 hauria de ser un terminal, ja que no hi ha produccions unàries a G. Però aleshores la talla de w seria 1.

Sigui $w = w_1 \ldots w_r$, on cada w_i deriva del seu χ_i corresponent. (En el cas que χ_i sigui un terminal, aquesta derivació té 0 passos i es té $w_i = \chi_i$.) Com que, per a cada i, tenim $1 \leqslant |w_i| < n$, resulta, per hipòtesi d'inducció, que les derivacions

$$\chi_i \overset{*}{\Rightarrow} w_i$$

[4] En anglès, *membership problem*.

tenen llargària més petita o igual que $2|w_i| - 1$.

La llargària de
$$S \Rightarrow \chi_1 \ldots \chi_r \stackrel{*}{\Rightarrow} w_1 \ldots w_r = w$$
està fitada, doncs, per
$$1 + \sum_{i=1}^{r}(2|w_i| - 1) = 1 + \sum_{i=1}^{r} 2|w_i| - r = 1 + 2|w| - r \leqslant 1 + 2|w| - 2 = 2|w| - 1.$$
□

Estem ara en condicions de demostrar l'afirmació amb què hem començat aquesta subsecció.

TEOREMA. *Sigui $L \subseteq \Sigma^*$ un* CFL *i $w \in \Sigma^*$. Aleshores, podem decidir si $w \in L$.*

DEMOSTRACIÓ. Sigui G una CFG quasi-λ-exempta i sense produccions unàries que generi L. Si $w = \lambda$, w pertany a L si i sols si existeix a G la producció $S \to \lambda$, on S és la variable inicial de G.

Si $w \neq \lambda$, construïm totes les possibles derivacions en G de nombre de passos inferior a $2|w|$. Segons la proposició anterior, $w \in L$ si i sols si w resulta generat en alguna d'aquestes derivacions. □

Convé advertir que hi ha algorismes per resoldre el problema de la pertinença en CFL que són més eficients que l'exposat al teorema anterior, el qual té un cost exponencial. L'objectiu d'aquest teorema és sols posar de manifest la *decidibilitat* del problema de la pertinença en el cas dels CFL, en contrast amb el que passa amb altres famílies de llenguatges com, per exemple, els definits per *màquines de Turing*.[5]

3.6 Forma normal de Chomsky

DEFINICIÓ. Diem que una gramàtica quasi-λ-exempta està en *forma normal de Chomsky* (abreviadament, CNF[6]) quan totes les seves produccions —a excepció eventualment de $S \to \lambda$— són o bé de la forma $A \to BC$ o bé de la forma $A \to a$, on A, B i C són variables i a és un terminal.

La utilitat principal de la forma normal de Chomsky és que permet una millora de l'eficiència dels algorismes que operen amb gramàtiques. Un exemple d'això ha estat vist en aquest mateix capítol en parlar de la complexitat de l'eliminació de les produccions nul·les. És interessant observar que la conversió d'una gramàtica qualsevol a CNF és un procés anàleg al de la implementació d'un *arbre general* mitjançant un *arbre binari*. Pot trobar-se aquesta implementació a la secció 5.2.2 de [Fra93].

TEOREMA. *Tot* CFL *pot ser generat per una gramàtica en forma normal de Chomsky.*

DEMOSTRACIÓ. Podem partir d'una gramàtica $G = \langle V, \Sigma, P, S \rangle$ que generi el llenguatge considerat, que sigui quasi-λ-exempta i que no tingui ni produccions unàries ni símbols inútils. Fem la construcció d'una gramàtica en CNF equivalent a G en dues etapes.

[5] Trobareu la definició d'aquest model de computació al capítol 10.
[6] De l'anglès *Chomsky Normal Form*.

En una primera etapa, considerem una producció qualsevol de la forma $A \to \chi_1 \dots \chi_m$, on cada χ_i pot ser una variable o un terminal. Si $m = 1$, com que G no té produccions unàries, aquesta ha de ser del tipus $A \to a$, que ja està en CNF.

Suposem, doncs, que $m \geqslant 2$. Considerem tots els símbols terminals que apareixen en el conjunt de produccions d'aquest tipus. Per cada un d'aquests terminals $a \in \Sigma$ definim una nova variable C_a i afegim la producció $C_a \to a$.

Reemplacem cada producció de la forma $A \to \chi_1 \dots \chi_m$, en què no totes les χ_i siguin variables, per una producció de la mateixa forma en què cada terminal $\chi_i = a$ ha estat substituït per la variable C_a corresponent.

Totes les produccions —a part de $S \to \lambda$— són ara de la forma

$$\begin{cases} A \to a & \text{o bé} \\ A \to X_1 \dots X_m\,, & \text{amb } m \geqslant 2. \end{cases}$$

on totes les $X_1 \dots X_m$ són variables.

Sigui $G' = \langle V', \Sigma, P, S \rangle$ la nova gramàtica amb les modificacions prèvies. És fàcil de demostrar que $\mathsf{L}(G') = \mathsf{L}(G)$.

En una segona etapa, considerem la gramàtica $G'' = \langle V'', \Sigma, P'', S \rangle$ obtinguda de la manera següent. Per cada producció de G' del tipus $A \to B_1 \dots B_m$ amb $m > 2$, definim noves variables $D_1, \dots, D_{m-2}$ i substituïm la producció $A \to B_1 \dots B_m$ pel conjunt de produccions

$$\begin{aligned} A &\to B_1 D_1 \\ D_1 &\to B_2 D_2 \\ &\vdots \\ D_{m-3} &\to B_{m-2} D_{m-2} \\ D_{m-2} &\to B_{m-1} B_m. \end{aligned}$$

És clar que la gramàtica G'' obtinguda està en CNF, i de nou no presenta cap dificultat demostrar que $\mathsf{L}(G'') = \mathsf{L}(G')$. □

Exemple 3.10 Considerem la gramàtica depurada obtinguda a l'exemple 3.7, és a dir, la gramàtica $G = \langle \{S', S, A, B\}, \{a, b\}, P, S' \rangle$, on P és el conjunt de produccions

$$\begin{aligned} S' &\to \lambda \mid aSa \mid aa \mid bSb \mid bb \mid AB \\ S &\to aSa \mid aa \mid bSb \mid bb \mid AB \\ A &\to aaA \mid aa \\ B &\to Bbb \mid b\,. \end{aligned}$$

La primera etapa del mètode anterior ens dóna les produccions

$$\begin{aligned} S' &\to \lambda \mid C_aSC_a \mid C_aC_a \mid C_bSC_b \mid C_bC_b \mid AB \\ S &\to C_aSC_a \mid C_aC_a \mid C_bSC_b \mid C_bC_b \mid AB \\ A &\to C_aC_aA \mid C_aC_a \\ B &\to BC_bC_b \mid b \\ C_a &\to a \\ C_b &\to b\,. \end{aligned}$$

La segona etapa del mètode dóna les produccions

$$\begin{aligned}
S' &\to \lambda \mid C_aX \mid C_aC_a \mid C_bT \mid C_bC_b \mid AB \\
S &\to C_aX \mid C_aC_a \mid C_bT \mid C_bC_b \mid AB \\
X &\to SC_a \\
T &\to SC_b \\
A &\to C_aY \mid C_aC_a \\
Y &\to C_aA \\
B &\to BZ \mid b \\
Z &\to C_bC_b \\
C_a &\to a \\
C_b &\to b\,.
\end{aligned}$$

De fet, aquest no és exactament el resultat que hauríem obtingut si haguéssim seguit el mètode al peu de la lletra, ja que hauríem introduït les variables X' i T' diferenciades de les X i T per a les dues produccions ternàries que depenen de S'. És clar, tanmateix, que es tracta, en aquest cas, de parells de variables equivalents.

Exercicis

3.1 Construïu una gramàtica sense λ-produccions que sigui equivalent a la següent:

$$\begin{aligned}
S &\to aSa \mid bSb \mid aXb \mid bXa \\
X &\to aX \mid bX \mid \lambda\,.
\end{aligned}$$

3.2 Construïu una gramàtica quasi-λ-exempta que sigui equivalent a la següent:

$$\begin{aligned}
S &\to XY \\
X &\to aXb \mid \lambda \\
Y &\to bYc \mid \lambda\,.
\end{aligned}$$

3.3 Construïu una gramàtica sense produccions unàries que sigui equivalent a la següent:

$$\begin{aligned}
S &\to X \mid X+S \\
X &\to T \mid Y \times Z \\
Y &\to T \mid (X+S) \\
Z &\to Y \mid Y \times Z \\
T &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\,.
\end{aligned}$$

3.4 Construïu una gramàtica sense produccions unàries que sigui equivalent a la següent:

$$\begin{aligned}
S &\to A \mid B \\
A &\to aA \mid aAb \mid a \\
B &\to Bb \mid aBb \mid b\,.
\end{aligned}$$

3.5 Construïu una gramàtica quasi-λ-exempta i sense produccions unàries que sigui equivalent a la següent:

$$\begin{aligned}
S &\to aBS \mid bAS \mid \lambda \\
A &\to bAA \mid a \\
B &\to aBB \mid b\,.
\end{aligned}$$

3.6 Demostreu que totes les transformacions de gramàtiques exposades en aquest capítol preserven la no-ambigüitat de la gramàtica original.

3.7 Trobeu un algorisme per determinar si un CFL qualsevol, definit mitjançant alguna CFG, és finit o infinit.

3.8 Una gramàtica quasi-λ-exempta es diu que està en *forma normal de Greibach* quan totes les seves produccions, a excepció eventualment de $S \rightarrow \lambda$, són de la forma $A \rightarrow a\alpha$, on a és un símbol terminal i α és una cadena intermèdia que o bé és buida o bé està composta exclusivament de variables. Trobeu un procediment per transformar una CFG qualsevol en una altra en forma normal de Greibach.

3.9 Trobeu el nombre de passos de qualsevol derivació d'un mot en funció de la longitud del mot, per a cada un dels casos següents:

1. Quan la gramàtica està en forma normal de Chomsky.
2. Quan la gramàtica està en forma normal de Greibach.

3.10 Una gramàtica quasi-λ-exempta es diu que és una *gramàtica d'operadors* quan cap de les seves produccions no conté dues variables adjacents al seu costat dret. Trobeu un procediment per transformar una CFG qualsevol en una gramàtica d'operadors.

3.11 Trobeu un procediment per transformar una CFG qualsevol en una gramàtica quasi-λ-exempta en què totes les produccions, a excepció eventualment de $S \rightarrow \lambda$, són d'una de les tres formes següents: $A \rightarrow a$, $A \rightarrow aB$ i $A \rightarrow aBC$.

Capítol 4 Autòmats finits

4.1 Introducció

Considerem el clàssic problema de cercar una seqüència de caràcters, que se sol anomenar 'patró', en un text. Partim d'un text t (que suposem relativament llarg respecte del patró) escrit sobre un cert alfabet Σ, i d'un patró p, que sol ser una paraula o un prefix d'alguna paraula, i volem trobar si aquest patró apareix en algun lloc del text.

```
funció cerca ( text: vector [1..n] de caràcters;
               patró: vector [1..m] de caràcters;
               n, m: naturals)
               retorna (trobat: booleà)
    variables i, j: naturals fvars
    trobat := fals;
    i := 0;
    mentre (i ⩽ n−m ∧ no trobat) fer
        j := 1;
        mentre (text[i + j] = patró[j] ∧ j < m) fer
            j := j + 1
        fmentre;
        trobat := (text[i + j] = patró[j]);
        i := i + 1
    fmentre;
    retorna trobat
ffunció.
```

Fig. 4.1 *Cerca d'un patró en un text*

Un algorisme "ingenu", que es limita a cercar la primera ocurrència del patró, és el que apareix a la figura 4.1. Es tracta de situar-se en una posició del text i acarar un a un els símbols successius del patró amb els corresponents del text a partir de la posició considerada, interrompent l'acarament a la primera discrepància i passant a la posició següent del text. El procés finalitza així que es troba la primera ocurrència del patró.

Una fita superior del nombre de comparacions entre parells de símbols que efectua aquest algorisme és $n(m+1)/2$, on n i m són les longituds de *'text'* i *'patró'*, respectivament. Hem qualificat l'algorisme d'ingenu perquè no treu cap partit de la informació que va recollint a mesura que explora el text. Veurem com pot millorar-se substancialment el temps de procés utilitzant un autòmat finit.

Per centrar-nos en un cas concret, suposem $\Sigma = \{a\,, b\}$ i $p = abaa$. La figura 4.2 mostra el graf d'un autòmat finit determinista de quatre estats que *accepta* exactament els textos que contenen aquest patró. L'autòmat engega en l'estat q_0 i va llegint el text t, d'esquerra a dreta, canviant d'estat segons li ho indica el símbol que llegeix a cada instant. Si el text t conté el patró p com a submot, l'autòmat entra en l'estat q_4, que serà anomenat estat d'acceptació, just quan acaba de llegir per primera vegada aquest submot.

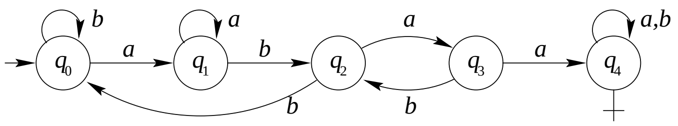

Fig. 4.2 *Autòmat finit que accepta mots que contenen abaa*

És fàcil demostrar que, qualsevol que sigui el patró p considerat, si fem $m = |p|$, existeix un autòmat finit determinista de $m + 1$ estats que efectua el procés de cerca. Aquest autòmat executa exactament n passos per llegir un text de longitud n. Així doncs, la feina total de construir l'autòmat a partir del patró i processar a continuació el text amb aquest autòmat requereix $\Theta(m + n)$ passos, mentre que l'algorisme ingenu en requeria $\Theta(m \cdot n)$.

4.2 Autòmats finits deterministes

Un *autòmat finit determinista* (abreujadament un DFA, de l'anglès *deterministic finite automaton*) és una estructura de la forma

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

els components de la qual es defineixen a continuació:

- Q és un conjunt finit no buit, els elements del qual s'anomenen *estats*.
- Σ és un alfabet (anomenat *d'entrada*).
- δ és la *funció de transició* que definirem tot seguit.
- $q_0 \in Q$ s'anomena estat *inicial*.
- $F \subseteq Q$ s'anomena conjunt d'estats *acceptadors*.

Quant a la funció de transició, és una aplicació de la forma

$$\delta\colon Q \times \Sigma^* \longrightarrow Q,$$

que representem amb notació multiplicativa,

$$\forall q \in Q \quad \forall w \in \Sigma^* \qquad \delta(q, w) = q \cdot w$$

i que satisfà els dos axiomes següents:

$$\forall q \in Q \quad \forall x, y \in \Sigma^* \qquad \begin{cases} (1) & q \cdot \lambda = q \\ (2) & q \cdot (xy) = (q \cdot x) \cdot y. \end{cases}$$

A partir d'aquests axiomes és immediat comprovar que qualsevol funció de transició queda completament definida especificant-ne solament els valors sobre el conjunt finit $Q \times \Sigma$. En efecte, si $x = a_1 a_2 ... a_n$, on tots els a_i són símbols de Σ, tenim per aplicació reiterada de l'axioma 2

$$q \cdot a_1 a_2 ... a_n = ((q \cdot a_1) \cdot a_2) \cdots a_n,$$

és a dir que el càlcul de la transició d'un estat per un mot de longitud n es resol amb n càlculs elementals consecutius de la forma "un estat per un símbol". Es pot especificar la funció de transició del DFA M mitjançant una taula, anomenada *taula de transicions*, que té com a entrades els estats de Q i els símbols de Σ.

Exemple 4.1 La taula 4.1 especifica les transicions d'un autòmat finit de vuit estats $\{q_0, q_1, \ldots q_7\}$, que té q_0 com a estat inicial, q_2 com a únic estat acceptador i els símbols a i b com a alfabet d'entrada.

Taula 4.1 *Taula de transicions d'un* DFA

	a	b
q_0	q_1	q_0
q_1	q_2	q_3
† q_2	q_2	q_2
q_3	q_4	q_5
q_4	q_2	q_6
q_5	q_1	q_1
q_6	q_2	q_7
q_7	q_4	q_4

Amb vista a facilitar la comparació entre autòmats, se sol adoptar el criteri d'escriure l'estat inicial a la primera línia i d'anar introduint els altres estats a les línies successives a mesura que van apareixent referenciats a la mateixa taula. Utilitzem una creu (†) per assenyalar els estats acceptadors.

Una altra manera de descriure un autòmat finit, que en facilita molt la comprensió en el cas d'autòmats petits, és mitjançant el que s'anomena *diagrama de transicions* de l'autòmat. Es tracta d'un graf dirigit que té per vèrtexs els estats de l'autòmat. Per cada transició de la forma $q_i \cdot a = q_j$ hi ha un arc que va de q_i a q_j i porta l'etiqueta a. Si hi ha més d'una transició entre dos estats q_i i q_j, per exemple $q_i \cdot a = q_j$ i $q_i \cdot b = q_j$, es dibuixa un únic arc entre aquests dos estats, l'etiqueta del qual inclou els símbols implicats separats per comes (a, b en aquest exemple). L'estat inicial se sol identificar amb una fletxeta que hi incideix. Els estats acceptadors se solen identificar amb una creueta (o amb un doble cercle en alguns textos).

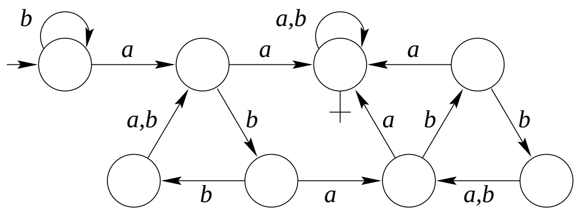

Fig. 4.3 *Diagrama de transicions de l'autòmat de la taula 4.1*

Exemple 4.2 La figura 4.3 il·lustra el diagrama de transicions de l'autòmat finit especificat a la taula 4.1.

Interpretació d'un DFA com un mecanisme

Si bé és imprescindible partir d'una descripció formal del concepte de DFA per tal de poder establir-ne rigorosament les propietats, també és necessari donar una *interpretació* del DFA que permeti la seva utilització en informàtica. Al capdavall, és d'aquest ús pràctic que se n'ha derivat el concepte com a model abstracte de computació.

Nosaltres interpretem els autòmats com unes màquines del tipus dels ordinadors, si bé d'una senzillesa estructural extrema. En el cas dels DFA que ara ens ocupa, aquestes màquines estan reduïdes a una unitat de control i a un capçal que llegeix una cinta d'entrada. La unitat de control consta d'una memòria finita. De fet, interpretem cada estat de l'autòmat com un contingut possible de la memòria. Així, una memòria de n bits de capacitat pot adoptar 2^n estats diferents. La cinta d'entrada està dividida en cel·les, cada una de les quals pot contenir un símbol de l'alfabet d'entrada. La longitud de la cinta es limita en cada cas al nombre de les cel·les que contenen el mot que ha de ser processat. Inicialment, el capçal es troba situat davant del primer símbol de l'esquerra del mot que està escrit a la cinta d'entrada. L'autòmat funciona com una màquina síncrona, actuant en temps discret com ho faria sota els impulsos d'un rellotge. A cada pas, el capçal es mou una posició cap a la cel·la de la dreta. Llegeix el símbol que hi apareix i canvia d'estat en funció del contingut de la memòria i del símbol acabat de llegir. No se li permet modificar el contingut de les cel·les d'entrada, només el de la memòria interna. Ni tan sols posar una "marca" a determinades cel·les. Tampoc no se li permet fer recular el capçal cap a l'esquerra per tal de tornar a llegir algun símbol precedent. Cal remarcar que un dels conceptes que adquireixen més rellevància és el que s'interpreta com a *pas* d'un procés de càlcul. Es tracta de considerar que una computació requereix un nombre de passos igual al de vegades en què cal desglossar-la en càlculs elementals de la forma "estat per símbol". En el cas dels DFA, aquest nombre coincideix amb la longitud del mot processat. Però, en altres tipus d'autòmats, aquesta coincidència no existeix.

En el decurs d'aquest text ens permetrem utilitzar sovint termes que corresponen a aquesta interpretació no formal, fins i tot quan estiguem immersos en plantejaments formals. Així, per exemple, parlarem de *lectura* d'un mot, o de *funcionament* de l'autòmat, o encara, com acabem de remarcar, de *pas* d'una transició. Aquest ús ens permetrà fer més senzilla l'exposició sense que, com esperem, se'n ressentin ni la claredat ni el rigor.

Llenguatge reconegut per un DFA. Llenguatges regulars

Donat un autòmat finit $M = \langle Q, \Sigma, \delta, q_0, F\rangle$, s'anomena llenguatge *reconegut* per aquest autòmat, i es representa per $\mathsf{L}(M)$, el conjunt dels mots que en operar sobre l'estat inicial donen un estat acceptador. Formalment:

$$\mathsf{L}(M) = \{w \in \Sigma^* \mid q_0 \cdot w \in F\}.$$

En un autòmat determinista hi ha una correspondència biunívoca entre mots acceptats i camins que van de l'estat inicial a algun estat acceptador en el graf de transicions.

Exemple 4.3 L'autòmat especificat als exemples 4.1 i 4.2 accepta els mots de $\{a\,,b\}^*$ que tenen algun parell de a's separades per un submot de longitud múltiple de 3. Es proposa al lector, com a exercici, que construeixi directament, a partir d'aquesta definició, l'autòmat corresponent i que compari el resultat obtingut amb el donat.

Podem aprofitar aquest exemple per posar de manifest la *mecànica* de l'autòmat en el procés d'acceptació d'un mot. Els passos successius que duen a l'acceptació del mot $bababab$ són els següents:

$$q_0 bababab = q_0 ababab = q_1 babab = q_3 abab = q_4 bab = q_6 ab = q_2 b = q_2 \in F$$

Exemple 4.4 Ens proposem construir un autòmat que accepti els mots sobre $\{a\,,b\}$ que tenen una a a l'antepenúltima posició. Observem que durant la lectura d'un mot qualsevol, només els tres últims símbols llegits tenen rellevància. Els anteriors ja no poden influir en la decisió d'acceptar o no el mot. Els estats possibles s'identifiquen, doncs, amb els vuit possibles valors d'aquests tres últims símbols llegits. Si representem per $\langle xyz\rangle$ l'estat que correspon a haver llegit últimament un símbol x, seguit d'un símbol y i aquest seguit d'un símbol z, és clar que la funció de transició ha de prendre els valors

$$\delta(\langle xyz\rangle\,, a) = \langle yza\rangle \qquad \delta(\langle xyz\rangle\,, b) = \langle yzb\rangle\,.$$

Només ens queda identificar els estats en què encara no han estat llegits un mínim de tres símbols amb els estats que tenen b's a l'esquerra. En particular, l'estat inicial és el $\langle bbb\rangle$. Quant als estats acceptadors, són els quatre que tenen una a a l'antepenúltima posició. El diagrama de transicions d'aquest autòmat apareix a la figura 4.4.

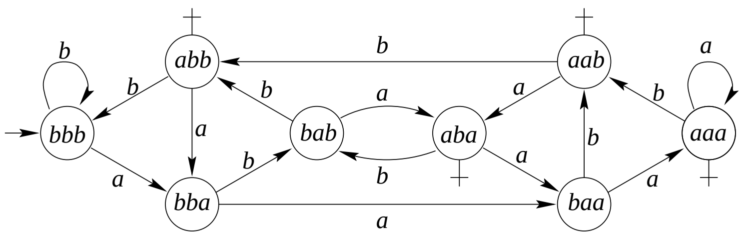

Fig. 4.4 *Autòmat que reconeix el llenguatge format pels mots que tenen un símbol a a l'antepenúltima posició*

Exemple 4.5 L'exemple anterior és fàcilment generalitzable al cas de mots sobre $\{a\,,b\}$ que tenen una a a la n-èsima posició començant per la dreta. El DFA resultant té aleshores

2^n estats, la meitat dels quals són acceptadors. Al capítol següent veurem que no hi ha cap DFA amb menys estats que accepti aquest llenguatge.

DEFINICIÓ. Un llenguatge s'anomena *regular* quan és reconegut per algun autòmat finit. Representem per Reg la família dels llenguatges regulars.

Diem que dos autòmats són *equivalents* quan reconeixen el mateix llenguatge. Formalment, M_1 és equivalent a M_2 si $\mathsf{L}(M_1) = \mathsf{L}(M_2)$.

4.3 Verificació d'autòmats finits

Entenem per *verificació* d'un autòmat la demostració que aquest autòmat reconeix un llenguatge donat. De manera similar a com hem fet al capítol 2 amb la verificació de gramàtiques, hi ha un plantejament general per dur a terme la verificació d'un autòmat finit que consisteix a associar a cada estat de l'autòmat un llenguatge que el caracteritzi, com pot ser el llenguatge format pels mots que porten de l'estat inicial a l'estat considerat.[1] Es tracta d'anticipar una descripció de cada un d'aquests llenguatges i de fer-ne la demostració conjuntament per a tots ells. Formalment, si L és el llenguatge que volem demostrar que és reconegut per l'autòmat, si el conjunt d'estats d'aquest autòmat és $\{q_1, \ldots, q_n\}$, si $L_1, \ldots, L_n$ en són els llenguatges associats, i si suposem que l'estat inicial és q_1, el que fem és establir n enunciats de la forma

$$\forall w \quad (q_1 w = q_i \iff w \in L_i)\,,$$

i demostrar-los conjuntament per inducció sobre la longitud del mot w. Els llenguatges L_i han de satisfer la condició que $L = \bigcup_{q_i \in F} L_i$, on F és el conjunt d'estats acceptadors. Veurem tot això amb detall en un exemple. Abans, però, remarquem que el fet de qualificar de general el procediment proposat no significa que sigui automàtic, perquè cal saber trobar descripcions dels llenguatges L_i que permetin identificar L amb la reunió dels que corresponen a estats acceptadors.

Exemple 4.6 L'autòmat de la figura 4.5 reconeix el llenguatge format per tots els mots sobre l'alfabet $\Sigma = \{a\,, b\}$, excepte el mot $bbbb$, tals que tot submot de 5 símbols conté com a mínim dues a's.

DEMOSTRACIÓ. Sigui L el llenguatge considerat. Per tal de facilitar l'exposició que segueix introduirem dos llenguatges auxiliars. Representem per X el subconjunt de mots de L que no acaben ni en b ni en ba, és a dir, que no contenen cap b en les dues posicions finals; i representem per Y el subconjunt de mots de L acabats en b però no en bb. Formalment podem expressar aquests llenguatges així:

$$X = (\lambda \cup a \cup \Sigma^* aa) \cap L$$
$$Y = (b \cup \Sigma^* ab) \cap L.$$

L'enunciat següent conté les onze equivalències que caracteritzen els llenguatges associats a cada un dels estats de l'autòmat.

[1] També podria considerar-se, alternativament, el llenguatge format pels mots que duen de l'estat considerat (pres com a inicial) a algun dels estats acceptadors.

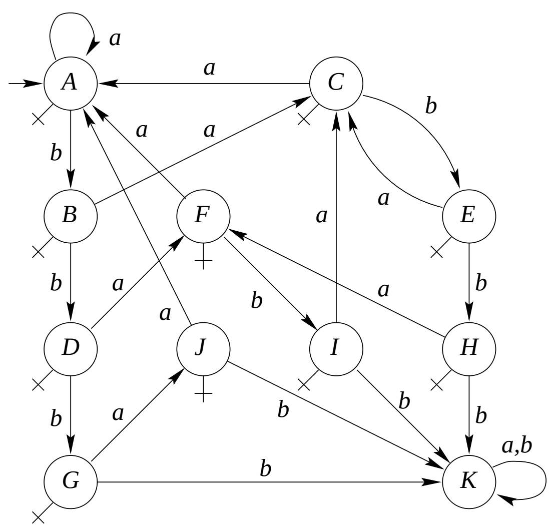

Fig. 4.5 DFA *que accepta tots els mots, tret del bbbb, en què tot submot de 5 símbols conté almenys dues a's*

$$\forall w \in \Sigma^* \quad \begin{cases} Aw = A \iff w \in X \\ Aw = B \iff w \in Xb \\ Aw = C \iff w \in Ya \\ Aw = D \iff w \in Xbb \\ Aw = E \iff w \in Yab \\ Aw = F \iff w \in Yba \\ Aw = G \iff w \in Xbbb \\ Aw = H \iff w \in Yabb \\ Aw = I \iff w \in Ybab \\ Aw = J \iff w \in Xbbba \\ Aw = K \iff w \notin L. \end{cases}$$

Observem que és el mateix verificar que la reunió dels llenguatges associats als estats acceptadors és L i verificar que la dels associats als estats no acceptadors és el complementari de L. Això és el que estableix l'equivalència corresponent a l'estat K. Farem la demostració per inducció sobre $n = |w|$.

Base. Tenim $n = 0 \iff w = \lambda$. La primera equivalència se satisfà pel fet de ser certa als dos costats. Les restants se satisfan en ser falses als dos costats.

Pas inductiu. Sigui $|w| = n > 0$ i suposem, per h.i., que són certes totes les equivalències $\forall u \in \Sigma^*$ tal que $|u| < n$. Es poden donar dos casos: o bé $w = va$, o bé $w = vb$, amb $v \in \{a, b\}^*$ i $|v| = n - 1$. Per a cada un d'aquests dos casos, estudiarem separadament cada una de les onze equivalències.

Cas 1.a Hem de demostrar que $Ava = A \iff va \in X$. Observem que, per tal que el mot va dugui a l'estat A, és necessari i suficient que el mot v dugui a un dels quatre estats, A, C, F o J, dels quals surt un arc amb etiqueta a que va a A. En la resta de casos que queden per analitzar partirem sempre d'una caracterització anàloga, sense que ens entretinguem a repetir-ne l'observació. En aquest cas tenim:

$$Ava = A \Leftrightarrow \left\{\begin{array}{l} Av = A \overset{\text{h.i.}}{\iff} v \in X \\ Av = C \overset{\text{h.i.}}{\iff} v \in Ya \\ Av = F \overset{\text{h.i.}}{\iff} v \in Yba \\ Av = J \overset{\text{h.i.}}{\iff} v \in Xbbba \end{array}\right\} \overset{?}{\iff} va \in X.$$

La implicació d'esquerra a dreta de l'última equivalència és immediata. Vegem el sentit contrari. Si $va \in X$, resulta que v no acaba en b i aleshores

$$\left\{\begin{array}{l} \text{si } v \text{ tampoc no acaba en } ba\text{, llavors } v \in X. \\ \text{si } v \text{ acaba en } ba\text{, llavors} \\ \quad \left\{\begin{array}{l} \left.\begin{array}{l} \text{o bé } v = ba \\ \text{o bé } v \text{ acaba en } aba \end{array}\right\} \text{ i en aquests dos casos } v \in Ya. \\ \text{o bé } v \text{ acaba en } bba \text{ i en aquest cas:} \\ \quad \left\{\begin{array}{l} \left.\begin{array}{l} \text{o bé } v = bba \\ \text{o bé } v \text{ acaba en } abba \end{array}\right\} \text{ i en aquests dos casos } v \in Yba. \\ \text{o bé } v \text{ acaba en } bbba \text{ i en aquest cas, com que } v \in L\text{, els} \\ \qquad \text{símbols precedents no poden ser ni } b \text{ ni } ba\text{, és a dir que} \\ \qquad v \in Xbbba. \end{array}\right. \end{array}\right. \end{array}\right.$$

Cas 1.*b* $\quad Ava = B \iff \text{fals} \iff va \in Xb.$

$$\textit{Cas } 1.c \quad Ava = C \iff \left\{\begin{array}{l} Av = B \overset{\text{h.i.}}{\iff} v \in Xb \\ Av = E \overset{\text{h.i.}}{\iff} v \in Yab \\ Av = I \overset{\text{h.i.}}{\iff} v \in Ybab \end{array}\right\}$$

$$\overset{?}{\iff} v \in Y \iff va \in Ya.$$

També aquí la penúltima equivalència és immediata d'esquerra a dreta. En sentit contrari, tenim el següent. Si $v \in Y$, resulta que v ha de ser de la forma ub amb u no acabat en b i aleshores

$$\left\{\begin{array}{l} \text{si } u \text{ tampoc no acaba en } ba\text{, llavors } u \in X. \\ \text{si } u \text{ acaba en } ba\text{, llavors} \\ \quad \left\{\begin{array}{l} \left.\begin{array}{l} \text{o bé } u = ba \\ \text{o bé } u \text{ acaba en } aba \end{array}\right\} \text{ i en aquests dos casos } u \in Ya. \\ \text{o bé } u \text{ acaba en } bba \text{ i en aquest cas:} \\ \quad \left\{\begin{array}{l} \text{o bé } u = bba \\ \text{o bé } u \text{ acaba en } abba \end{array}\right\} \begin{array}{l} \text{ja que } v \text{ no pot acabar en } bbbab\text{, i en els} \\ \qquad \text{dos casos } u \in Yba. \end{array} \end{array}\right. \end{array}\right.$$

Cas 1.*d* $\quad Ava = D \iff \text{fals} \iff va \in Xbb.$

Cas 1.*e* $\quad Ava = E \iff \text{fals} \iff va \in Yab.$

$$\textit{Cas } 1.f \quad Ava = F \iff \left\{\begin{array}{l} Av = D \overset{\text{h.i.}}{\iff} v \in Xbb \\ Av = H \overset{\text{h.i.}}{\iff} v \in Yabb \end{array}\right\}$$

$$\overset{?}{\iff} v \in Yb \iff va \in Yba.$$

De nou és immediat provar la penúltima implicació d'esquerra a dreta. Per fer-ho de dreta a esquerra, *no* podem basar-nos en la inclusió $Y \subseteq Xb \cup Yab$ perquè no és certa (el mot $bbab$ pertany a Y, però ni $bba \in X$ ni $bb \in Y$). Sí que és cert, en canvi, que $Yb \subseteq Xbb \cup Yabb$. Vegem-ho. Si $v \in Yb$, aleshores v acaba en bb però no en bbb. Per tant,

$$\begin{cases} \left.\begin{array}{l} \text{o bé } v = bb \\ \text{o bé } v = abb \\ \text{o bé } v \text{ acaba en } aabb \end{array}\right\} \text{ i en aquests tres casos } v \in Xbb. \\ \left.\begin{array}{l} \text{o bé } v = babb \\ \text{o bé } v \text{ acaba en } ababb \end{array}\right\} \text{ i en aquests dos casos } v \in Yabb. \\ (v \text{ no pot acabar en } bbabb) \end{cases}$$

Cas 1.g $\quad Ava = G \iff \text{fals} \iff va \in Xbbb.$

Cas 1.h $\quad Ava = H \iff \text{fals} \iff va \in Yabb.$

Cas 1.i $\quad Ava = I \iff \text{fals} \iff va \in Ybab.$

Cas 1.j $\quad Ava = J \iff Av = G \overset{\text{h.i.}}{\iff} v \in Xbbb \iff va \in Xbbba.$

En aquest cas, l'última equivalència és immediata en els dos sentits.

Cas 1.k $\quad Ava = K \iff Av = K \overset{\text{h.i.}}{\iff} v \notin L \iff va \notin L.$

L'última equivalència és, com sempre, immediata d'esquerra a dreta. Però també és clar que l'única forma que va no sigui de L és que v tampoc no ho sigui.

Cas 2.a $\quad Avb = A \iff \text{fals} \iff vb \in X.$

Cas 2.b $\quad Avb = B \iff Av = A \overset{\text{h.i.}}{\iff} v \in X \iff vb \in Xb.$

Cas 2.c $\quad Avb = C \iff \text{fals} \iff vb \in Ya.$

Cas 2.d $\quad Avb = D \iff Av = B \overset{\text{h.i.}}{\iff} v \in Xb \iff vb \in Xbb.$

Cas 2.e $\quad Avb = E \iff Av = C \overset{\text{h.i.}}{\iff} v \in Ya \iff vb \in Yab.$

Cas 2.f $\quad Avb = F \iff \text{fals} \iff vb \in Yba.$

Cas 2.g $\quad Avb = G \iff Av = D \overset{\text{h.i.}}{\iff} v \in Xbb \iff vb \in Xbbb.$

Cas 2.h $\quad Avb = H \iff Av = E \overset{\text{h.i.}}{\iff} v \in Yab \iff vb \in Yabb.$

Cas 2.i $\quad Avb = I \iff Av = F \overset{\text{h.i.}}{\iff} v \in Yba \iff vb \in Ybab.$

Cas 2.j $\quad Avb = J \iff \text{fals} \iff vb \in Xbbba.$

$$\textit{Cas } 2.k \quad Avb = K \iff \left\{\begin{array}{c} Av = G \overset{\text{h.i.}}{\iff} v \in Xbbb \\ Av = H \overset{\text{h.i.}}{\iff} v \in Yabb \\ Av = I \overset{\text{h.i.}}{\iff} v \in Ybab \\ Av = J \overset{\text{h.i.}}{\iff} v \in Xbbba \\ Av = K \overset{\text{h.i.}}{\iff} v \notin L \end{array}\right\} \overset{?}{\iff} vb \notin L.$$

La implicació d'esquerra a dreta és immediata en tots cinc casos. Vegem la implicació recíproca. Si $vb \notin L$, aleshores

- o bé $v \notin L$, i estem en el cinquè cas,
- o bé $v \in L$, i la causa que $vb \notin L$ està en el sufix. Cal diferenciar dos casos:
  - o bé vb és el mot $bbbb$, cas en el qual $v \in Xbbb$,
  - o bé els últims 5 símbols de vb contenen, com a màxim, una a. És a dir, que els quatre últims símbols de v contenen exactament una a (si v acabés en $bbbb$ ja hauria estat considerat com a $v \notin L$). Es tracta dels quatre casos següents:
    - $v = uabbb$ i u no acaba en b; en aquest cas $v \in Xbbb$.
    - $v = ubabb$ i u no acaba en b; en aquest cas $v \in Yabb$.
    - $v = ubbab$ i u no acaba en b; en aquest cas $v \in Ybab$.
    - $v = ubbba$ i u no acaba en b ni tampoc en ba; en aquest cas $v \in Xbbba$.

4.4 Autòmats finits indeterministes

Justificació

La construcció d'un autòmat finit determinista que reconegui un llenguatge donat és un problema que es va fent més difícil a mesura que creix el nombre d'estats involucrats. En aquesta secció introduirem el concepte d'autòmat finit indeterminista, que ens ajudarà a resoldre aquest problema. La raó d'això és que el nombre d'estats d'un autòmat d'aquest tipus pot arribar a ser logarítmicament menor que el nombre d'estats del mínim DFA que reconegui el mateix llenguatge. Com veurem immediatament, el recurs a aquests autòmats indeterministes és sempre possible: la família de llenguatges reconeguts per autòmats finits indeterministes és la mateixa família que la dels reconeguts per DFA.

Definició

Un *autòmat finit indeterminista* (abreujadament un NFA, de l'anglès *non-deterministic finite automaton*) és una estructura de la forma

$$M = \langle Q, \Sigma, \delta, I, F \rangle$$

en què, com ara veurem, Q, Σ i F tenen el mateix significat que en els DFA,

- Q és un conjunt finit d'estats.
- Σ és l'alfabet d'entrada.
- δ és la *funció de transició* que definirem tot seguit.
- $I \subseteq Q$ és el conjunt d'estats *inicials*.
- $F \subseteq Q$ és el conjunt d'estats *acceptadors*.

La funció de transició és ara una aplicació de la forma

$$\delta \colon \mathcal{P}(Q) \times \Sigma^* \longrightarrow \mathcal{P}(Q)\,,$$

que representem també amb notació multiplicativa,

$$\forall P \subseteq Q \quad \forall w \in \Sigma^* \qquad \delta(P, w) = P \cdot w,$$

i que satisfà els quatre axiomes següents:

$$\forall P_1, P_2 \subseteq Q \quad \forall x, y \in \Sigma^* \qquad \begin{cases} (1) & \varnothing \cdot x = \varnothing \\ (2) & P_1 \cdot \lambda = P_1 \\ (3) & (P_1 \cup P_2) \cdot x = P_1 \cdot x \,\cup\, P_2 \cdot x \\ (4) & P_1 \cdot (xy) = (P_1 \cdot x) \cdot y. \end{cases}$$

També aquí, igual que en el cas dels DFA, la funció de transició queda completament determinada coneixent els valors que pren sobre el conjunt finit $Q \times \Sigma$. És a dir que dues funcions de transició que coincideixen sobre $Q \times \Sigma$ són la mateixa. En efecte: coneguts els valors de δ sobre $Q \times \Sigma$, i donats un subconjunt d'estats qualsevol P i un mot qualsevol w, podem calcular $\delta(P, w)$ a partir dels components de P i de w. Siguin $P = \{q_1, \ldots, q_m\}$ i $w = a_1 \ldots a_n$. Per aplicació reiterada dels axiomes 3 i 4 tenim

$$P \cdot w \quad = \bigcup_{1 \leqslant i \leqslant m} ((\{q_i\} \cdot a_1) \cdots a_n).$$

Tant la taula com el diagrama de transicions d'un NFA tenen una presentació similar als corresponents dels DFA, per més que ara poden aparèixer diversos estats (o no aparèixer-ne cap) per a cada estat i símbol d'entrada. A més, mentre que en una taula d'un DFA l'estat inicial (únic) queda identificat per ser el primer de la llista, en un NFA cal marcar (per exemple, amb una fletxeta) els estats inicials quan n'hi ha més d'un.

Exemple 4.7 A continuació s'especifica, primer en forma de taula i després, a la figura 4.6, en forma de diagrama, la funció de transició d'un NFA de sis estats $\{A, \ldots, F\}$, dels quals els A i B són inicials i els B, D i E són acceptadors. L'alfabet d'entrada és $\{a, b\}$.

Taula 4.2 *Taula de transicions d'un* NFA

	a	b
$\rightarrow A$	C	A, D
$\rightarrow B\,\dagger$	–	E, F
C	A, D, F	B
$D\,\dagger$	E	A
$E\,\dagger$	–	–
F	D	F

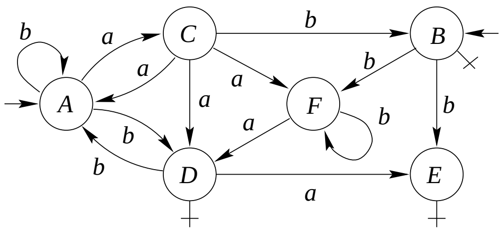

Fig. 4.6 *Diagrama de transicions de l'*NFA *de la taula 4.2*

Llenguatge reconegut per un NFA

Donats un autòmat finit indeterminista $M = \langle Q, \Sigma, \delta, I, F\rangle$ i un mot qualsevol $w = a_1 \dots a_n$, s'anomena *camí acceptador* d'aquest mot qualsevol seqüència d'estats (en cas d'haver-n'hi alguna) de la forma $(q_0, q_1, \dots, q_n)$ tal que

$$\begin{cases} q_0 \in I \\ q_n \in F \\ q_i \in q_{i-1} \cdot a_i \quad \forall i \colon 1 \leqslant i \leqslant n.^2 \end{cases}$$

S'anomena llenguatge *reconegut* per aquest autòmat, i es representa per $\mathsf{L}(M)$, el conjunt dels mots per als quals existeix almenys un camí acceptador. Formalment:

$$\mathsf{L}(M) = \{w \in \Sigma^* \mid \exists p \in I \; \exists q \in F \;\; q \in p \cdot w\} = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \varnothing\}.$$

Com hem dit al començament d'aquesta secció, la raó que justifica la definició dels NFA és la gran senzillesa de construcció, comparativament als DFA, dels autòmats indeterministes que accepten llenguatges donats. Posarem això de manifest en els exemples següents, en els quals exposarem els NFA que corresponen a llenguatges ja introduïts en exemples anteriors i per als quals ja coneixem els corresponents DFA.

Exemple 4.8 Un NFA que reconeix el llenguatge de l'exemple 4.3, és a dir, el conjunt de mots que tenen algun parell de a's separades per un submot de longitud múltiple de tres, és el que s'exposa a la figura 4.7. Observeu la simplicitat de la idea que guia el seu disseny. El camí acceptador es manté en bucle sobre l'estat inicial fins que arriba el moment de prendre en consideració la primera de les dues a's rellevants. Entre aquestes dues a's s'instal·la un comptador de tres estats que controla que el submot que les separa té efectivament longitud múltiple de tres. La segona a duu directament a l'únic estat acceptador, el qual ja no és abandonat.

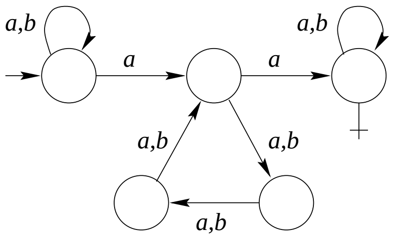

Fig. 4.7 NFA *que accepta els mots que tenen algun parell de a's separades per un submot de longitud múltiple de tres*

Exemple 4.9 Encara més senzilla resulta la construcció d'un NFA que accepti els mots que tenen una a a l'antepenúltima posició. Recordem que a l'exemple 4.4 ha estat construït un autòmat determinista que reconeix aquest llenguatge. Es tracta ara, simplement, de mantenir un bucle del qual només se surt amb una a seguida de dos símbols

[2] En el cas de $w = \lambda$, en fer $n = 0$, l'únic camí acceptador possible ha de ser de la forma (q_0), i aquestes condicions es redueixen a $q_0 \in I \cap F$.

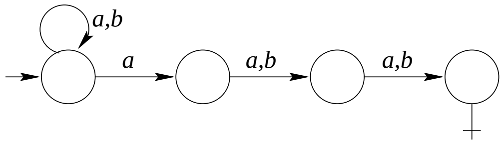

Fig. 4.8 NFA *que accepta els mots que tenen una a a l'antepenúltima posició*

addicionals. Això duu a l'estat acceptador a partir del qual ja no s'admet cap nou símbol. L'autòmat resultant s'ha dibuixat a la figura 4.8.

4.5 Equivalència dels NFA amb els DFA

Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ un NFA. Considerem un DFA M', sobre el mateix alfabet Σ, que tingui per estats els subconjunts d'estats de M, és a dir, que si anomenem Q' el conjunt d'estats de M', estem fent $Q' = \mathcal{P}(Q)$. En M, la funció de transició δ té la forma

$$\delta\colon \mathcal{P}(Q) \times \Sigma^* \longrightarrow \mathcal{P}(Q)\,.$$

Per tant, podem considerar aquesta mateixa funció, escrita en la forma

$$\delta\colon Q' \times \Sigma^* \longrightarrow Q',$$

com una funció de transició definida en M'. En efecte, d'una banda, té la forma correcta i, d'altra banda, satisfà els axiomes 1 i 2 de les funcions corresponents dels DFA, que no són més que la traducció dels axiomes 2 i 4 dels NFA en aquest cas. Només ens queda definir l'estat incial i els estats acceptadors de M'. Com a estat inicial prenem I (que per ser un subconjunt de Q és un estat de Q' ben definit). I com a conjunt F' d'estats acceptadors prenem els estats de Q' que continguin algun estat acceptador de M, és a dir,

$$F' = \{P \subseteq Q \mid P \cap F \neq \varnothing\}.$$

Demostrarem que el DFA $M' = \langle Q', \Sigma, \delta, I, F'\rangle$ construït així reconeix el mateix llenguatge que l'NFA M donat.

PROPOSICIÓ. *$\mathsf{L}(M') = \mathsf{L}(M)$.*

DEMOSTRACIÓ. De les tres igualtats següents

$$\mathsf{L}(M') = \{w \in \Sigma^* \mid I \cdot w \in F'\} = \{w \in \Sigma^* \mid I \cdot w \cap F \neq \varnothing\} = \mathsf{L}(M),$$

la primera és la definició del llenguatge reconegut pel DFA M', la segona resulta de la definició de F' i la tercera és la definició del llenguatge reconegut per l'NFA M. □

La proposició anterior ens permet afirmar que tot llenguatge reconegut per algun NFA és regular. Recíprocament, és immediat veure que tot llenguatge regular és reconegut per un NFA. En efecte, tot DFA pot ser entès com un cas particular d'NFA que satisfà dues condicions. La primera, que el conjunt d'estats inicials està reduït a un únic estat. La segona, que per a tot estat i tot símbol d'entrada, la transició corresponent consta exactament d'un estat. En conclusió, podem enunciar el teorema següent.

TEOREMA. *La família de llenguatges reconeguts pels* NFA *és la família dels llenguatges regulars.*

La definició que acabem de donar del DFA que correspon a un NFA donat és *constructiva*. Podem construir efectivament la taula de transicions d'aquest DFA partint de I (el conjunt d'estats inicials de l'NFA) a la primera línia de la taula, i anar definint les transicions que corresponen a cada línia de la taula i a cada entrada fent, simplement, la reunió de les transicions que corresponen als estats que componen l'agregat d'estats de la línia considerada. Només introduirem les transicions que corresponen als agregats que ens vagin apareixent en el decurs del càlcul, és a dir, només prendrem en consideració els estats del DFA que siguin accessibles des de l'estat inicial. Per tal de fer més llegible la notació dels estats, prescindim, sempre que és possible, de les claus i les comes que s'utilitzen en la descripció de conjunts. Així, per exemple, representem per ACD l'estat del DFA definit pel conjunt $\{A, C, D\}$, on A, C i D són estats de l'NFA del qual partim. Aquest procés de construcció d'un DFA a partir d'un NFA l'anomenem *determinització*. Veurem millor tot això amb un parell d'exemples.

Exemple 4.10 La figura 4.9 mostra la taula de transicions del DFA construït a partir de l'NFA de l'exemple 4.7. Observeu que només 14 estats dels $2^6 = 64$ possibles són accessibles.

	a	b
$\dagger\, AB$	C	$ADEF$
C	ADF	B
$\dagger\, ADEF$	CDE	ADF
$\dagger\, ADF$	CDE	ADF
$\dagger\, B$	$\varnothing$	EF
$\dagger\, CDE$	$ADEF$	AB
$\varnothing$	$\varnothing$	$\varnothing$
$\dagger\, EF$	D	F
$\dagger\, D$	E	A
F	D	F
$\dagger\, E$	$\varnothing$	$\varnothing$
A	C	AD
$\dagger\, AD$	CE	AD
$\dagger\, CE$	ADF	B

Fig. 4.9 *Determinització de l'*NFA *de l'exemple 4.7*

Exemple 4.11 La determinització de l'NFA definit a l'exemple 4.8 dóna lloc a la taula de transicions de la figura 4.10. Amb vista a dibuixar-ne el graf de transicions i preveient possibles utilitzacions ulteriors d'aquest DFA, hem assignat un número a cada estat en substitució de l'etiqueta formada amb els noms dels seus components.

El graf de transicions corresponent a aquesta taula es troba dibuixat a la figura 4.11. Pot observar-se que aquest graf no és el mateix que el de la figura 4.3, tot i correspondre al mateix llenguatge. La raó d'això és que el de la figura 4.3 és el DFA de menor nombre d'estats de tots els que reconeixen aquest llenguatge. Ara bé, el procés de determinització que acabem d'exposar no garanteix aquesta minimalitat del resultat. Així, en l'autòmat

		a	b
1	A	AB	A
2	AB	$ABCD$	AD
3	† $ABCD$	$ABCDE$	$ACDE$
4	AD	ABE	AE
5	† $ABCDE$	$ABCDE$	$ABCDE$
6	† $ACDE$	$ABCE$	$ABCE$
7	ABE	$ABCD$	ABD
8	AE	AB	AB
9	† $ABCE$	$ABCD$	$ABCD$
10	ABD	$ABCDE$	ADE
11	ADE	ABE	ABE

Fig. 4.10 *Determinització de l'*NFA *de l'exemple 4.8*

de la figura 4.11, els estats 3, 5, 6 i 9 poden reduir-se a un únic estat acceptador que bucleja sobre si mateix per a totes les entrades. Aquest únic estat seria l'equivalent de l'estat q_2 de l'autòmat definit a l'exemple 4.1, com és fàcil de comprovar. De tota manera, la construcció de l'autòmat mínim que correspon a un llenguatge regular donat serà l'objecte del pròxim capítol.

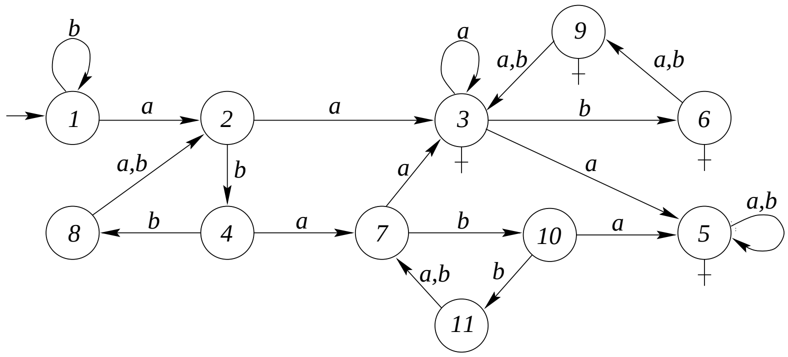

Fig. 4.11 *Graf corresponent al* DFA *de la taula anterior*

4.6 Autòmats finits amb λ-transicions

A la secció següent, veurem maneres de construir autòmats a partir de certes operacions bàsiques, com ara la reunió o la concatenació de llenguatges. Per a algunes de les operacions que considerarem, les construccions se simplifiquen notablement si utilitzem una variant dels autòmats finits indeterministes, la qual definirem en aquesta secció. Es tracta de permetre que l'autòmat pugui canviar d'estat sense haver de llegir cap símbol de l'entrada. O, per expressar-ho gràficament, que puguin haver-hi arcs sense etiquetes entre parells d'estats. Els anomenem autòmats finits amb λ-transicions, i utilitzem per a ells l'abreviació λNFA. Com demostrarem a continuació, els autòmats d'aquest tipus tenen la mateixa potència que els NFA (i, per tant, que els DFA). És a dir, tot llenguatge reconegut per un λNFA també ho és per un NFA normal (i viceversa).

DEFINICIÓ. Un *autòmat finit amb λ-transicions* és una estructura de la forma

$$M = \langle Q, \Sigma, \delta, I, F \rangle$$

en què Q, Σ, I i F tenen el mateix significat que en els NFA: es tracta, doncs, d'un conjunt finit d'estats, un alfabet d'entrada, i dos subconjunts d'estats, inicials i acceptadors, respectivament. La diferència, respecte dels NFA, radica en la *funció de transició*, δ, que aquí és una aplicació *qualsevol* (és a dir, no subjecta a cap condició o axioma) de la forma

$$\delta\colon\ Q \times (\Sigma \cup \{\lambda\}) \longrightarrow \mathcal{P}(Q)\,.$$

Observem que, mentre en el cas dels NFA, δ està definida sobre $\mathcal{P}(Q) \times \Sigma^*$, és a dir, per a qualsevol *subconjunt* d'estats i qualsevol *mot*, aquí hem restringit la seva definició a estats i símbols individuals (i λ). En els NFA, un dels axiomes requereix que per a tot estat q es tingui $\delta(q, \lambda) = q$. En canvi, aquí hem incorporat λ al domini de definició de δ, permetent-li prendre qualsevol valor. Així, per exemple, l'especificació $\delta(q_1, \lambda) = \{q_2, q_3\}$ expressa l'existència de dues λ-transicions, cap als estats q_2 i q_3, a partir de l'estat q_1.

La presentació d'un λNFA en forma de taula de transicions és similar a la dels NFA. Únicament cal afegir, a més de les columnes que corresponen a cada símbol de l'alfabet, una nova columna per especificar les λ-transicions. També els diagrames de transicions dels λNFA són com els dels NFA, amb la possibilitat addicional d'arcs amb etiqueta λ.

Exemple 4.12 La taula 4.3 i la figura 4.12 contenen, respectivament, la taula i el diagrama de transicions d'un mateix λNFA de cinc estats.

Taula 4.3 *Taula de transicions d'un λNFA*

	a	b	λ
$\to A$	$-$	B	$-$
$B\,\dagger$	C	D, E	$-$
$C\,\dagger$	B, C	D	D, E
D	$-$	A, B, D	C
E	E	E	A

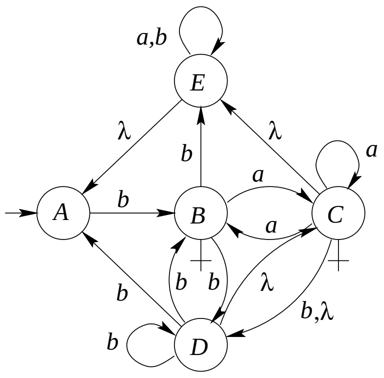

Fig. 4.12 *Diagrama de transicions del λNFA de la taula 4.3*

Llenguatge reconegut per un λNFA. Equivalència entre λNFA i NFA

Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ un λNFA. Diem que un mot w és *acceptat* per M quan hi ha un *camí acceptador* associat a w que duu d'un estat inicial a un estat acceptador, on ara aquest camí és una seqüència d'estats $(q_0, \ldots, q_m)$ tal que podem descompondre w de la forma $w = s_1 s_2 \ldots s_m$, amb $s_i \in \Sigma \cup \{\lambda\}$, de manera que:

$$\begin{cases} q_0 \in I \\ q_m \in F \\ \forall i : (1 \leqslant i \leqslant m) \quad q_i \in \delta(q_{i-1}, s_i). \end{cases}$$

El conjunt de tots els mots acceptats per M constitueix el llenguatge *reconegut* per M i es representa per $\mathsf{L}(M)$.

Exemple 4.13 El mot $w = babbbb$ és acceptat pel λNFA de l'exemple 4.12, ja que admet, entre d'altres possibles camins acceptadors, la seqüència d'estats

$$(A, B, C, D, B, E, A, B, D, C).$$

Les 9 transicions que corresponen a aquesta seqüència resulten de descompondre el mot considerat en la forma b-a-λ-b-b-λ-b-b-λ. Pot observar-se que hi ha altres camins acceptadors del mateix w. Alguns d'ells resulten de descompondre d'altres maneres el mot w. Però pot haver-hi, també, més d'un camí acceptador associat a la mateixa descomposició. Així, el camí

$$(A, B, C, E, E, E, A, B, D, C)$$

és un camí acceptador de w que es recorre considerant la mateixa descomposició en símbols alfabètics i λ-transicions que ha estat descrita anteriorment.

És clar que tot NFA normal pot ser considerat com un cas particular de λNFA. Es tracta, senzillament, de considerar que no hi ha cap λ-transició definida, és a dir, que $\delta(q, \lambda) = \varnothing$ per a tot estat q del λNFA. Per tal de fer la construcció recíproca, la de l'NFA que correspon a un λNFA donat, començarem introduint un concepte auxiliar.

Definició. Donat un subconjunt P qualsevol d'estats d'un λNFA, anomenem *λ-tancament* de P, i ho representem per $\Lambda(P)$, el conjunt d'estats als quals és possible accedir, a partir d'algun dels de P, mitjançant alguna seqüència de zero o més λ-transicions. Quan hàgim de referir-nos al λ-tancament d'un conjunt reduït a un sol estat q, utilitzarem generalment la notació $\Lambda(q)$ en lloc de $\Lambda(\{q\})$.

Estem ara en condicions de fer la construcció de l'NFA que reconeix el mateix llenguatge que un λNFA donat. Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ el λNFA de partida. Es tracta de substituir les λ-transicions per transicions alfabètiques que hi siguin equivalents, així com ampliar, si cal, el conjunt d'estats acceptadors. Formalment, l'NFA que construïm té la forma $M' = \langle Q, \Sigma, \delta', I, F'\rangle$, és a dir que el conjunt total d'estats i el subconjunt d'estats inicials són els mateixos que els de M. Per a cada estat $p \in Q$ i cada símbol $a \in \Sigma$ incorporem a les transicions directes des de p tots els estats als quals es pot accedir des de p per algun camí de λ-transicions acabat amb el símbol a. Formalment, la funció de transició δ' ve donada per

$$\delta'(p,a) = \bigcup_{q \in \Lambda(p)} \delta(q,a).$$

Finalment, el conjunt d'estats acceptadors està format pels estats des dels quals es pot accedir a un estat de F per un camí de λ-transicions. És a dir, formalment, $F' = \{q \mid \Lambda(q) \cap F \neq \varnothing\}$.

Exemple 4.14 El procés d'eliminació de les λ-transicions del λNFA definit a l'exemple 4.12 requereix, en primer lloc, el càlcul dels λ-tancaments dels estats de l'autòmat. El resultat és el de la taula 4.4.

Taula 4.4 *λ-tancaments dels estats de l'autòmat de l'exemple 4.12*

q	$\Lambda(q)$
A	A
B	B
C	A, C, D, E
D	A, C, D, E
E	A, E

Els estats d'acceptació del λNFA considerat són B i C. Com que aquests dos estats formen part dels λ-tancaments dels estats B, C i D, aquests tres estats constitueixen els estats d'acceptació de l'NFA que volem construir.

Per a cada estat q i cada símbol d'entrada a, el càlcul de $\delta'(q,a)$ consisteix a reunir els estats que corresponen a les transicions amb el símbol a des de cada un dels estats de $\Lambda(q)$. Així, en aquest exemple,

$$\delta'(E,b) = \delta(A,b) \cup \delta(E,b) = \{B\} \cup \{E\} = \{B,E\}.$$

És a dir, construïm la línia de l'estat E de la taula de δ', a base de considerar els estats de la taula de Λ que corresponen a E, i agrupar tots els estats que apareixen en la taula de δ com a transicions d'aquests estats considerats.

La funció de transició de l'NFA sense λ-transicions equivalent al de l'exemple que estem considerant s'exposa a la taula 4.5.

Taula 4.5 *Taula de transicions de l'NFA equivalent al λNFA de l'exemple 4.12*

	a	b
$\rightarrow A$	$-$	B
$B\,\dagger$	C	D, E
$C\,\dagger$	B, C, E	A, B, D, E
$D\,\dagger$	B, C, E	A, B, D, E
E	E	B, E

És interessant d'observar que aquest procediment d'eliminació de les λ-transicions d'un autòmat és anàleg al d'eliminació de les produccions unàries d'una gramàtica, estudiat al capítol 3. Com veurem més endavant, al capítol 6, aquesta analogia no és casual, sinó que és un reflex de la correspondència que existeix entre gramàtiques *regulars* i autòmats finits. Aquesta correspondència, que identifica les variables de la gramàtica amb els estats de l'autòmat i que identifica les produccions de la primera amb les transicions del segon, permet identificar igualment les produccions unàries amb les λ-transicions. En aquest sentit, el concepte de λ-tancament d'un estat donat és anàleg al de conjunt de variables unàriament derivables a partir d'una variable donada.

Proposició. $\mathsf{L}(M') = \mathsf{L}(M)$.

Demostració. Demostrem per separat les dues inclusions.

1. $\mathsf{L}(M') \subseteq \mathsf{L}(M)$.

Sigui w un mot qualsevol acceptat per l'NFA M'. Si $w = \lambda$, hi ha algun estat, q_0, que pertany alhora a I i a F'. Per tant, hi ha una seqüència de zero o més λ-transicions que duu de q_0 a algun estat de F. Els estats pels quals transcorre aquesta seqüència formen un camí acceptador de λ en el λNFA M. Si $w \neq \lambda$, sigui $w = a_1 a_2 \dots a_n$, amb $n > 0$. Com que $w \in \mathsf{L}(M')$, existeix un camí acceptador de w en M'. Sigui $(q_0, q_1, \dots, q_n)$ aquest camí. Es té

$$\begin{cases} q_0 \in I \\ q_n \in F' \\ q_i \in \delta'(q_{i-1}, a_i) \quad \forall i : (1 \leqslant i \leqslant n) \end{cases}$$

Per a cada i, la condició $q_i \in \delta'(q_{i-1}, a_i)$ implica que

$$\exists q'_{i-1} \in \Lambda(q_{i-1}) \quad q_i \in \delta(q'_{i-1}, a_i).$$

Això vol dir, en primer lloc, que existeix una seqüència de λ-transicions que duu de q_{i-1} a q'_{i-1}. Afegim al final d'aquesta seqüència la transició definida per $q_i \in \delta(q'_{i-1}, a_i)$ i podem considerar la cadena d'estats involucrats com el tros de camí acceptador de w que correspon a a_i. Ajuntant tots aquest trossos, formem un camí en M associat a w. Aquest camí duu, però, a l'estat q_n, que no té per què pertànyer a F. El fet que q_n pertanyi a F' ens permet afirmar tanmateix, partint de la definició de F', que existeix una seqüència de λ-transicions que duu de q_n a algun estat de F. L'afegiment d'aquests estats a la cadena que duu de q_0 a q_n constitueix un camí acceptador de w en M.

2. $\mathsf{L}(M) \subseteq \mathsf{L}(M')$.

Sigui w un mot qualsevol acceptat pel λNFA M. Sabem que hi ha una descomposició de w de la forma $w = s_1 s_2 \dots s_m$, que està associada a un camí acceptador de w i en la qual cada s_i és o bé un símbol de l'alfabet, o bé λ. Agrupem cada símbol alfabètic d'aquesta seqüència amb les λ-transicions que el precedeixen immediatament (a partir del símbol alfabètic anterior); si hi ha λ-transicions a final de la seqüència, constitueixen l'última agrupació. Així, per exemple, la seqüència b-a-λ-b-b-λ-b-b-λ, que hem considerat a l'exemple 4.13, s'agrupa en la forma $(b)(a)(\lambda\text{-}b)(b)(\lambda\text{-}b)(b)(\lambda)$. Considerem els estats del camí acceptador que corresponen al final de cada un d'aquests grups. Si intercalem aquests estats en el lloc que els correspon al final de cada grup, obtindrem una representació en què dos grups consecutius apareixeran en la forma

$$\dots(\lambda\text{-}\cdots\text{-}\lambda\text{-}a_i)q'_i\ (\lambda\text{-}\cdots\text{-}\lambda\text{-}a_{i+1})q'_{i+1}\ \dots$$

De la definició de la funció δ' de l'NFA M', resulta que entre cada dos d'aquests estats seleccionats existeix la relació $q'_{i+1} \in \delta'(q'_i, a_{i+1})$. A més, l'estat que segueix al grup de l'últim símbol alfabètic, o bé és un estat de F, o bé va seguit d'un grup de λ-transicions i, de la definició del conjunt F' d'estats acceptadors de M', deduïm que ha de pertànyer a F'. En conseqüència, la seqüència d'estats q'_i seleccionats així constitueix un camí acceptador de w en M'. □

Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família de llenguatges reconeguts pels λNFA és la família dels llenguatges regulars.*

4.7 Operacions bàsiques amb autòmats

En aquesta secció veurem que la classe de llenguatges regulars és tancada respecte de les operacions conjuntistes bàsiques (reunió, intersecció, complementació) i d'altres com la concatenació, el tancament de Kleene i el revessament. També respecte dels morfismes inversos. Tot això, a més del seu interès teòric, ens serà molt útil també a l'hora de dissenyar autòmats, perquè les demostracions donades seran constructives i ens permetran obtenir autòmats complexos per composició d'altres de més senzills.

Complementació

Sigui $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un DFA. Definim l'*autòmat complementari* M^{c} com aquell que s'obté a partir de M donant als estats no acceptadors el caràcter de acceptadors, i viceversa. Formalment: $M^{\mathsf{c}} = \langle Q, \Sigma, \delta, q_0, Q-F\rangle$.

PROPOSICIÓ. *$\mathsf{L}(M^{\mathsf{c}}) = \overline{\mathsf{L}(M)}$.*

DEMOSTRACIÓ. Per a tot mot $w \in \Sigma^*$,

$$w \in \mathsf{L}(M^{\mathsf{c}}) \Longleftrightarrow q_0 \cdot w \in Q-F \Longleftrightarrow q_0 \cdot w \notin F \Longleftrightarrow w \notin \mathsf{L}(M) \Longleftrightarrow w \in \overline{\mathsf{L}(M)} \ .$$

□

Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de l'operació de complementació.*

Exemple 4.15 Considerem el problema de trobar un autòmat finit que accepti els mots sobre $\{a\,,b\}$ que no comencen amb el prefix bb. Si bé és possible dissenyar directament un DFA que reconegui aquest llenguatge, és molt més natural partir de l'NFA que reconeix el llenguatge complementari, que hem dibuixat a la figura 4.13.

Determinitzar aquest autòmat es redueix a afegir-li un nou estat. Un cop convertits els estats no acceptadors en acceptadors i l'acceptador en no acceptador, en resulta l'autòmat de la figura 4.14.

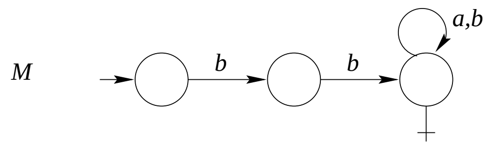

Fig. 4.13 NFA *que accepta els mots que comencen amb bb*

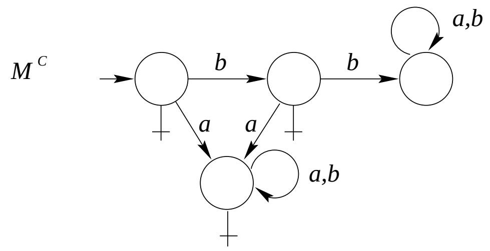

Fig. 4.14 DFA *que accepta els mots que no comencen amb bb*

Observeu que la demostració donada requereix que els autòmats siguin deterministes. Vegem a continuació un exemple senzill que demostra que és erroni aplicar la regla de la complementació sobre NFA.

Exemple 4.16 Considerem els dos autòmats de la figura 4.15. Clarament l'autòmat M és un NFA que reconeix el llenguatge L, sobre l'alfabet $\Sigma = \{a\,, b\}$, dels mots que acaben en a. M^c és l'NFA que resulta d'aplicar la regla de la complementació directament sobre M. És clar que el llenguatge que reconeix M^c és Σ^* i no pas el complementari de L.

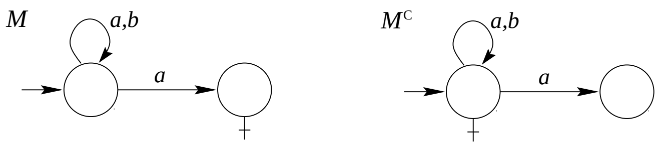

Fig. 4.15 *Un* NFA *M i el seu pseudocomplementari M^c*

Intersecció

Donats dos llenguatges regulars qualssevol, L_1 i L_2, i dos DFA, M_1 i M_2, que reconeixen aquests llenguatges, volem construir, a partir dels autòmats M_1 i M_2, un altre DFA que reconegui el llenguatge $L_1 \cap L_2$. La idea bàsica de la construcció és considerar, com a estats del nou autòmat, els parells del producte cartesià dels estats de M_1 pels de M_2 i simular la funció de transició de l'autòmat M_1 sobre la primera component i la funció de transició de l'autòmat M_2 sobre la segona. El conjunt d'estats acceptadors haurà de ser, per tant, el conjunt de parells que tenen estats acceptadors en les dues components. La definició formal és la següent.

Siguin $M_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$ i $M_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$ dos DFAqualssevol que reconeixen L_1 i L_2, respectivament. Definim $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ així:

- $Q = Q_1 \times Q_2$
- $q_0 = (q_1, q_2)$
- $F = F_1 \times F_2$
- $\forall p \in Q_1 \ \forall q \in Q_2 \ \forall w \in \Sigma^* \quad \delta((p,q), w) = (\delta_1(p,w), \delta_2(q,w))$.

Escriurem, simplement, $(p,q) \cdot w = (pw, qw)$ per descriure la funció de transició. Remarquem que està ben definida, ja que és immediat comprovar que satisfà els axiomes corresponents a les funcions de transició per a DFA.

PROPOSICIÓ. $\mathsf{L}(M) = L_1 \cap L_2$.

DEMOSTRACIÓ. Per a tot mot w de Σ^* es té

$$\begin{aligned} w \in \mathsf{L}(M) &\iff (q_1, q_2)\cdot w \in F \\ &\iff (q_1 w, q_2 w) \in F_1 \times F_2 \\ &\iff q_1 w \in F_1 \ \wedge \ q_2 w \in F_2 \\ &\iff w \in L_1 \ \wedge \ w \in L_2 \\ &\iff w \in L_1 \cap L_2. \end{aligned}$$

□

Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de l'operació d'intersecció.*

Exemple 4.17 Volem dissenyar un autòmat determinista que reconegui el llenguatge L format per λ i els mots sobre l'alfabet $\Sigma = \{0,1\}$ que són nombres binaris múltiples de sis. Fem servir la notació '$\mathsf{valor}_b(x)$' per representar una funció numèrica que pren el valor 0 quan $x = \lambda$ i que pren altrament el valor del mot x interpretat com un nombre en base b. Amb aquesta notació, el llenguatge L s'expressa així

$$L = \{w \in \{0,1\}^* \mid \exists n \geqslant 0 \ \mathsf{valor}_2(w) = 6n\}.$$

Partirem dels autòmats M_1 i M_2 de la figura 4.16, que reconeixen els llenguatges, sobre l'alfabet $\Sigma = \{0,1\}$, dels mots de valor binari múltiple de dos i de tres, respectivament. Així doncs, podem plantejar la construcció de l'autòmat que reconeix L com la intersecció d'aquests dos autòmats, ja que els nombres múltiples de sis són els que són, alhora, múltiples de dos i de tres.

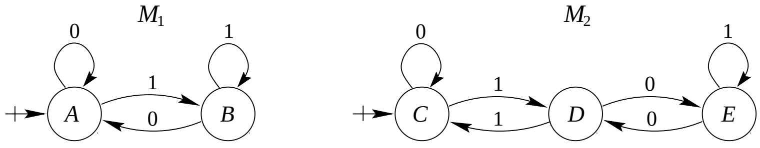

Fig. 4.16 DFA *que reconeixen els múltiples de dos i de tres*

La taula de transicions del DFA que reconeix L i que resuta d'aplicar directament l'algorisme d'intersecció als autòmats M_1 i M_2 és la taula 4.6. Hem prescindit dels parèntesis i la coma a l'hora de representar els parells del producte cartesià. Així, l'estat (B,D) és representat simplement per BD. La primera línia de la taula conté les transicions corresponents a l'estat inicial, que és el format pel parell d'estats inicials components, en aquest cas AC. Hem afegit a aquesta taula una primera columna que enumera els estats que apareixen a la segona columna, amb la finalitat de donar un nom significatiu a cadascun dels estats del diagrama d'estats corresponent (Fig. 4.17). En efecte, el nombre associat a cada estat, q, és el residu de la divisió entera per sis del valor binari dels mots que porten des de l'estat inicial fins a q. De fet, aquest autòmat correspon al disseny que hauríem obtingut construint el DFA directament, pensant els estats com les classes quocient de $\mathbb{N}$ mòdul sis.

Taula 4.6 *Transicions d'un* DFA *que accepta* λ *i els nombres binaris múltiples de sis*

		0	1
0	$\dagger AC$	AC	BD
1	BD	AE	BC
2	AE	AD	BE
3	BC	AC	BD
4	AD	AE	BC
5	BE	AD	BE

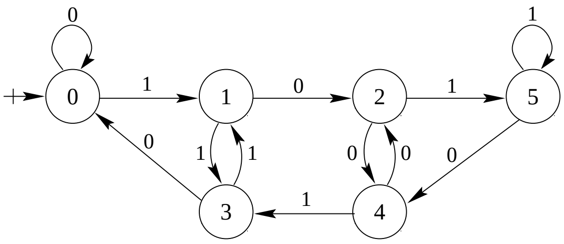

Fig. 4.17 *Representació en diagrama d'estats de la taula anterior*

Fixeu-vos que l'autòmat construït no és l'*autòmat mínim* per al llenguatge L (al capítol 5 es veurà el significat precís de l'expressió 'autòmat mínim per a un llenguatge donat' i també un algorisme per construir-lo), ja que els estats 1,4 i 2,5 són *equivalents* i es poden aparellar i donar lloc a l'autòmat de la figura 4.18, el qual té menys estats i reconeix també el llenguatge L.

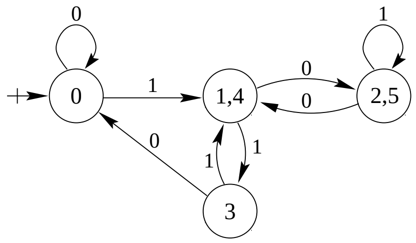

Fig. 4.18 DFA *mínim que accepta* λ *i els nombres binaris múltiples de sis*

L'algorisme d'intersecció de DFA via producte cartesià d'estats es pot estendre de manera natural al cas general d'intersecció d'NFA. L'extensió és la següent. Siguin $M_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1\rangle$ i $M_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2\rangle$ dos NFAqualssevol que reconeixen L_1 i L_2, respectivament. Definim $M = \langle Q, \Sigma, \delta, I, F\rangle$ així:

- $Q = Q_1 \times Q_2$
- $I = I_1 \times I_2$
- $F = F_1 \times F_2$
- $\forall P \subseteq Q \ \forall w \in \Sigma^* \quad \delta(P, w) \ = \bigcup_{(p,q)\in P} \delta_1(p, w) \times \delta_2(q, w)\,.$

Escriurem, simplement, $(p, q) \cdot w = pw \times qw$ per descriure la funció de transició. Remarquem que, com en el cas determinista, aquesta funció està ben definida, ja que és fàcil comprovar que satisfà els axiomes corresponents a les funcions de transició per a NFA.

PROPOSICIÓ. $\mathsf{L}(M) = L_1 \cap L_2\,.$

DEMOSTRACIÓ. Per a tot mot w de Σ^*:

$$\begin{aligned} w \in \mathsf{L}(M) &\iff \exists i\in I\ \exists f\in F \quad f \in iw \\ &\iff \exists i_1\in I_1\ \exists i_2\in I_2\ \exists f_1\in F_1\ \exists f_2\in F_2 \quad (f_1, f_2) \in (i_1, i_2)w \\ &\iff \exists i_1\in I_1\ \exists i_2\in I_2\ \exists f_1\in F_1\ \exists f_2\in F_2 \quad (f_1, f_2) \in i_1w \times i_2w \\ &\iff \exists i_1\in I_1\ \exists i_2\in I_2\ \exists f_1\in F_1\ \exists f_2\in F_2 \quad f_1\in i_1w \wedge f_2\in i_2w \\ &\iff w\in L_1 \wedge w\in L_2 \\ &\iff w \in L_1 \cap L_2. \end{aligned}$$

□

Fixeu-vos que si partim d'autòmats M_1 i M_2 deterministes, el resultat de l'algorisme d'intersecció és també un autòmat finit determinista, mentre que en el cas general d'intersecció d'NFA el resultat és un autòmat finit no determinista (vegeu la figura 4.19).

Observeu també que l'algorisme d'intersecció pot fer créixer, *en el cas pitjor*, el nombre d'estats de l'autòmat resultant en un factor quadràtic respecte de la talla dels autòmats de partida, ja que el nombre de parells en $Q_1 \times Q_2$ és $\|Q_1\|\cdot\|Q_2\|$. Diem en el cas pitjor perquè, si bé tal com l'hem definit l'autòmat intersecció té sempre $\|Q_1\|\cdot\|Q_2\|$ estats, habitualment no comptem els estats que no són accessibles, ja que a la pràctica construirem els autòmats intersecció partint dels estats inicials i considerant a cada pas només els estats accessibles. Tot i així, el cas més desfavorable no permet estalviar-nos cap estat.

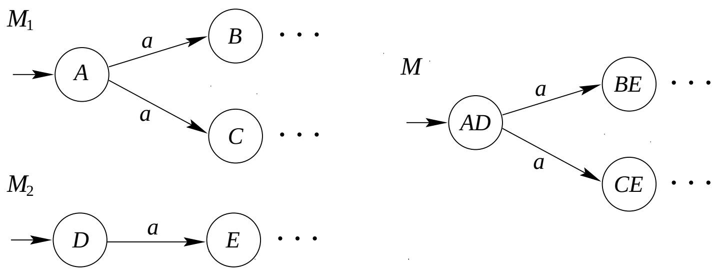

Fig. 4.19 *Dos* NFA, M_1 *i* M_2, *i l'*NFA *"intersecció" corresponent* M

L'objectiu final de la majoria de problemes pràctics sobre llenguatges regulars és l'obtenció d'un autòmat finit determinista (i, per tant, "implementable") que reconegui un llenguatge determinat. En el cas que aquest autòmat s'obtingui per intersecció de dos NFA, M_1 i M_2 qualssevol, tenim dues possibilitats diferents d'actuació:

1. Determinitzar primer els autòmats M_1 i M_2 i fer després el producte cartesià de les versions deterministes corresponents.

2. Fer primer el producte cartesià de M_1 per M_2 i determinitzar-ne l'autòmat resultant.

Fixeu-vos que el primer mètode és clarament preferible al segon. En efecte, si M_1 i M_2 tenen $n/2$ estats cadascun, podem calcular així les fites que corresponen al cas pitjor:

1. Determinitzar els autòmats M_1 i M_2 suposa passar de dos NFA de $n/2$ estats a dos DFA de $2^{n/2}$ estats. El producte cartesià d'aquests dos darrers autòmats ens proporciona un DFA de 2^n estats.
2. Fer el producte cartesià de M_1 per M_2 suposa passar de dos NFA de $n/2$ estats cadascun a un de $n^2/4$ estats. Determinitzar aquest darrer autòmat pot portar a un DFA de $2^{n^2/4}$ estats.

Reunió

Donats dos NFA qualssevol, una manera senzilla d'obtenir un altre NFA que reconegui el llenguatge reunió dels llenguatges reconeguts pels dos autòmats de partida és la construcció que anomenem *junció*. Aquesta construcció es basa simplement a *ajuntar* els dos autòmats en un de sol i permetre, de manera indeterminista, que la computació de qualsevol mot pugui començar en un estat inicial del primer autòmat o en un estat inicial del segon. Formalment, és el següent.

Siguin $M_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1 \rangle$ i $M_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2 \rangle$ dos NFA qualssevol que reconeixen L_1 i L_2, respectivament. Suposem, sense pèrdua de generalitat, que els conjunts d'estats Q_1 i Q_2 són disjunts (tal com hem fet al capítol 2 amb els conjunts de variables quan tractàvem les operacions amb gramàtiques). Definim, a partir d'ells dos, l'autòmat $M = \langle Q, \Sigma, \delta, I, F \rangle$ així:

- $Q = Q_1 \cup Q_2$
- $I = I_1 \cup I_2$
- $F = F_1 \cup F_2$
- $\delta = \delta_1 \cup \delta_2$, és a dir, $\delta(q,w) = \begin{cases} \delta_1(q,w) & \text{si} \quad q \in Q_1 \\ \delta_2(q,w) & \text{si} \quad q \in Q_2\,. \end{cases}$

PROPOSICIÓ. $\mathsf{L}(M) = L_1 \cup L_2$.

DEMOSTRACIÓ. Per a tot mot w de Σ^* es té

$$w \in \mathsf{L}(M) \iff \exists i \in I_1 \cup I_2 \ \exists f \in F_1 \cup F_2 \quad f \in iw.$$

Com que, per construcció, els subgrafs corresponents als autòmats M_1 i M_2 són disjunts en l'autòmat M, la condició anterior és equivalent a

$$(\,\exists i \in I_1 \ \exists f \in F_1 \ \ f \in \delta_1(i,w)\,) \ \vee \ (\,\exists i \in I_2 \ \exists f \in F_2 \ \ f \in \delta_2(i,w)\,)$$

i aquesta disjunció està reflectint precisament que $w \in L_1 \vee w \in L_2$, és a dir que $w \in L_1 \cup L_2$.
□

Fixeu-vos que el resultat de la junció de dos autòmats finits sempre és un autòmat finit no determinista. Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de l'operació de reunió.*

Exemple 4.18 L'NFA M de la figura 4.20 és l'autòmat resultant de la junció dels autòmats M_1 i M_2 de la figura 4.16. El llenguatge que reconeix és, doncs, el dels mots sobre l'alfabet $\{0,1\}$ que tenen un valor binari múltiple de dos o de tres.

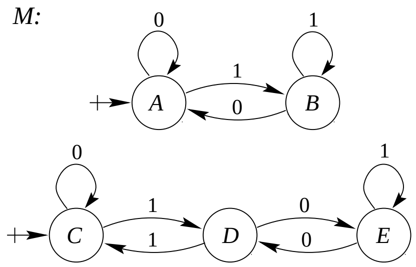

Fig. 4.20 NFA *M obtingut per junció dels* DFA *M_1 i M_2 de la figura 4.16*

El mètode de junció presentat s'aplica a NFA en general. Ara bé, si partim de dos DFA també podem plantejar la reunió via producte cartesià d'estats, tal com havíem fet per a la intersecció de DFA. Únicament s'ha de modificar el conjunt d'estats acceptadors, que ara estarà format per tots aquells parells que tinguin un estat acceptador del primer autòmat en la primera component, o bé un estat acceptador del segon autòmat en la segona. Formalment, la construcció és la següent.

Siguin $M_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1\rangle$ i $M_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2\rangle$ dos DFAqualssevol que reconeixen L_1 i L_2, respectivament. Definim $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ així:

- $Q = Q_1 \times Q_2$
- $q_0 = (q_1, q_2)$
- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$
- $\forall p \in Q_1\ \forall q \in Q_2\ \forall w \in \Sigma^* \quad \delta((p,q),w) = (\delta_1(p,w), \delta_2(q,w))$.

Igual que hem fet en casos precedents, utilitzem notació multiplicativa per descriure la funció de transició: $(p,q)\cdot w = (pw, qw)$. També aquí podem comprovar que la definició satisfà els axiomes corresponents.

PROPOSICIÓ. $\mathsf{L}(M) = L_1 \cup L_2$.

DEMOSTRACIÓ. Per a tot mot w de Σ^* es té

$$\begin{aligned} w \in \mathsf{L}(M) &\iff (q_1,q_2)\cdot w \in F \\ &\iff (q_1w, q_2w) \in (F_1 \times Q_2) \cup (Q_1 \times F_2) \\ &\iff q_1w \in F_1 \vee q_2w \in F_2 \\ &\iff w \in L_1 \vee w \in L_2 \\ &\iff w \in L_1 \cup L_2\,. \end{aligned}$$

□

De la mateixa manera que en el cas d'intersecció via producte cartesià, com que partim de dos autòmats deterministes, el resultat també és un autòmat determinista.

Exemple 4.19 Com en l'exemple 4.18, prenem els autòmats M_1 i M_2 de la figura 4.16 i construïm el DFA, M, unió d'aquests dos. Evidentment el llenguatge reconegut per M és el dels mots sobre l'alfabet $\{0, 1\}$ que, interpretats com a nombres binaris, són múltiples de dos o de tres. Com que la construcció via el producte cartesià d'estats és idèntica al cas de la intersecció, l'autòmat resultant tindrà el mateix diagrama de transicions que el de la figura 4.17, i únicament diferirà d'aquell en el conjunt d'estats acceptadors, que ara s'amplia amb tres nous estats. El resultat (que tampoc no és mínim, ja que els estats 0 i 3 són equivalents) és el de la figura 4.21.

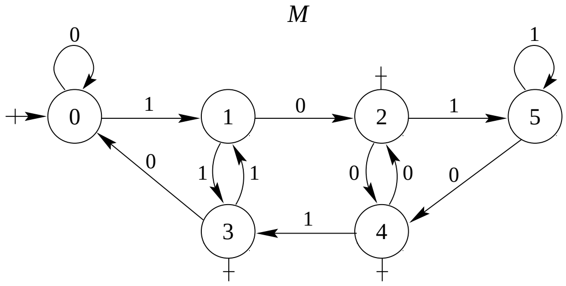

Fig. 4.21 DFA *equivalent al de la figura 4.20, obtingut per producte cartesià*

Sembla, doncs, que si partim de dos DFA tenim dues possibilitats d'actuació per construir un DFA que sigui la reunió d'ells dos.

1. Fer la junció dels dos autòmats i determinitzar-ne l'NFA resultant.
2. Fer el producte cartesià dels dos autòmats inicials.

De tota manera, no es tracta pròpiament de dos *mètodes* diferents, ja que la determinització de l'NFA obtingut per junció de dos DFA coincideix exactament amb el producte cartesià d'aquests. Vegem-ho en un exemple.

Exemple 4.20 Treballem una vegada més amb l'exemple del llenguatge L dels mots sobre l'alfabet $\{0, 1\}$, de valor binari múltiple de dos o de tres. Utilitzant els DFA M_1 i M_2 de la figura 4.16, hem construït ja dos autòmats finits que reconeixen L. El primer (exemple 4.18) és un NFA obtingut per junció de M_1 i M_2, mentre que el segon (exemple 4.19) és un DFA obtingut per producte cartesià de M_1 i M_2. La taula de transicions que surt de determinitzar el primer dels autòmats és la taula B de la figura 4.22, mentre que la taula de transicions corresponent al segon autòmat, obtinguda per producte cartesià de DFA, és la taula A de la mateixa figura 4.22. Fixeu-vos que les dues taules són la mateixa i que, per tant, corresponen a un mateix DFA. Això és degut al fet que els dos autòmats inicials són deterministes i, per tant, els estats que apareixen en el procés de determinització de l'autòmat obtingut per junció són conjunts de dos estats, els quals coincideixen exactament amb els parells d'estats de l'algorisme de reunió via producte cartesià.

En el cas general de partir de dos NFA, també tenim dues possibilitats d'actuació per construir el DFA reunió corresponent. Són les següents.

1. Ajuntar els dos NFA i determinitzar-ne l'autòmat resultant.

Taula A

		0	1
0	$\dagger(A,C)$	(A,C)	(B,D)
1	(B,D)	(A,E)	(B,C)
2	$\dagger(A,E)$	(A,D)	(B,E)
3	$\dagger(B,C)$	(A,C)	(B,D)
4	$\dagger(A,D)$	(A,E)	(B,C)
5	(B,E)	(A,D)	(B,E)

Taula B

		0	1
0	$\dagger\{A,C\}$	$\{A,C\}$	$\{B,D\}$
1	$\{B,D\}$	$\{A,E\}$	$\{B,C\}$
2	$\dagger\{A,E\}$	$\{A,D\}$	$\{B,E\}$
3	$\dagger\{B,C\}$	$\{A,C\}$	$\{B,D\}$
4	$\dagger\{A,D\}$	$\{A,E\}$	$\{B,C\}$
5	$\{B,E\}$	$\{A,D\}$	$\{B,E\}$

Fig. 4.22 *Producte cartesià i junció d'uns mateixos* DFA

2. Determinitzar cadascun dels autòmats de partida i fer-ne després el producte cartesià.

Una anàlisi de la complexitat en el cas pitjor, similar a la realitzada en el cas de la intersecció, posa de manifest que aquests dos mètodes tenen la mateixa complexitat. Concretament, si partim de dos autòmats amb $n/2$ estats cadascun, els dos algorismes construeixen DFA amb un nombre d'estats $O(2^n)$. Així, els dos mètodes són equivalents des del punt de vista de la complexitat, però en general preferirem el primer, ja que permet obtenir de forma immediata l'autòmat reunió indeterminista.

Com que per a tot parell de llenguatges es té $L_1 \cup L_2 = \overline{(\overline{L_1} \cap \overline{L_2})}$, també podem plantejar la reunió com a complementari de la intersecció de complementaris dels autòmats inicials. Com a mètode, però, és equivalent al segon dels proposats anteriorment, ja que per poder aplicar el primer pas de complementació s'han de determinitzar prèviament els autòmats i aleshores ja s'ha de fer el producte cartesià per a la intersecció.

També aquí podríem considerar la reunió d'NFA directament via producte cartesià. Aquest algorisme de producte cartesià sobre NFA es faria definint la funció de transició de la mateixa manera que en l'algorisme d'intersecció, amb l'únic requeriment addicional d'haver de considerar que els NFA inicials tenen totes les transicions definides. Això no és cap problema greu, ja que el pas previ de completar les transicions no definides només suposa haver d'afegir un únic estat a l'NFA considerat. De tota manera, no ho desenvoluparem perquè els inconvenients de la complexitat són els mateixos que els vistos per al cas de la intersecció.

Si haguéssim introduït els algorismes de reunió d'autòmats abans que els d'intersecció, hauríem pogut plantejat la intersecció com a complementació de la reunió de complementaris dels autòmats inicials. De tota manera, aquesta forma de procedir no introdueix cap mètode nou, ja que, de manera semblant al que acabem de veure amb l'operació de reunió, resulta equivalent al mètode 1 d'intersecció (primer determinitzar i després fer el producte cartesià). Això és cert independentment del sistema seguit per a la reunió, ja que el primer pas de complementació exigeix la determinització prèvia dels autòmats de partida i, tractant-se d'una reunió de DFA, la reunió per producte cartesià és, com ja hem vist, equivalent a la reunió per junció.

Concatenació

Donats dos NFA qualssevol, M_1 i M_2, que reconeguin els llenguatges L_1 i L_2, per tal de construir un λNFA que reconegui la concatenació $L_1 \cdot L_2$ incorporem λ-transicions que connectin cada un dels estats acceptadors de M_1 amb cada un dels estats inicials de M_2. Podem expressar formalment aquesta construcció de la manera següent.

Sigui $M_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1\rangle$ i sigui $M_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2\rangle$. Considerem, com sempre, que Q_1 i Q_2 són disjunts. Definim $M = \langle Q, \Sigma, \delta, I, F\rangle$ així:

- $Q = Q_1 \cup Q_2$
- $I = I_1$
- $F = F_2$
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{si } q \in Q_1 \text{ i } a \in \Sigma \\ \delta_2(q,a) & \text{si } q \in Q_2 \text{ i } a \in \Sigma\,. \end{cases}$
- $\delta(q,\lambda) = I_2 \quad$ per a cada $q \in F_1$

PROPOSICIÓ. $\mathsf{L}(M) = L_1 \cdot L_2$.

DEMOSTRACIÓ. Demostrarem per separat les dues inclusions.

1. $\mathsf{L}(M) \subseteq L_1 L_2$.

Sigui w un mot de $\mathsf{L}(M)$. Qualsevol camí acceptador de w en M consisteix en una seqüència d'estats el primer dels quals pertany a I_1 i l'últim dels quals pertany a F_2. Com que l'única connexió entre els grafs de M_1 i M_2 és a través de les λ-transicions introduïdes en la construcció de M, el camí acceptador que estem considerant està format per dues subseqüèncias consecutives. La primera està constituïda exclusivament per estats de Q_1, l'últim dels quals ha de pertànyer a F_1. La segona ho està per estats de Q_2, el primer dels quals pertany a I_2. En conseqüència, el camí acceptador pot descompondre's en dos camins acceptadors, el primer sobre M_1 i el segon sobre M_2. Aquesta descomposició porta associada la del mot w en la forma w_1-λ-w_2, amb $w_1 \in L_1$ i $w_2 \in L_2$.

2. $L_1 L_2 \subseteq \mathsf{L}(M)$.

Sigui w un mot del llenguatge $L_1 L_2$. En aquest cas existeixen $w_1 \in L_1$ i $w_2 \in L_2$ tals que $w = w_1 w_2$. Han d'existir també els camins acceptadors $(p_0, ..., p_n)$ de w_1 sobre M_1 i $(q_0, ..., q_m)$ de w_2 sobre M_2. Com que $p_n \in F_1$ i $q_0 \in I_2$, hi ha una λ-transició definida a M entre p_n i q_0. Com que totes les transicions de M_1 i M_2 s'han incorporat a M, la seqüència $(p_0, ..., p_n, q_0, ..., q_m)$ és un camí acceptador de w en M. $\square$

Sovint, sol ser convenient eliminar les λ-transicions introduïdes a la construcció anterior, per tal d'evitar que la utilització repetida d'aquesta construcció pugui conduir a la creació d'autòmats amb cadenes llargues de λ-transicions. En la pràctica, podem evitar la introducció de λ-transicions, característica del mètode anterior, a base d'ajuntar en un sol pas la introducció de cada λ-transició i la seva eliminació. Es tracta, simplement, de fer el següent:

Connectar els estats acceptadors de M_1 amb els successors dels inicials de M_2, i mantenir com a acceptadors els estats de F_1 quan algun dels estats de I_2 pertany també a F_2.

Expressat formalment és:

- $Q = Q_1 \cup Q_2$
- $I = I_1$

- $F = \begin{cases} F_2 & \text{si } I_2 \cap F_2 = \varnothing \quad (\text{és a dir, si } \lambda \notin \mathsf{L}(M_2)) \\ F_1 \cup F_2 & \text{altrament} \end{cases}$
- $\delta(p,a) = \begin{cases} \delta_1(p,a) & \text{si } p \in Q_1 - F_1 \\ \delta_1(p,a) \cup \delta_2(I_2,a) & \text{si } p \in F_1 \\ \delta_2(p,a) & \text{si } p \in Q_2\,. \end{cases}$

La figura 4.23 il·lustra el procés de connectar els grafs de M_1 i M_2 a base d'afegir els arcs que van dels estats acceptadors de M_1 als estats successors dels inicials de M_2. D'aquesta manera, els estats acceptadors de M_1 passen a comportar-se com a estats inicials de M_2. Com a estats inicials de l'autòmat resultant, prenem els de M_1; i com a estats acceptadors prenem els de M_2, i també els de M_1 en el cas que l'autòmat M_2 accepti el mot buit. A la figura s'hi fan constar, en traç discontinu, els arcs afegits.

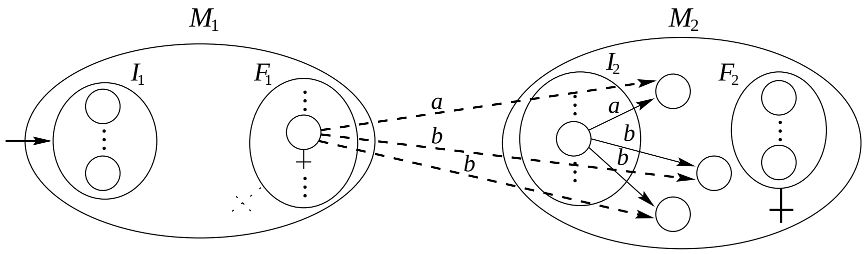

Fig. 4.23 *Esquema de la concatenació de dos autòmats*

A vegades es poden fer algunes simplificacions que estalvien feina a l'hora de determinitzar i minimitzar posteriorment l'autòmat resultant. Per exemple, tots els estats $p \in I_2$ als quals no arribi cap transició des d'algun estat de M_2 poden ser eliminats directament de l'autòmat concatenació resultant, ja que esdevenen inaccessibles en l'autòmat resultant.

Com a conseqüència de la proposició precedent, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de l'operació de concatenació.*

Exemple 4.21 Volem construir un autòmat que reconegui el llenguatge L sobre l'alfabet $\{a\,,b\}$ format pels mots en què tot parell de a's consecutives va seguit immediatament per un parell de b's consecutives. Per tal de facilitar aquesta construcció, començarem considerant els dos llenguatges auxiliars següents: representem per L_1 el conjunt de mots acabats amb dues a's i representem per L_2 el conjunt de mots que comencen amb dues b's. Podem expressar formalment el llenguatge L en funció de L_1 i L_2 així:

$$L = \{w \in \{a\,,b\}^* \mid \forall x,y \ \ (w = xy \ \wedge \ x \in L_1) \implies y \in L_2\}.$$

N'obtindrem una expressió més senzilla si considerem el complementari de L, que és

$$\begin{aligned} \overline{L} =& \{w \in \{a\,,b\}^* \mid \exists x,y \ \ w = xy \ \wedge \ x \in L_1 \ \wedge \ y \notin L_2\} \\ =& \{w \in \{a\,,b\}^* \mid \exists x \in L_1 \ \exists y \in \overline{L_2} \ \ w = xy\} = L_1 \cdot \overline{L_2}. \end{aligned}$$

A la figura 4.24 s'han dibuixat els autòmats M_1, M_2 i M_2^c, que reconeixen L_1, L_2 i $\overline{L_2}$, respectivament. Recordem que els autòmats M_2 i M_2^c ja han estat introduïts a l'exemple 4.15.

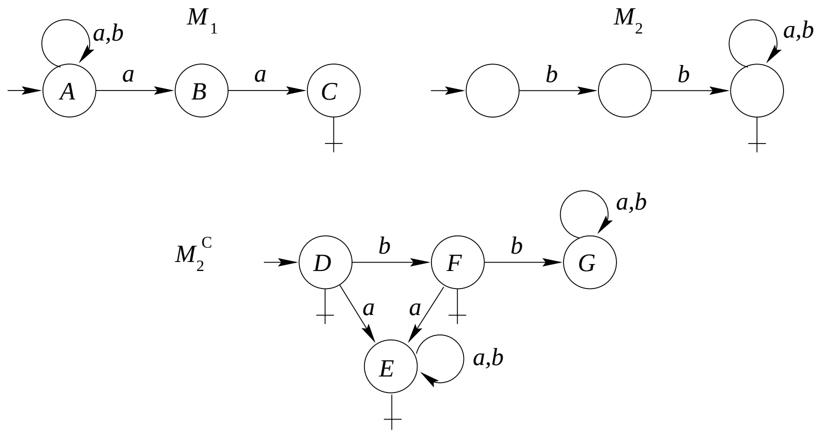

Fig. 4.24 *Autòmats que reconeixen* $\{a\,,b\}^*aa$, $bb\{a\,,b\}^*$ *i* $\overline{bb\{a\,,b\}^*}$

A la figura 4.25 s'ha dibuixat l'autòmat que s'obté a partir de M_1 i M_2^c per aplicació de la construcció que acabem de donar. Remarquem que l'estat D esdevé inaccessible i pot ser eliminat.

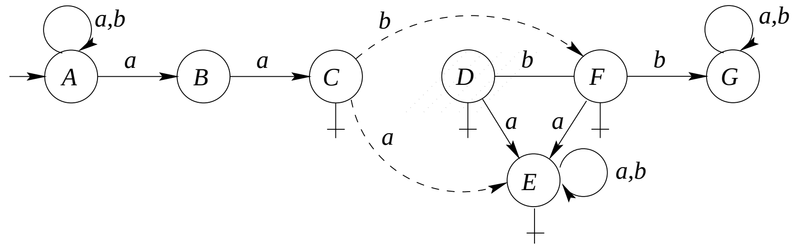

Fig. 4.25 *Concatenació de dos autòmats*

Finalment, a la figura 4.26 apareix l'autòmat determinista que reconeix el llenguatge L, obtingut a partir del de la figura 4.25 un cop determinitzat i passat al complementari.

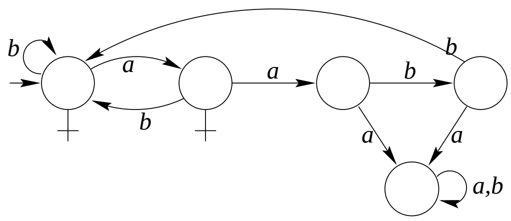

Fig. 4.26 DFA *que accepta mots en què dues a's sempre van seguides de dues b's*

Observeu que el resultat de la concatenació de dos autòmats és, en general, un autòmat no determinista, encara que partim d'autòmats, M_1 i M_2, deterministes.

Tancament positiu i tancament de Kleene

1. Tancament positiu. Sigui L un llenguatge regular. Demostrarem que el tancament positiu de L també és regular. A tal fi, construirem un λNFA que reconeixerà L^+ a partir de qualsevol NFA M que reconegui L. La idea subjacent en aquesta construcció és, dita de manera informal, concatenar l'autòmat M amb si mateix. Es tracta d'incorporar a M les λ-transicions que connectin els estats acceptadors amb els inicials. Descriurem formalment aquesta construcció a continuació.

Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$. Definim $M^+ = \langle Q, \Sigma, \delta^+, I, F\rangle$, de manera que el conjunt total d'estats i els subconjunts d'estats inicials i acceptadors són els mateixos que els de M. La funció de transició δ^+ ve definida així:

- $\delta^+(q,a) = \delta(q,a)$ per a tot $q \in Q$ i tot $a \in \Sigma$
- $\delta^+(q,\lambda) = I$ per a tot $q \in F$.

PROPOSICIÓ. $\mathsf{L}(M^+) = L^+$.

DEMOSTRACIÓ. Demostrarem per separat les dues inclusions.

1. $L^+ \subseteq \mathsf{L}(M^+)$.

Demostrarem, per inducció sobre n, que $\forall n \geqslant 1 \;\; L^n \subseteq \mathsf{L}(M^+)$.

- *Base:* $n = 1$. Per a tot mot $w \in L$, existeix un camí acceptador sobre l'autòmat M. Aquest mateix camí serà un camí acceptador de w sobre M^+, ja que M^+ conté totes les transicions de M i conserva els mateixos estats acceptadors.
- *Pas inductiu.* Sigui $w \in L^{n+1}$. Aleshores existeixen $x \in L^n$ i $y \in L$ tals que $w = xy$. Per hipòtesi d'inducció, tenim $x \in \mathsf{L}(M^+)$, és a dir, existeix un camí acceptador, $(p_0, ..., p_i)$, de x sobre M^+. També sabem que existeix un camí acceptador, $(q_0, ..., q_j)$, de y sobre M, ja que y pertany a L. Com que $p_i \in F$ i $q_0 \in I$, hi ha una λ-transició definida a M^+ entre p_i i q_0. I com que totes les transicions de M s'han incorporat a M^+, la seqüència $(p_0, ..., p_i, q_0, ..., q_j)$ és un camí acceptador de w en M^+.

2. $\mathsf{L}(M^+) \subseteq L^+$.

Sigui $w \in \mathsf{L}(M^+)$ un mot qualsevol. Sabem que existeix un camí acceptador, $(p_0, ..., p_n)$, de w sobre l'autòmat M^+. La demostració, que preferim donar de manera menys formalitzada que la precedent, es fa per inducció sobre el nombre de transicions de δ^+ que no estan en δ i que intervenen en el camí acceptador $(p_0, ..., p_n)$ de w sobre M^+. Si aquest nombre és zero, aleshores el mateix camí és un camí acceptador de w sobre M i, per tant, $w \in L \subseteq L^+$. Si el nombre és igual o superior a 1, aleshores podem partir el camí en dos trossos, $p_0, ..., p_i$ i $p_{i+1}, ..., p_n$, on el pas de p_i a p_{i+1} representa la darrera transició específica de M^+, que forçosament ha de ser una λ-transició. De la construcció de M^+, sabem que $p_i \in F$, que $p_{i+1} \in \delta^+(p_i, \lambda)$ i que $p_{i+1} \in I$. Així doncs, el primer tros és un camí acceptador d'algun mot x sobre M^+. El segon tros ho és d'algun mot y sobre M. I es té $xy = w$. La hipòtesi d'inducció ens diu que el mot x pertany a L^+. En conseqüència, $w = xy \in L^+L \subseteq L^+$. □

De manera anàloga a com hem fet en el cas de l'operació de concatenació, també aquí podem ajuntar en un sol pas la introducció de les λ-transicions efectuada a la construcció anterior i la seva eliminació. El procediment per fer el tancament positiu d'un llenguatge regular consistex, aleshores, a fer el següent:

Afegir a l'autòmat donat transicions alfabètiques que vagin dels estats acceptadors als *successors* dels inicials. El conjunt total d'estats i els subconjunts d'estats inicials i acceptadors queden inalterats.

La figura 4.27 dóna l'esquema d'aquesta construcció. Formalment, la funció de transició de M^+ es defineix així:

$$\delta^+(p,a) = \begin{cases} \delta(p,a) \cup \delta(I,a) & \text{si } a \in \Sigma \text{ i } p \in F \\ \delta(p,a) & \text{si } a \in \Sigma \text{ i } p \in Q-F \end{cases}$$

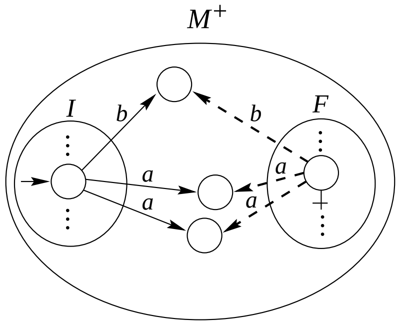

Fig. 4.27 *Esquema del tancament positiu d'un* NFA

2. Tancament de Kleene. Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ un autòmat que reconeix L. La construcció de l'autòmat que reconeix L^* és pràcticament idèntica a la de l'autòmat per al tancament positiu. En particular, si $\lambda \in L$, aleshores, com que $L^+ = L^*$, la mateixa construcció d'abans ens serveix. Si, en canvi, $\lambda \notin L$, l'única diferència és que farà falta afegir un nou estat inicial, q_0, que també sigui acceptador, per acceptar el mot buit. Formalment, sigui $q_0 \notin Q$. Definim el λNFA $M' = \langle Q', \Sigma, \delta', I', F'\rangle$ així:

- $Q' = Q \cup \{q_0\}$
- $I' = I \cup \{q_0\}$
- $F' = F \cup \{q_0\}$.

Quant a la funció δ', tenim

- $\delta'(q_0, a) = \varnothing$ per a tot $a \in \Sigma \cup \{\lambda\}$
- $\delta'(q, a) = \delta(q, a)$ per a tot $q \in Q$ i tot $a \in \Sigma$
- $\delta'(q, \lambda) = I$ per a tot $q \in F$.

PROPOSICIÓ. $\mathsf{L}(M') = L^*$.

DEMOSTRACIÓ. La demostració es faria de la mateixa manera que per al tancament positiu, considerant a part el cas del mot buit λ. □

Si volem fer una construcció equivalent a l'anterior, sense introduir λ-transicions, de manera semblant a com hem fet en els casos de la concatenació i del tancament positiu, només hem de modificar la funció de transició de la manera següent:

$$\delta'(p,a) = \begin{cases} \varnothing & \text{si } p = q_0 \\ \delta(p,a) \cup \delta(I,a) & \text{si } p \in F \\ \delta(p,a) & \text{altrament} \end{cases}$$

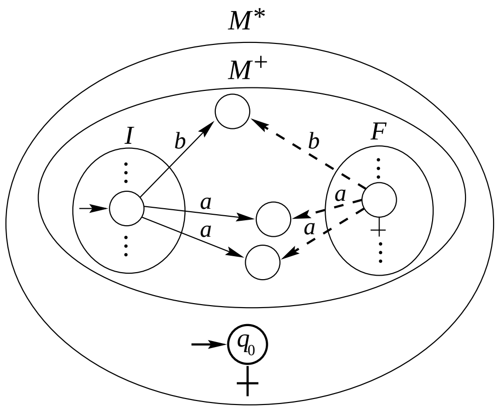

Fig. 4.28 *Esquema del tancament de Kleene d'un* NFA

L'esquema que correspon a aquesta construcció és el descrit a la figura 4.28.

Com a conseqüència de les dues proposicions precedents, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de les operacions de tancament positiu i tancament de Kleene.*

Exemple 4.22 Considerem el llenguatge L dels mots sobre l'alfabet $\Sigma = \{a\,, b\}$ que tenen longitud múltiple de tres i tals que tots els símbols que ocupen posicions múltiples de tres són a's. Els mots d'aquest llenguatge tenen la forma genèrica següent:

$$w = s_1 s_2 a s_4 s_5 a \dots s_{n-2} s_{n-1} a\,, \quad \text{on } \forall i \;\; s_i \in \Sigma.$$

Podem dividir aquests mots en fragments de tres símbols, així $x_1 = s_1 s_2 a\,,\ x_2 = s_4 s_5 a\,,$ $\dots\ x_m = s_{n-2} s_{n-1} a$, en què els dos primers símbols de cada fragment són qualssevol i el tercer és una a. Això permet considerar cada mot com a concatenació de zero o més mots del llenguatge $L_1 = \Sigma^2 a$, de manera que el llenguatge L és el tancament de Kleene de L_1,

$$L = L_1^* = \{aaa, aba, baa, bba\}^*.$$

A la figura 4.29 hi ha un autòmat que reconeix el llenguatge L_1, mentre que a la figura següent hi ha l'NFA M_1, que surt d'aplicar estrictament l'algorisme de tancament de Kleene sobre el primer autòmat. Aquest és, de fet, l'autòmat que buscàvem per al L. Fixeu-vos que els autòmats M_2 i M_3 de la mateixa figura són autòmats més senzills, equivalents a M_1. M_3 correspon al disseny més senzill que se'ns podia haver acudit d'entrada. Representa un comptador de símbols mòdul 3 tal que les transicions corresponents als símbols de posició múltiple de tres són possibles només amb un símbol a.

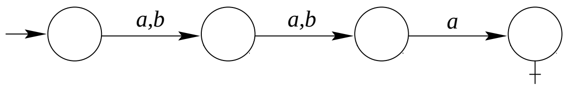

Fig. 4.29 *Autòmat finit que reconeix el llenguatge* $L_1 = \Sigma^2 a$

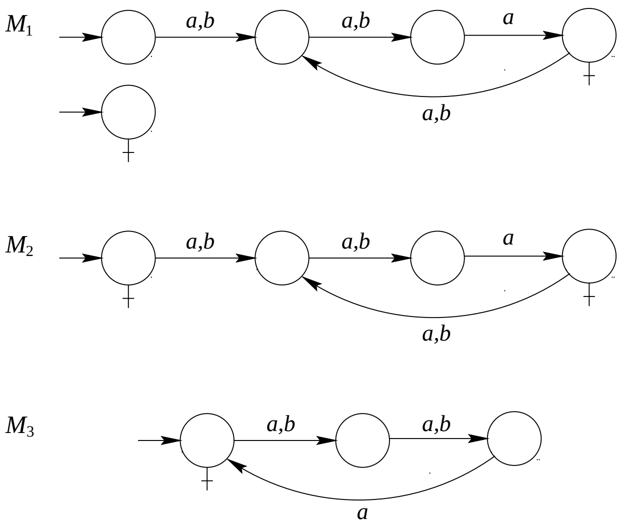

Fig. 4.30 *Autòmats que reconeixen el llenguatge $L = (\Sigma^2 a)^*$*

Revessament

La idea constructiva de la demostració és molt senzilla, ja que només fa falta *donar la volta* a l'autòmat que reconeix L per tenir-ne un altre que reconegui L^{R}. Això vol dir canviar el sentit de les transicions i intercanviar els estats inicials i acceptadors. Formalment, és el següent.

Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ un NFA qualsevol que reconeix un llenguatge L donat. A partir d'ell construïm l'autòmat $M_{\mathsf{R}} = \langle Q, \Sigma, \delta_{\mathsf{R}}, I_{\mathsf{R}}, F_{\mathsf{R}}\rangle$, on:

- $I_{\mathsf{R}} = F$
- $F_{\mathsf{R}} = I$
- $\forall q \in Q\ \forall a \in \Sigma \quad \delta_{\mathsf{R}}(q, a) = \{p \mid q \in \delta(p, a)\}$.

PROPOSICIÓ. *Se satisfà que $\mathsf{L}(M_{\mathsf{R}}) = L^{\mathsf{R}}$.*

DEMOSTRACIÓ. Observeu primer que de la definició de la funció de transició es dedueix fàcilment que, si a l'autòmat M hi ha un camí de l'estat p a l'estat q llegint el mot w, en M_{R} ha d'existir un camí de q a p llegint els símbols de w a l'inrevés (w^{R}). És a dir,

$$\forall w \in \Sigma^*\ \forall p, q \in Q \qquad q \in \delta(p, w) \iff p \in \delta_{\mathsf{R}}(q, w^{\mathsf{R}}).$$

Per tant, $\forall w \in \Sigma^*$,

$$\begin{aligned} w \in \mathsf{L}(M_{\mathsf{R}}) &\iff \exists q \in I_{\mathsf{R}}\ \exists p \in F_{\mathsf{R}}\ \ p \in \delta_{\mathsf{R}}(q, w) \\ &\iff \exists q \in F\ \exists p \in I\ \ q \in \delta(p, w^{\mathsf{R}}) \\ &\iff w^{\mathsf{R}} \in \mathsf{L}(M) = L. \end{aligned}$$ □

Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de l'operació de revessament.*

Observeu que el caràcter determinista o no determinista de l'autòmat M de partida no ens diu res sobre el determinisme de M_{R}. En general, no té sentit determinitzar un NFA abans de passar al seu revessat, ja que el procés de revessament pot tornar a donar un altre autòmat indeterminista que caldrà determinitzar novament per obtenir el resultat definitiu.

Exemple 4.23 Construirem un autòmat per al llenguatge L^{R} sobre l'alfabet $\Sigma = \{a\,, b\}$, on L és el llenguatge

$$\{w \in \Sigma^* \mid \exists x, y, z \in \Sigma^* \; \exists n \geqslant 0 \;\; w = xayaz \wedge |y| = 3n\,\}.$$

Ja coneixem un DFA per al llenguatge L (vegeu l'autòmat de la figura 4.3); per tant, l'autòmat M_{R} que reconeix L^{R} s'obté d'aquell invertint el sentit de les transicions i intercanviant estats inicials per acceptadors, i viceversa. El resultat és l'autòmat de la figura 4.31.

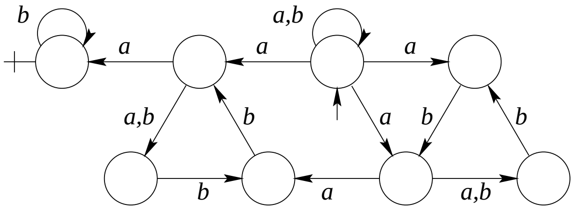

Fig. 4.31 NFA *que reconeix el revessat del llenguatge de l'exemple 4.2*

L'autòmat de l'exemple precedent és altament indeterminista. Si el determinitzem i n'obtenim la versió de nombre mínim d'estats (al capítol 5 donem un algorisme per trobar autòmats mínims), tornem a obtenir el DFA M de la figura 4.3. És a dir que M i M_{R} són equivalents i, per tant, L i L^{R} són el mateix llenguatge. En efecte, el fet que un mot tingui un parell de símbols a separats per una cadena de longitud múltiple de tres és independent de si l'escrivim en un sentit o altre. Aquest no és un fet aïllat. Hi ha molts llenguatges que són iguals als seus revessats. Els següents en són alguns exemples:

1. Σ^*, $\varnothing$
2. $\{w \in \Sigma^* \mid w = w^{\mathsf{R}}\,\}$
3. $\{w \in \{a\,, b\}^* \mid \exists n \geqslant 0 \; |w|_a = 5n\,\}$
4. $\{w \in \{0\,, 1\}^* \mid \exists n \geqslant 0 \; \mathsf{valor}_2(w) = 3n\,\}$
5. $\{w \in \{a\,, b\}^* \mid |w|_a = |w|_b\,\}$.

Per als llenguatges 1, 3 i 4, que són regulars, podeu comprovar aquesta igualtat via autòmats finits.

Morfisme invers

Volem demostrar que si tenim un morfisme $h : \Sigma_1^* \longrightarrow \Sigma_2^*$ i un llenguatge $L_2 \subseteq \Sigma_2^*$ que és regular, aleshores el llenguatge $h^{-1}(L_2)$ també és regular. Com que L_2 és regular, existeix un DFA $M_2 = \langle Q, \Sigma_2, \delta_2, q_0, F\rangle$ tal que $\mathsf{L}(M_2) = L_2$. La idea essencial és construir un DFA, M_1, amb els mateixos estats que M_2, i que davant d'un símbol $a \in \Sigma_1$ simuli el comportament de M_2 sobre la cadena $h(a) \in \Sigma_2^*$. D'aquesta manera, és clar que si M_1

accepta un mot w és perquè M_2 accepta $h(w)$ i, per tant, $\mathsf{L}(M_1) = h^{-1}(\mathsf{L}(M_2)) = h^{-1}(L_2)$. Aquesta simulació es pot fer en un sol pas de càlcul de M_1, ja que la computació de la cadena $h(a)$ en M_2 no és res més que un encadenament de transicions entre estats, és a dir, el pas des d'un estat de partida p, abans de llegir el primer símbol de $h(a)$, a un estat d'arribada q, després d'haver llegit el darrer símbol de $h(a)$.

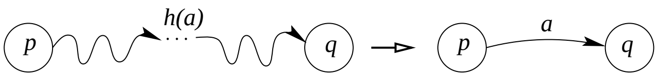

Fig. 4.32 *Correspondència entre transicions de M_2 i de M_1*

Formalitzant aquestes idees, definim $M_1 = \langle Q, \Sigma_1, \delta_1, q_0, F\rangle$, on per a tot $p \in Q$ i $w \in \Sigma_1^*$ fem $\delta_1(p, w) = \delta_2(p, h(w))$. Es tracta d'una funció de transició ben definida, ja que satisfà els axiomes:

1. $\delta_1(p, \lambda) = \delta_2(p, h(\lambda)) = \delta_2(p, \lambda) = p$
2. $\delta_1(p, xy) = \delta_2(p, h(xy)) = \delta_2(p, h(x)\cdot h(y)) = \delta_2(\delta_2(p, h(x)), h(y))$
   $= \delta_1(\delta_1(p, x), y)\,.$

Proposició. $\mathsf{L}(M_1) = h^{-1}(L_2)$.

Demostració. Per a tot mot w de Σ^* es té

$w \in \mathsf{L}(M_1) \iff \delta_1(q_0, w) \in F \iff \delta_2(q_0, h(w)) \in F \iff h(w) \in \mathsf{L}(M_2)$. □

Observeu que la definició de δ_1 és constructiva, ja que el càlcul de δ_1 a partir de δ_2 només cal fer-lo, com sempre, sobre els símbols de Σ_1.

Exemple 4.24 Considerem de nou el llenguatge format pels múltiples de tres en binari més λ, del qual ja hem donat un autòmat a l'exemple 4.17

$$L_2 = \{w \in \{0\,,1\}^* \ \mid\ \exists n \geqslant 0 \ \mathsf{valor}_2(w) = 3n\}\,.$$

Considerem ara el morfisme $h\colon \{a\,,b\}^* \longrightarrow \{0,1\}^*$, definit per $h(a) = 01$ i $h(b) = 1001$. Volem conèixer quin és el llenguatge $L_1 = h^{-1}(L_2)$.

A la figura 4.33 hem reproduït l'autòmat donat a la figura 4.16

$$M_2 = \langle \{[0], [1], [2]\}\,, \{0\,,1\}\,, \delta_2\,, [0], \{[0]\}\rangle\,,$$

que reconeix el llenguatge L_2.

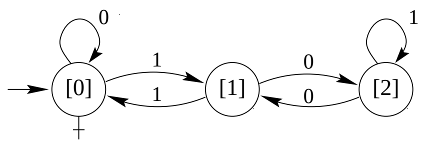

Fig. 4.33 DFA *per a* $L_2 = \{w \in \{0,1\}^* \mid \mathsf{valor}_2(w) = \dot{3}\}$

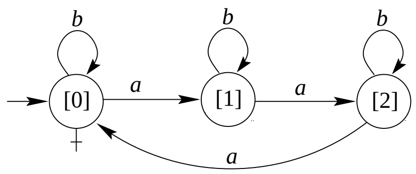

Fig. 4.34 *Autòmat M_1 que reconeix $L_1 = h^{-1}(L_2)$*

El DFA que reconeix L_1 és, doncs, $M_1 = \langle\{[0],[1],[2]\},\{a\,,b\},\delta_1,[0],\{[0]\}\rangle$ on δ_1 ve definida així:

$$\begin{array}{ll}
\delta_1([0],a) = \delta_2([0],01) = [1]\ , & \delta_1([0],b) = \delta_2([0],1001) = [0] \\
\delta_1([1],a) = \delta_2([1],01) = [2]\ , & \delta_1([1],b) = \delta_2([1],1001) = [1] \\
\delta_1([2],a) = \delta_2([2],01) = [0]\ , & \delta_1([2],b) = \delta_2([2],1001) = [2]\,.
\end{array}$$

Aquest autòmat, el graf del qual apareix a la figura 4.34, és un comptador de a's mòdul tres. Per tant, el llenguatge $h^{-1}(L_2)$ cercat és, en definitiva, $\{w \in \{a\,,b\}^* | \ \exists n \geqslant 0 \quad |w|_a = 3n\}$.

4.8 Llenguatges no regulars

Els llenguatges regulars constitueixen la família de llenguatges més bàsica de les estudiades per la teoria de llenguatges formals. Més endavant, al capítol 6, demostrarem que es tracta d'una subfamília de la dels llenguatges incontextuals. Hi ha, tanmateix, llenguatges incontextuals que no són regulars. Això és degut, essencialment, al fet que per reconèixer una àmplia varietat de CFL cal disposar de recursos de memòria més potents que la memòria finita dels DFA. Podem prendre com a exemple d'aquests llenguatges el definit a l'exemple 2.3.

Exemple 4.25 El llenguatge $L = \{a^n b^n \mid n \geqslant 0\}$ no és regular.

DEMOSTRACIÓ. Suposem que existís un DFA $M = \langle Q, \Sigma, \delta, q_0, F\rangle$, amb $\Sigma = \{a\,,b\}$, que reconegués L. Com que Q hauria de ser finit, existirien dos nombres naturals diferents i i j tals que $q_0 a^i = q_0 a^j$. Considerem els dos mots $x = a^i b^i$ i $y = a^j b^i$. És clar que $x \in L$ i que, en canvi, $y \notin L$. Però l'autòmat M és incapaç de diferenciar-los, ja que

$$q_0 x = q_0(a^i b^i) = (q_0 a^i)b^i = (q_0 a^j)b^i = q_0(a^j b^i) = q_0 y.$$

Concloem que tal autòmat M no pot existir.

Aquesta manera de demostrar la no-regularitat de determinats llenguatges serà generalitzada al capítol 7 en forma del que anomenem *lemes de bombament*.

Exercicis

4.1 Construïu DFA per als llenguatges regulars següents. Considereu-los tots definits sobre l'alfabet $\{a\,,b\}$.

1. $\{\lambda, aab, aaabab\}$

2. El conjunt dels mots tals que si comencen per aa llavors no contenen dues b's seguides
3. El conjunt dels mots en què tot parell de a's adjacents apareix davant (no necessàriament de manera consecutiva) d'algun parell de b's adjacents
4. El conjunt dels mots en què cada símbol b és immediatament precedit i seguit d'un símbol a
5. El conjunt dels mots amb un nombre parell de a's i un nombre parell de b's
6. El conjunt dels mots tals que tota cadena de cinc símbols conté com a mínim dues a's
7. El conjunt dels mots que no contenen cap prefix en a^+b^+ de longitud parella
8. El conjunt dels mots en què tot prefix de longitud més gran o igual que tres té un nombre parell de a's o un nombre parell de b's
9. El conjunt dels mots que contenen un nombre parell de vegades (comptant-hi les encavalcades) el submot aba (p. ex., els mots $ababaaba$ i $abababa$ contenen tots dos tres vegades aba.)

4.2 Doneu DFA per als llenguatges següents definits sobre alfabets numèrics.
1. $\{x2y \mid x, y \in \{0,1\}^* \wedge \mathsf{valor}_2(x) + \mathsf{valor}_2(y) = \dot{3}\}$
2. $\{w \in \{0,1,2,3,4\}^* \mid \mathsf{valor}_5(w) = \dot{5}\}$

4.3 Demostreu la regularitat dels llenguatges següents sobre l'alfabet $\{a, b\}$.
1. $\{xyx^{\mathrm{R}} \mid x, y \in \{a,b\}^+\}$
2. $\{xyy^{\mathrm{R}}z \mid x, y, z \in \{a,b\}^+\}$
3. El conjunt dels mots en què tot prefix de longitud parella té el mateix nombre de a's que de b's
4. El conjunt de mots sobre $\Sigma = \{a, b\}$ que no contenen cap submot repetit dos cops consecutius, és a dir, que no són de la forma $xyyz$ per a cap terna de mots $x, z \in \Sigma^*$ i $y \in \Sigma^+$

4.4 Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ un NFA qualsevol. Definim el conjunt d'estats *accessibles* i el conjunt d'estats *coaccessibles* d'aquest autòmat, i els representem per $\mathsf{A}(M)$ i per $\mathsf{C}(M)$, respectivament, de la manera següent:

$$\mathsf{A}(M) = \{q \in Q \mid \exists x \in \Sigma^* \ \exists p \in I \ \ q \in px\}$$
$$\mathsf{C}(M) = \{p \in Q \mid \exists y \in \Sigma^* \ \exists q \in F \ \ q \in py\}.$$

Doneu algorismes per calcular els conjunts d'estats accessibles i coaccessibles d'un NFA donat.

4.5 Demostreu que el llenguatge dels mots sobre l'alfabet $\{0,1\}$ que no tenen cap submot de longitud més gran o igual que tres que representi en binari un nombre múltiple de tres, és un llenguatge finit.

4.6 Es defineix el conjunt de *prefixos* d'un llenguatge L com el format pels mots que són prefixos de mots de L. De manera semblant, definim els llenguatges dels *sufixos* i dels *factors*. Formalment és el següent:

$$\mathsf{prefixos}(L) = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \ \ xy \in L\} = \{x \in \Sigma^* \mid x\Sigma^* \cap L \neq \varnothing\},$$
$$\mathsf{sufixos}(L) = \{y \in \Sigma^* \mid \exists x \in \Sigma^* \ \ xy \in L\} = \{y \in \Sigma^* \mid \Sigma^* y \cap L \neq \varnothing\} \text{ i}$$
$$\mathsf{factors}(L) = \{w \in \Sigma^* \mid \exists x, y \in \Sigma^* \ \ xwy \in L\} = \{w \in \Sigma^* \mid \Sigma^* w \Sigma^* \cap L \neq \varnothing\}.$$

Sigui L el llenguatge $\{a^{2i}b^{3j}a^{2k+1} \mid i, j, k \geqslant 0\}$. Construïu DFA per als llenguatges $\mathsf{prefixos}(L)$, $\mathsf{sufixos}(L)$ i $\mathsf{factors}(L)$.

4.7 Donat un llenguatge L qualsevol, definim el seu *centre* com el llenguatge

$$\mathsf{centre}(L) = \{x \in \Sigma^* \mid (x\Sigma^* \cap L) \text{ és infinit}\},$$

és a dir, el conjunt dels prefixos de mots de L que es poden completar una infinitat de cops en L. Es demana:
1. Donat el llenguatge $L = \{a^{2i}b^{3j}a^{2k+1} \mid i, j, k \geqslant 0\}$, construïu un DFA per a $\mathsf{centre}(L)$.
2. Doneu un llenguatge L, infinit i regular, tal que $\mathsf{prefixos}(L) \not\subseteq \mathsf{centre}(L)$.

4.8 Demostreu que si L és un llenguatge regular, aleshores també ho són els llenguatges: $\mathsf{prefixos}(L)$, $\mathsf{sufixos}(L)$, $\mathsf{factors}(L)$ i $\mathsf{centre}(L)$.

4.9 Demostreu que si L és un llenguatge regular sobre un alfabet Σ, aleshores també ho són els llenguatges següents:

1. El conjunt de mots de L que no són prefixos de cap altre mot de L
2. El conjunt de mots de L que no contenen com a prefix cap altre mot de L
3. El conjunt dels mots de Σ^* que contenen com a prefix algun mot de L
4. $\{x \in L \mid \forall y \in \Sigma^* \;\; xy \in L\}$
5. $\{xy \in \Sigma^* \mid yx \in L\}$
6. $\{x \in \Sigma^* \mid \exists y \in L \;\; |x| \leqslant |y| + 2\}$

4.10 Demostreu que la reunió d'un llenguatge no regular i un llenguatge finit és un llenguatge no regular. Trobeu, en canvi, un contraexemple que posi de manifest que la concatenació d'un llenguatge no regular amb un llenguatge finit no buit pot ser un llenguatge regular.

4.11 Demostreu que la reunió, la concatenació i l'estrella de Kleene de llenguatges no regulars poden ser llenguatges regulars.

4.12 Sigui $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un DFA qualsevol. Demostreu que els llenguatges següents són regulars.

1. $\{w \in \Sigma^* \mid \forall q \in Q \;\; qw = q\}$
2. $\{w \in \Sigma^* \mid \forall q \in Q \;\; qw = qab\}$
3. $\{w \in \Sigma^* \mid \forall q \in Q \;\; qw = qabw\}$

4.13 Siguin M_1 i M_2 dos NFA qualssevol i siguin L_1 i L_2, respectivament, els llenguatges que reconeixen. Doneu algorismes que permetin respondre les qüestions següents.

1. $L_1 \neq \varnothing$.
2. L_1 és infinit.
3. L_1 és *cofinit* (el seu complementari és finit).
4. L_1 i L_2 són disjunts.
5. L_1 està inclòs en L_2.
6. L_1 és un subconjunt estricte de L_2.

4.14 Demostreu que un llenguatge $L \subseteq \Sigma^*$ és regular si i solament si la família de llenguatges $\{L_x \mid x \in \Sigma^*\}$ és finita, on per a qualsevol x, $L_x = \{y \in \Sigma^* \mid xy \in L\}$.

Capítol 5 Minimització d'autòmats finits

Al capítol precedent hem vist com l'indeterminisme és una eina que aporta, en molts casos, senzillesa i concisió al disseny d'autòmats finits. L'equivalència amb els DFA a través d'un procediment algorísmic de determinització ens assegura la *implementabilitat* dels autòmats dissenyats així. De tota manera, també hem vist que aquest procés de determinització pot fer créixer el nombre d'estats exponencialment, en el cas pitjor. És evident que preferirem els autòmats *petits* als *grans*; per tant, sembla lògic preguntar-se davant un DFA qualsevol, si n'existeix algun altre de talla més petita que hi sigui equivalent i, en tal cas, com construir-lo a partir del que ja disposem. En aquest capítol respondrem aquesta pregunta presentant un algorisme que permet, donat un autòmat determinista, trobar-ne un altre que reconeix el mateix llenguatge amb un nombre *mínim* d'estats. Addicionalment, veurem que aquest *autòmat mínim* és únic i resoldrem el problema de l'equivalència de llenguatges regulars, fent èmfasi en la importància d'aquesta propietat.

5.1 Minimització d'un DFA

La idea bàsica que permet reduir el nombre d'estats d'un autòmat determinista és eliminar un estat quan ja n'hi ha un altre que fa la mateixa funció. Dos estats fan la mateixa funció quan qualsevol entrada dóna el mateix resultat tant si es parteix d'un estat com de l'altre. Diem que els dos estats són, en aquest cas, *indistingibles*. Cal tenir en compte que, quan ens referim al resultat que dóna una entrada a partir d'un estat, ens limitem a un dels dos que tenen rellevància: que s'accedeixi a un estat acceptador o que s'accedeixi a un estat no acceptador. L'eliminació d'un estat ha de comportar que les entrades que hi arribaven hauran de ser readreçades a l'estat indistingible romanent. De fet, pot haver-hi una pluralitat d'estats que siguin indistingibles d'un de donat. N'hi ha prou, doncs, de conservar un estat per a cada classe d'indistingibles. Formalitzarem a continuació aquests conceptes.

DEFINICIÓ. Sigui $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un DFA. Dos estats qualssevol de M, p i q, diem que són *indistingibles*, o que p és *equivalent* a q, i ho representem per $p \equiv q$, quan

$$\forall w \in \Sigma^* \quad p \cdot w \in F \iff q \cdot w \in F\,.$$

Exemple 5.1 Els estats D i E de l'autòmat de la figura 5.1 són indistingibles ja que, per a tot mot w, es compleix

$$D \cdot w \in F \iff E \cdot w \in F \iff w \in b^* a\{a\,, b\}^*\,.$$

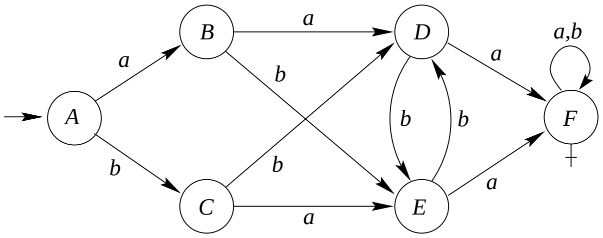

Fig. 5.1 *Un* DFA *que conté estats indistingibles*

Observeu que la relació d'indistingibilitat és una relació d'equivalència. La representarem per E_M. Com a tal, indueix una partició del conjunt d'estats en classes d'estats indistingibles. Considerant aquestes classes com a estats individuals, obtenim l'*autòmat quocient*, definit de la manera següent.

Definició. Sigui $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un DFA. Anomenarem *autòmat quocient de M per* E_M el DFA $\bar{M} = \langle \bar{Q}, \Sigma, \bar{\delta}, [q_0], \bar{F}\rangle$, tal que

- $\bar{Q}$ és el conjunt quocient de Q per E_M,
- $\bar{\delta}$ és la funció de transició que definirem tot seguit,
- $[q_0] \in \bar{Q}$ és la classe de q_0,
- $\bar{F} \subseteq \bar{Q}$ és la imatge de F, és a dir, $\bar{F} = \{[q] \mid q \in F\}$.

La funció de transició quocient, $\bar{\delta}: \bar{Q} \times \Sigma^* \to \bar{Q}$, es defineix, per a tot símbol a de Σ, com a $\bar{\delta}([q], a) = [\delta(q, a)]$, o bé $[q] \cdot a = [q \cdot a]$ amb la nostra notació habitual.

Observeu que la funció $\bar{\delta}$ està ben definida, és a dir, que no depèn del representant q elegit, ja que si $p \equiv q$ aleshores, per a tot símbol a de Σ, $p \cdot a \equiv q \cdot a$. En efecte, si suposem que existeix $w \in \Sigma^*$ tal que $(pa) \cdot w \in F$ i $(qa) \cdot w \notin F$, aleshores és clar que $p \cdot (aw) \in F$, i $q \cdot (aw) \notin F$, i això està en contradicció amb la hipòtesi $p \equiv q$.

Exemple 5.2 La relació d'indistingibilitat que hem definit ens dóna les classes d'equivalència següents sobre l'autòmat de la figura 5.1:

$$\{A\}\,,\ \{B,C\}\,,\ \{D,E\}\,,\ \{F\}\,.$$

Per tant, l'autòmat quocient corresponent és el de la figura 5.2.

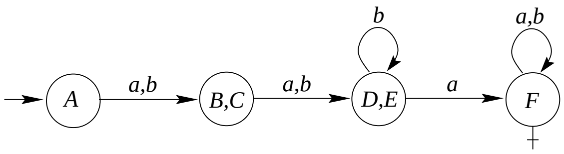

Fig. 5.2 *L'autòmat quocient corresponent al* DFA *de la figura 5.1*

L'autòmat quocient que acabem de descriure és, de fet, l'autòmat mínim que estem buscant. Les proposicions que es presenten a continuació demostren, d'una banda, que el llenguatge que reconeix l'autòmat quocient és el mateix que el que reconeix l'autòmat original i, d'altra banda, que sigui quin sigui l'autòmat de què partim, si els seus estats són tots accessibles, llavors l'autòmat quocient que obtenim és sempre el mateix, no n'hi ha cap amb un nombre menor d'estats i és l'únic amb aquest nombre d'estats.

PROPOSICIÓ. $\mathsf{L}(M) = \mathsf{L}(\bar{M})$.

DEMOSTRACIÓ. Tenim que, per a tot mot w de Σ^*,

$$w \in \mathsf{L}(\bar{M}) \Longleftrightarrow [q_0]\cdot w \in \bar{F} \Longleftrightarrow [q_0 \cdot w] \in \bar{F} \Longleftrightarrow q_0 \cdot w \in F \Longleftrightarrow w \in \mathsf{L}(M)\,,$$

on la primera i la quarta equivalències provenen de la definició de llenguatge reconegut per $\bar{M}$ i M, respectivament, la segona prové de la definició de $\bar{\delta}$ i la tercera de la definció de $\bar{F}$. □

PROPOSICIÓ. *El nombre d'estats de $\bar{M}$ és menor o igual que el de M, i quan és igual es té $M = \bar{M}$.*

DEMOSTRACIÓ. La primera part de l'enunciat és conseqüència directa de la construcció, ja que els estats de $\bar{M}$ s'han obtingut per agrupació dels de M. Quant a la segona part, quan M i $\bar{M}$ tenen el mateix nombre d'estats, l'epimorfisme canònic que transforma M en $\bar{M}$ és un isomorfisme. En altres paraules, cada classe d'equivalència està constituïda exactament per un estat. Es tracta, doncs, que per a cada estat q de M hi ha l'estat $\{q\}$ de $\bar{M}$, i tota transició $q \cdot w = p$ de M es converteix en una transició $\{q\} \cdot w = \{p\}$ de $\bar{M}$. □

PROPOSICIÓ. *Si M_1 i M_2 són dos autòmats que reconeixen un mateix llenguatge, i si tots els estats d'aquests dos autòmats són accessibles, aleshores els seus autòmats quocient respectius, $\bar{M}_1$ i $\bar{M}_2$, són iguals.*

DEMOSTRACIÓ. Siguin $\bar{M}_1 = \langle Q_1, \Sigma, \delta_1, p_0, F_1\rangle$ i $\bar{M}_2 = \langle Q_2, \Sigma, \delta_2, q_0, F_2\rangle$. De la construcció de $\bar{M}_1$ resulta que tots els seus estats són distingibles

$$\forall p_i, p_j \in Q_1 \quad p_i \neq p_j \implies \exists z \in \Sigma^* \, \neg(p_i z \in F_1 \iff p_j z \in F_1),$$

és a dir, almenys hi ha un mot que a partir d'un d'ells duu a un estat acceptador i a partir de l'altre no hi duu. El mateix pot dir-se dels estats de Q_2.

Podem afirmar ara el següent:

$$\forall x, y \in \Sigma^* \quad p_0 x = p_0 y \iff q_0 x = q_0 y. \tag{1}$$

En efecte. Altrament existirien mots x i y tals que

$$p_0 x = p_0 y \quad \text{i} \quad q_0 x \neq q_0 y \qquad \text{(o a l'inrevés)}.$$

Com que $q_0 x$ i $q_0 y$ són estats distingibles diferents, hi ha algun mot z que els distingeix, així, per exemple, $q_0 x z \in F_2$ i $q_0 y z \notin F_2$. En canvi, $p_0 x z = p_0 y z$, és a dir, els dos autòmats no reconeixerien el mateix llenguatge.

La condició anterior permet definir un isomorfisme entre els autòmats $\bar{M}_1$ i $\bar{M}_2$. Considerem una enumeració arbitrària dels estats de Q_1 que manté p_0 com a estat inicial. Sigui $Q_1 = \{p_0, p_1, \ldots, p_n\}$. Siguin $w_1, \ldots, w_n$ mots tals que $p_0 w_1 = p_1, \ \ldots, \ p_0 w_n = p_n$. Aquests mots existeixen perquè tots els estats de $\bar{M}_1$ són accessibles. Enumerem ara els estats de Q_2, començant per l'inicial q_0, de la manera següent: $q_0 w_1 = q_1, \ \ldots, \ q_0 w_n = q_n$. En virtut de 1, tots els estats q_i obtinguts així són diferents, és a dir, l'aplicació $p_i \mapsto q_i$ és injectiva. Com que igualment hauríem pogut construir una injecció en sentit contrari, de Q_2 a Q_1, resulta forçós que els dos autòmats tinguin el mateix nombre d'estats. Per tant, l'aplicació que estem considerant és una bijecció. Anem a veure que es tracta d'un isomorfisme, és a dir

$$\forall i, j\ \forall w \in \Sigma^* \quad p_i w = p_j \Longrightarrow q_i w = q_j .$$

En efecte, $p_i w = p_0 w_i w$ i tenim $p_0 w_i w = p_0 w_j$. Per tant, en virtut de 1 resulta $q_0 w_i w = q_0 w_j$, és a dir que $q_i w = q_j$. □

Com a resultat de les proposicions anteriors podem enunciar el teorema següent.

TEOREMA. *Per a tot llenguatge regular L existeix un únic autòmat determinista de mínim nombre d'estats que el reconeix. Aquest autòmat és el que s'obté a partir de qualsevol* DFA *que reconegui L i que tingui tots els estats accessibles, fent-ne el quocient per la relació d'indistingibilitat.*

5.2 Algorisme de minimització

Acabem de veure que, per trobar l'autòmat mínim associat a un DFA donat, en tenim prou a fer-ne el quocient per la relació d'indistingibilitat. Ara l'únic problema és construir la partició associada a aquesta relació. Com que la seva definició fa intervenir un conjunt infinit de condicions (una per a cada mot w), no sembla evident com decidir quan $p \equiv q$, per a p i q donats. A continuació veurem una manera d'aconseguir-ho.

DEFINICIÓ. Donats $p, q \in Q$ i un natural k, diem que p i q són *k-indistingibles* i ho representem per $p \equiv_k q$ quan

$$\forall w \in \Sigma^* \quad |w| \leqslant k \implies (pw \in F \Longleftrightarrow qw \in F) .$$

És a dir, quan no és possible diferenciar el comportament dels dos estats sense utilitzar mots de longitud superior a k.

És clar que per a qualsevol k la relació de k-indistingibilitat és també una relació d'equivalència. La representarem per E_M^k. A més, aquestes relacions E_M^k ens permeten redefinir de manera senzilla la relació E_M. Aquest fet es troba enunciat a la proposició següent, la demostració de la qual és immediata.

PROPOSICIÓ. *Per a tot parell d'estats p i q,* $p \equiv q \iff \forall k \geqslant 0\ \ p \equiv_k q$.

Les relacions de k-indistingibilitat també es poden caracteritzar de manera recurrent. Això ens permetrà calcular-les algorísmicament.

PROPOSICIÓ. *Per a tot parell d'estats p i q, i per a tot natural k, la k-indistingibilitat es caracteritza recurrentment de la manera següent:*

$$k = 0. \qquad p \equiv_0 q \iff (p \in F \ \Leftrightarrow\ q \in F)$$

$$k > 0. \qquad p \equiv_k q \iff p \equiv_{k-1} q \ \wedge\ \forall a \in \Sigma\ \ p \cdot a \equiv_{k-1} q \cdot a .$$

DEMOSTRACIÓ. Representem per Σ^k el conjunt de tots els mots sobre Σ de longitud k i per $\Sigma^{\leqslant k}$ el conjunt de mots sobre Σ de longitud menor o igual a k. Separem la demostració en dos casos dependents del valor de k.

a) $k = 0$

$$\begin{aligned} p \equiv_0 q &\Longleftrightarrow \forall w \in \Sigma^{\leqslant 0} \ \ (p \cdot w \in F \Leftrightarrow q \cdot w \in F) \\ &\Longleftrightarrow (p \in F \Leftrightarrow q \in F) \end{aligned}$$

b) $k > 0$

($\Rightarrow$) En primer lloc, del fet que $\forall w \in \Sigma^{\leqslant k} \ \ p \cdot w \in F \Longleftrightarrow q \cdot w \in F$, deduïm com a cas particular que $p \equiv_{k-1} q$. En segon lloc, per a qualsevol $a \in \Sigma$ i per a qualsevol $w \in \Sigma^{\leqslant k-1}$ es té

$$p \cdot (aw) \in F \Longleftrightarrow q \cdot (aw) \in F,$$

és a dir, $(pa) \cdot w \in F \Longleftrightarrow (qa) \cdot w \in F$, que és el mateix que $p \cdot a \equiv_{k-1} q \cdot a$.

($\Leftarrow$) Sigui $w \in \Sigma^{\leqslant k}$. Si $|w| < k$, com que $p \equiv_{k-1} q$, es té per definició que $p \cdot w \in F \Longleftrightarrow q \cdot w \in F$. Si, en canvi, $|w| = k$, aleshores $w = aw_1$ amb $|w_1| = k - 1$. Com que per hipòtesi es té $p \cdot a \equiv_{k-1} q \cdot a$, aleshores

$$\begin{aligned} (pa) \cdot w_1 \in F \Leftrightarrow (qa) \cdot w_1 \in F &\Longrightarrow p \cdot (aw_1) \in F \Leftrightarrow q \cdot (aw_1) \in F \\ &\Longrightarrow p \cdot w \in F \Leftrightarrow q \cdot w \in F \, . \end{aligned}$$

Dels dos casos precedents, se'n dedueix $p \equiv_k q$. □

Fins ara hem posat de manifest la correspondència entre les relacions d'equivalència E_M i E_M^k, i hem vist com es podrien calcular recursivament les classes de k-indistingibilitat. Finalment, la proposició següent ens diu que no ens caldrà calcular totes les relacions E_M^k.

PROPOSICIÓ. *Existeix un natural* $m \leqslant |Q| - 1$ *tal que* $\forall k \geqslant m \quad \mathsf{E}_M^m = \mathsf{E}_M^k$.

DEMOSTRACIÓ. El fet que dos estats k-equivalents hagin de ser prèviament $(k-1)$-equivalents fa que les relacions E_M^k siguin refinaments successius de E_M^0, la qual simplement separa els estats de l'autòmat en dues classes: F i $Q{-}F$. Com que el nombre d'estats de l'autòmat és finit ha d'existir un m més petit que aquest nombre tal que $\mathsf{E}_M^m = \mathsf{E}_M^{m+1}$. En aquest cas es té

$$\begin{aligned} \forall p, q \in Q, \ p \equiv_{m+2} q &\Longleftrightarrow p \equiv_{m+1} q \ \wedge \ \forall a \in \Sigma \ p \cdot a \equiv_{m+1} q \cdot a \\ &\Longleftrightarrow p \equiv_m q \ \wedge \ \forall a \in \Sigma \ p \cdot a \equiv_m q \cdot a \\ &\Longleftrightarrow p \equiv_{m+1} q \end{aligned}$$

Per tant, $\mathsf{E}_M^{m+1} = \mathsf{E}_M^{m+2}$ i, en general, $\mathsf{E}_M^m = \mathsf{E}_M^k$ per a tot $k \geqslant m$. □

COROL·LARI. *Amb les notacions utilitzades és clar que* $\mathsf{E}_M = \mathsf{E}_M^m$.

Els resultats de les tres proposicions anteriors ens permeten el càlcul de les classes d'indistingibilitat i, per tant, la construcció efectiva de l'autòmat mínim. La figura 5.3 reflecteix a grans trets els passos bàsics d'aquest algorisme de minimització.

algorisme autòmat_mínim (M: un DFA $\langle Q, \Sigma, \delta, q_0, F\rangle$)
  retorna ($\bar{M}$: un DFA)
  variables C_{ant}, C, k **fvars**
  $C_{\text{ant}} :=$ classes_de_Q_per_a E^0_M;
  $C :=$ classes_de_Q_per_a E^1_M;
  $k := 2$;
  mentre $C_{\text{ant}} \neq C$ **fer**
    $C_{\text{ant}} := C$;
    $C :=$ classes_de_Q_per_a E^k_M;
    $k := k + 1$
  fmentre
  $\bar{M} :=$ autòmat_quocient_de_M_per_la_relació E^k_M;
  retorna $\bar{M}$
falgorisme.

Fig. 5.3 *Algorisme de minimització*

Observeu que les proposicions demostrades anteriorment garanteixen l'acabament i la correctesa de l'algorisme de minimització. Vegem-ne ara el funcionament amb alguns exemples.

Exemple 5.3 Considereu el DFA M de la figura 5.4.

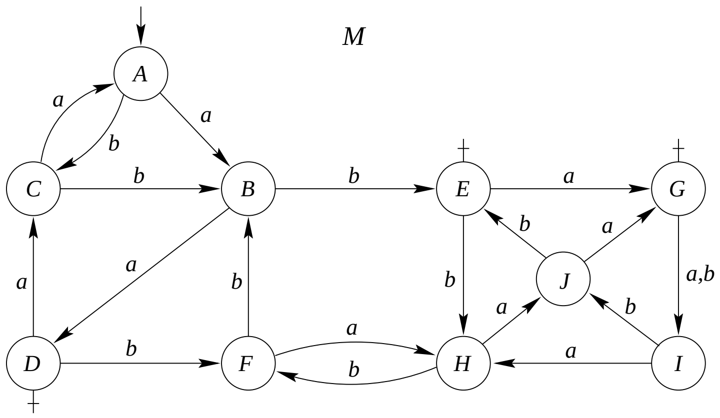

Fig. 5.4 *Un* DFA *que no és mínim*

A l'hora d'aplicar l'algorisme de minimització ens serà més còmode treballar amb la taula de transicions, reflectida a la taula 5.1.

Apliquem l'algorisme de minimització, pas a pas, per veure com funciona. Les classes d'equivalència per a E^0_M són les que separen els estats acceptadors dels que no ho són,

$$\{A, B, C, F, H, I, J\} \qquad \text{i} \qquad \{D, E, G\}\,.$$

Taula 5.1 *Taula de transicions del* DFA *de la figura 5.4*

	a	b
A	B	C
B	D	E
C	A	B
$\dagger\, D$	C	F
$\dagger\, E$	G	H
F	H	B
$\dagger\, G$	I	I
H	J	F
I	H	J
J	G	E

Calculem ara les classes per a E^1_M i obtenim

$$\{A, C, F, H, I\}, \quad \{B, J\}, \quad \{D, G\} \quad \text{i} \quad \{E\}\,.$$

Com que $\mathsf{E}^0_M \neq \mathsf{E}^1_M$, calculem les classes per aE^2_M,

$$\{A, H\}, \quad \{C, F, I\}, \quad \{B, J\}, \quad \{D, G\} \quad \text{i} \quad \{E\}\,.$$

Com que $\mathsf{E}^1_M \neq \mathsf{E}^2_M$, calculem les classes per a E^3_M,

$$\{A, H\}, \quad \{C, F, I\}, \quad \{B, J\}, \quad \{D, G\} \quad \text{i} \quad \{E\}\,,$$

i ens trobem amb $\mathsf{E}^3_M = \mathsf{E}^2_M$. Així, l'autòmat determinista mínim buscat és l'autòmat de cinc estats de la figura 5.5 que correspon a l'autòmat quocient de M per E^3_M.

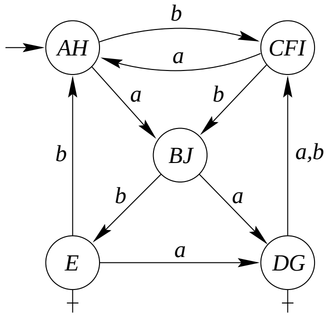

Fig. 5.5 *El* DFA *mínim equivalent a l'autòmat de la figura 5.4*

Exemple 5.4 Presentarem a continuació un exemple complet de construcció d'un DFA mínim. Reprendrem el llenguatge L, definit a l'exemple 4.3, format pels mots sobre l'alfabet $\{a, b\}$ que contenen algun parell de símbols a separats per una cadena de longitud múltiple de tres i construirem el DFA mínim corresponent d'una manera alternativa a les proposades fins ara. Observeu que una formalització possible d'aquest llenguatge és la següent:

$$L = \{w \in \Sigma^* \mid \exists x, y, z \in \Sigma^* \; \exists n \geqslant 0 \;\; w = xayaz \,\wedge\, |y| = 3n\,\}\,.$$

Per tant, podem pensar el llenguatge L com a concatenació de tres llenguatges més, $L = L_1 L_2 L_3$, on L_1 és el llenguatge dels mots que acaben en a, L_2 és el llenguatge dels mots de longitud múltiple de tres i L_3 és el llenguatge dels mots que comencen per a. Els autòmats M_1, M_2 i M_3 de la figura 5.6 són autòmats senzills que reconeixen els

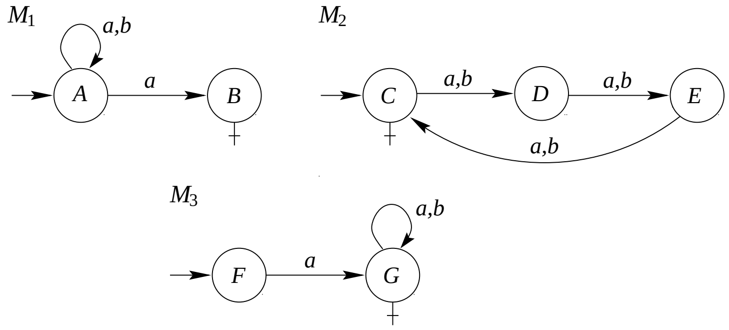

Fig. 5.6 *Autòmats finits que reconeixen els llenguatges L_1, L_2 i L_3*

llenguatges L_1, L_2 i L_3, respectivament. Concatenant aquests tres autòmats obtindrem fàcilment l'autòmat buscat.

L'NFA que resulta de la concatenació dels autòmats M_1 i M_2 de la figura 5.6 està representat a la figura 5.7.

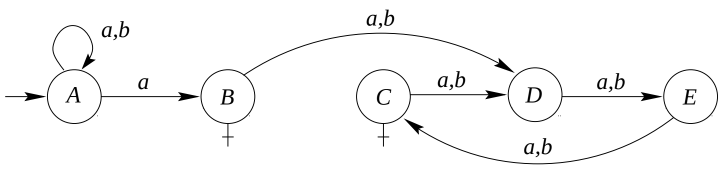

Fig. 5.7 NFA *que reconeix el llenguatge $L_1 L_2$*

L'NFA que resulta de la concatenació de l'autòmat de la figura 5.7 amb l'autòmat M_3 de la figura 5.6 està representat a la figura 5.8.

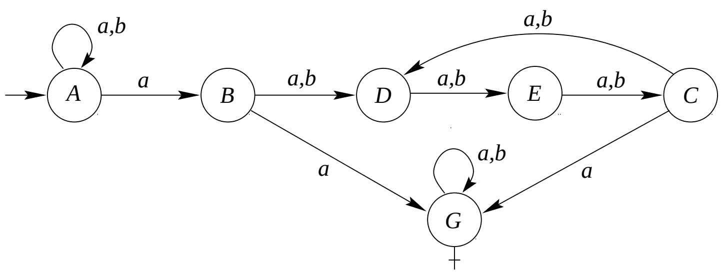

Fig. 5.8 NFA *que reconeix el llenguatge $L_1 L_2 L_3 = L$*

La determinització d'aquest darrer autòmat dóna lloc a un DFA de dotze estats, representat a la part A de la taula 5.2. La part B de la mateixa taula és igual que la A, amb els estats canviats de nom per tal de facilitar el procés de minimització.

Fem notar que l'estat de nom F de la taula A de transicions precedent *resumeix* tots els estats del DFA que contenen l'estat G de l'NFA de la figura 5.8. Això porta implícit un primer pas de minimització que és correcte, ja que tots aquests estats serien acceptadors i, per tant, $0\_equivalents$ i l'algorisme de minimització no els separaria mai, ja que formen un subgraf tancat per causa de l'arc amb qualsevol símbol de l'estat G cap a ell mateix. Adonar-nos d'un fet com aquest pot estalviar molta feina en el procés

Taula 5.2 *Determinització de l'NFA de la figura 5.8*

Taula A

	a	b
A	AB	A
AB	F	AD
$\dagger F$	F	F
AD	ABE	F
ABE	F	ACD
AE	ABC	AC
ACD	F	ADE
ABC	F	AD
AC	F	AD
ADE	$ABCE$	ACE
$ABCE$	F	ACD
ACE	F	ACD

Taula B

	a	b
1	2	1
2	3	4
†3	3	3
4	5	6
5	3	7
6	8	9
7	3	10
8	3	4
9	3	4
10	11	12
11	3	7
12	3	7

de determinització previ a l'obtenció de l'autòmat mínim (sobretot si el fem a mà!). En aquest mateix exemple, ens hauria sortit un DFA de setze estats.

El darrer pas per arribar a la solució serà minimitzar l'autòmat determinista de dotze estats de la taula B seguint l'algorisme de minimització presentat en aquesta secció. Les classes d'equivalència per a E_M^0 són les que separen els estats acceptadors dels no acceptadors,

$$\{1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12\} \quad \text{i} \quad \{3\}\,.$$

Calculem ara les classes per a E_M^1 i obtenim

$$\{1, 4, 6, 10\}, \quad \{2, 5, 7, 8, 9, 11, 12\} \quad \text{i} \quad \{3\}\,.$$

Com que $\mathsf{E}_M^0 \neq \mathsf{E}_M^1$, calculem les classes per a E_M^2,

$$\{1, 4\}, \quad \{6, 10\}, \quad \{2, 8, 9\}, \quad \{5, 11, 12\}, \quad \{7\} \quad \text{i} \quad \{3\}\,.$$

Com que $\mathsf{E}_M^1 \neq \mathsf{E}_M^2$, calculem les classes per a E_M^3,

$$\{1\}, \quad \{4\}, \quad \{6\}, \quad \{10\}, \quad \{2, 8, 9\}, \quad \{5, 11, 12\}, \quad \{7\} \quad \text{i} \quad \{3\}\,.$$

Com que $\mathsf{E}_M^2 \neq \mathsf{E}_M^3$, calculem les classes per a E_M^4, i ens trobem amb $\mathsf{E}_M^3 = \mathsf{E}_M^4$. Així, l'autòmat determinista mínim per reconèixer L és l'autòmat de vuit estats de la figura 5.9, en què hem etiquetat amb la lletra A l'estat format per $\{2, 8, 9\}$ i amb la lletra B l'estat $\{5, 11, 12\}$. Adoneu-vos que resulta, tal com era previsible, el mateix autòmat que havíem presentat a la figura 4.3 del capítol 4.

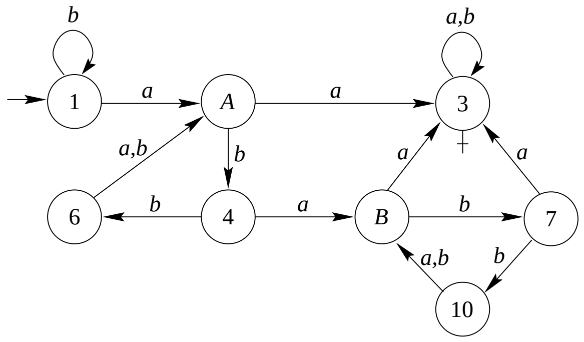

Fig. 5.9 DFA *mínim que reconeix el llenguatge L*

5.3 Sobre la talla del DFA mínim

L'indeterminisme en autòmats finits és una eina molt útil per dissenyar autòmats de manera senzilla. L'algorisme de determinització construeix DFA equivalents, però pot fer créixer el nombre d'estats de manera exponencial. L'aplicació posterior de l'algorisme de minimització, presentat en aquest capítol, ens permet reduir aquests DFA als de mínim nombre d'estats. Ara podríem preguntar-nos si aquesta reducció a l'autòmat mínim és sempre significativa, en altres paraules, si evita sempre la possible explosió exponencial en el nombre d'estats de la versió determinista enfront de la indeterminista.La resposta és negativa en el cas pitjor, ja que hi ha llenguatges per als quals el DFA mínim té un nombre d'estats exponencial respecte del nombre d'estats del seu equivalent indeterminista. Estudiem-ne un exemple concret.

Considereu l'NFA de la figura 5.10 que reconeix el llenguatge dels mots, sobre l'alfabet $\Sigma = \{a\,,b\}$, que tenen un símbol a a l'antepenúltima posició. Anomenarem L_2 aquest llenguatge. Es té $L_2 = \Sigma^* a \Sigma^2$.

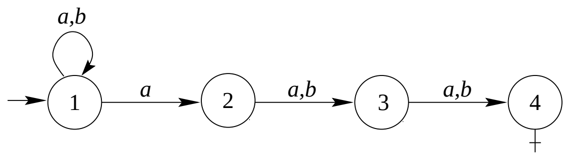

Fig. 5.10 NFA *que reconeix el llenguatge* $L_2 = \Sigma^* a \Sigma^2$

És immediat veure com es pot generalitzar la construcció de la figura 5.10 a NFA que reconeguin els llenguatges $L_3 = \Sigma^* a \Sigma^3$, $L_4 = \Sigma^* a \Sigma^4$, etc. Simplement es tracta d'anar allargant la cadena d'estats que porten cap a l'estat acceptador amb qualsevol símbol, a mesura que anem anticipant la posició del símbol a a reconèixer.

En general, l'NFA que reconeix el llenguatge $L_n = \Sigma^* a \Sigma^n$, per a un n donat, té $n+2$ estats (vegeu la figura 5.11), però què passa amb el DFA mínim corresponent? La resposta és que té 2^{n+1} estats. Vegem formalment que no pot tenir-ne menys (deixem per al lector, a l'exercici 5.1, la demostració de l'existència de tal autòmat).

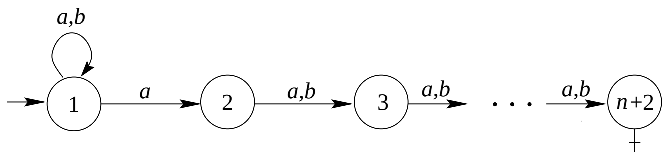

Fig. 5.11 NFA *que reconeix el llenguatge* $L_n = \Sigma^* a \Sigma^n$

PROPOSICIÓ. *Donat el llenguatge* $L_n = \Sigma^* a \Sigma^n$ *sobre* $\Sigma = \{a, b\}$, *on* $n \geqslant 0$, *se satisfà que tot* DFA *que el reconeix té, com a mínim,* 2^{n+1} *estats.*

DEMOSTRACIÓ. Procedirem per reducció a l'absurd. Suposem que existeix un autòmat $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ tal que $\mathsf{L}(M) = L_n$ i $\|Q\| < 2^{n+1}$. Per cada mot x de Σ^{n+1}, sigui $q_x = q_0 \cdot x$. Com que hi ha 2^{n+1} mots diferents a Σ^{n+1} i només $\|Q\| < 2^{n+1}$ estats, han d'existir dos mots, $x, y \in \Sigma^{n+1}$, diferents, per als quals $q_x = q_y$. Sigui z el sufix comú més llarg de x i y. Aleshores existeixen x', $y' \in \Sigma^*$ tals que o bé $x = x'az$ i $y = y'bz$, o bé $x = x'bz$ i $y = y'az$. Intercanviant el rol de x i y si és necessari, podem assumir, sense pèrdua de generalitat, que $x = x'az$ i $y = y'bz$. Sigui $m = n - |z|$. Considereu ara els mots $w_1 = x'aza^m$ i $w_2 = y'bza^m$. Clarament $w_1 \in L_n$ (ja que té el símbol a en la posició n-èsima començant pel final) i $w_2 \notin L_n$ (ja que té el símbol b en aquesta mateixa posició). Tanmateix, l'autòmat M és incapaç de distingir-los, ja que

$$q_0 \cdot w_1 = q_0 \cdot (xa^m) = (q_0 \cdot x) \cdot a^m = q_x \cdot a^m = q_y \cdot a^m = (q_0 \cdot y) \cdot a^m = q_0 \cdot (ya^m) = q_0 \cdot w_2 \,.$$

Arribem a la contradicció que M accepta o rebutja tots dos mots. Per tant, podem concloure que no existeix l'autòmat M suposat. □

Si generalitzem la proposició anterior sobre un alfabet Σ amb s símbols, la fita inferior en el nombre d'estats és s^{n+1}, com es pot comprovar fàcilment.

Exemple 5.5 Per al cas particular de $n = 2$, un NFA que reconeix L_2 és el de la figura 5.10. Anem a comprovar que el DFA mínim equivalent és un autòmat de $2^3 = 8$ estats.

Si determinitzem l'NFA de la figura 5.10, obtenim directament el DFA de vuit estats de la figura 5.12.

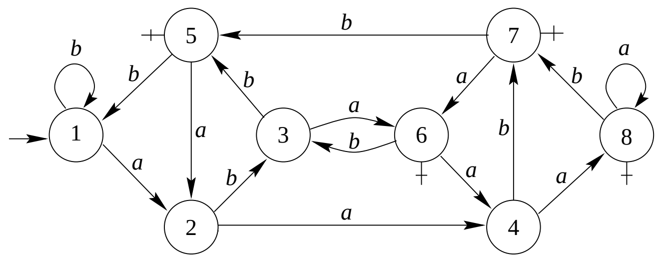

Fig. 5.12 DFA *que reconeix el llenguatge* $L_2 = \Sigma^* a \Sigma^2$

Comprovem ara que aquest autòmat és mínim.

Classes d'equivalència per a E^0_M: $\{1,2,3,4\}$ i $\{5,6,7,8\}$.

Classes d'equivalència per a E^1_M: $\{1,2\}$, $\{3,4\}$, $\{5,6\}$ i $\{7,8\}$.

Classes d'equivalència per a E^2_M: $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{6\}$, $\{7\}$ i $\{8\}$.

Evidentment, $\mathsf{E}^3_M = \mathsf{E}^2_M$ i ja hem acabat la minimització. Per tant, l'autòmat de partida era mínim, tal com havíem previst. Fixeu-vos que l'autòmat que hem construït és el mateix DFA de vuit estats construït a l'exemple 4.4, amb l'única diferència de les etiquetes associades a cada estat. No costa gaire comprovar que el criteri de construcció seguit a l'exemple 4.4 ens porta a l'obtenció directa de DFA mínims per als llenguatges L_n.

Resumint, hem vist que hi ha llenguatges que necessiten autòmats finits amb *molts* estats i que, en el cas pitjor, el pas d'un autòmat no determinista al corresponent determinista mínim comporta un creixement exponencial del nombre d'estats.

5.4 Equivalència entre autòmats

Donat un model d'especificació de llenguatges, per exemple, el dels autòmats finits, un problema decisional típic és determinar l'equivalència de dos representants qualssevol d'aquest model. Els algorismes generals de determinització i minimització d'autòmats finits, juntament amb la propietat d'unicitat de l'autòmat mínim, permeten establir un marc clar de comparació d'autòmats i resoldre el problema de l'equivalència d'NFA. En efecte:

Donats dos NFA *qualssevol, M_1 i M_2, aquests són equivalents (reconeixen el mateix llenguatge) si i només si els autòmats mínims corresponents de M_1 i M_2 són el mateix autòmat.*

La decidibilitat del problema de l'equivalència entre autòmats finits és una bona propietat, associada a la família dels llenguatges regulars, que no podem extrapolar a la resta de models més complexos que veiem al llarg d'aquest curs. Per exemple, en el marc dels llenguatges incontextuals, el problema de l'equivalència de gramàtiques no és decidible, si bé quan ens restringim als llenguatges incontextuals deterministes (subfamília dels incontextuals que estudiarem al capítol 8) sí que és decidible[1]. Podem afegir que el problema de l'equivalència de programes és també indecidible. Observeu, a més, que, com ja hem posat de manifest al capítol precedent, la família dels llenguatges regulars és tancada respecte de les operacions conjuntistes clàssiques, cosa que no passa en altres famílies (recordeu, per exemple, la intersecció de CFLvista al capítol 2). El fet que en la família dels llenguatges regulars es compleixin totes aquestes propietats confereix a aquesta família una senzillesa que no tenen altres classes derivades de models de computació més complexos.

És interessant d'observar que existeix una altra demostració de la decidibilitat del problema de l'equivalència d'autòmats finits que no apel·la en absolut a la propietat d'unicitat de l'autòmat mínim. Vegem-la a continuació.

Suposem que tenim dos autòmats finits M_1 i M_2 que reconeixen, respectivament, els llenguatges L_1 i L_2. Considerem ara el llenguatge L_3 definit a partir d'ells dos de la manera següent: $L_3 = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$. És immediat comprovar que se satisfà l'equivalència següent, $L_1 = L_2 \iff L_3 = \varnothing$. Com que les operacions de reunió, intersecció i complementació són construïbles, aleshores podem trobar un DFA M_3 que reconeix el llenguatge L_3. Per tant, els autòmats M_1 i M_2 són equivalents si i només si l'autòmat M_3 no accepta cap mot. Observeu finalment que aquesta propietat és decidible, ja que és equivalent a comprovar que tots els estats accessibles de M_3 són no acceptadors.

Tot i que aquesta demostració també comporta un mètode per decidir l'equivalència d'autòmats finits, a la pràctica és millor utilitzar el mètode de comparació dels autòmats mínims ja que té una complexitat menor en el cas pitjor.

Exercicis

5.1 Trobeu DFA mínims per als autòmats construïts a l'exercici 4.1.

5.2 Donat el llenguatge $L_n = \Sigma^* a \Sigma^n$ sobre $\Sigma = \{a, b\}$, on $n \geqslant 0$. Demostreu que el DFA mínim que el reconeix té exactament 2^{n+1} estats.

[1] Aquest resultat ha estat demostrat en [Sen97].

5.3 Per a cada enter $n \geqslant 1$, definim L_n com el llenguatge format pels mots sobre l'alfabet $\{0,1\}$ tals que el seu valor binari és *una potència* de 2^n. Demostreu que l'autòmat mínim que reconeix L_n té $n+2$ estats.

5.4 Per a cada enter $n \geqslant 1$, considerem el llenguatge format per λ i pels mots sobre l'alfabet $\{0,1\}$ tals que el seu valor binari és *un múltiple* de n. Demostreu que l'autòmat mínim que reconeix aquest llenguatge té $\alpha + q$ estats, on $\alpha \geqslant 0$ i el nombre senar q són els dos únics enters tals que $n = 2^\alpha \cdot q$.

5.5 Demostreu que, si un DFA és mínim, llavors per a tot parell d'estats diferents, q_i i q_j, existeix un mot w tal que o bé $q_i w$ és un estat acceptador i $q_j w$ no ho és o bé $q_j w$ és un estat acceptador i $q_i w$ no ho és.

5.6 Demostreu que, si un DFA és mínim, llavors si hi ha algun estat acceptador q tal que per a tot símbol a de l'alfabet satisfà que $qa = q$, llavors q és l'únic estat acceptador amb aquesta propietat.

5.7 Sigui L un llenguatge regular i sigui $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ el seu DFA mínim. Demostreu que el llenguatge $L' = \{w \in \Sigma^* \mid \forall q \in Q\ \delta(q,w) \notin F\}$ està format pels mots que no són sufixos de L. Trobeu un contraexemple d'aquest resultat quan l'autòmat M no és mínim.

5.8 Associem a cada autòmat finit determinista $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ la congruència següent entre els mots de Σ^*

$$x \underset{M}{\equiv} y \ \overset{\text{def}}{\iff}\ \forall q \in Q\ qx = qy.$$

1. Demostreu que es tracta efectivament d'una congruència, és a dir, que
   1.1 és una relació d'equivalència,
   1.2 és compatible amb la concatenació, és a dir, que
   $$\forall x_1, x_2, y_1, y_2 \quad x_1 \underset{M}{\equiv} x_2 \wedge y_1 \underset{M}{\equiv} y_2 \implies x_1 y_1 \underset{M}{\equiv} x_2 y_2.$$
2. Demostreu que aquesta congruència és d'*índex finit*, és a dir, que el conjunt de les classes d'equivalència és finit.
3. Demostreu que, per a tot mot $x \in \Sigma^*$, la classe $[x]$ de mots congruents amb x és un llenguatge regular, tot definint un DFA que la reconeix.
4. Demostreu que el llenguatge reconegut per M és la reunió d'algunes de les classes d'aquesta congruència.

5.9 Donat un llenguatge qualsevol $L \subseteq \Sigma^*$, regular o no, li associem la congruència següent entre els mots de Σ^*

$$x \underset{L}{\equiv} y \ \overset{\text{def}}{\iff}\ (\forall z_1, z_2 \in \Sigma^*\quad z_1 x z_2 \in L \iff z_1 y z_2 \in L).$$

1. Demostreu que es tracta efectivament d'una congruència.
2. Demostreu que $\forall x \in \Sigma^* \quad x \in L \iff [x] \subseteq L$.
3. Demostreu que, quan la congruència associada a L és d'índex finit, el llenguatge L és regular (teorema de Myhill, 1957).

5.10 Sigui $L \subseteq \Sigma^*$ un llenguatge regular. Considerem la congruència associada a aquest llenguatge.

1. Demostreu que, per a qualsevol DFA M que reconegui L, la congruència associada a M és un *refinament* de l'associada a L. És a dir, que
   $$\forall x, y \in \Sigma^* \quad x \underset{M}{\equiv} y \implies x \underset{L}{\equiv} y.$$
2. Demostreu que, si M és l'autòmat mínim que reconeix L, la congruència associada a M coincideix amb l'associada a L.

5.11 Donat un llenguatge qualsevol $L \subseteq \Sigma^*$, regular o no, li associem la relació d'equivalència següent entre els mots de Σ^*

$$x \underset{L}{\sqsubseteq} y \ \overset{\text{def}}{\iff}\ (\forall z \in \Sigma^*\quad xz \in L \iff yz \in L).$$

1. Demostreu que es tracta efectivament d'una relació d'equivalència.
2. Demostreu que és invariant a dreta, és a dir, que
$$x \underset{L}{\equiv} y \implies \forall z \in \Sigma^* \;\; xz \underset{L}{\equiv} yz.$$
3. Demostreu que $\forall x \in \Sigma^* \quad x \in L \iff [x] \subseteq L$.
4. Demostreu que, quan la relació d'equivalència associada a L és d'índex finit, el llenguatge L és regular (teorema de Nerode, 1958).

5.12 Donat un llenguatge regular L, considerem la relació d'equivalència definida a l'exercici anterior. Construïm, a partir d'ella, el DFA que té per estats les classes d'equivalència; per estat inicial la classe $[\lambda]$; per estats acceptadors les classes que corresponen als mots que pertanyen a L; i per funció de transició la que fa correspondre a cada classe $[x]$ i a cada símbol a de l'alfabet la classe $[xa]$. Demostreu que aquest autòmat és precisament el DFA mínim que reconeix L.

Capítol 6 Expressions regulars i gramàtiques regulars

6.1 Introducció

Els llenguatges regulars han estat tractats als dos capítols anteriors a partir de la seva definició com a llenguatges reconeguts per autòmats finits. El concepte d'autòmat finit pot ser generalitzat a monoides qualssevol (més enllà dels de la forma Σ^*, formats per mots) tot mantenint-ne la definició. Donat un monoide M qualsevol, s'anomenen llenguatges *recognoscibles* sobre M, els subconjunts de M que són reconeguts per algun autòmat finit sobre M. La família d'aquests llenguatges es representa per $\mathsf{Rec}(M)$.

Exemple 6.1 Considerem el monoide $(\mathbb{Z}, +)$ dels enters amb l'addició, al qual ens referirem simplement per $\mathbb{Z}$. Un autòmat finit sobre $\mathbb{Z}$ és una estructura de la forma $\langle Q, \mathbb{Z}, \delta, q_0, F\rangle$, on Q, q_0 i F tenen el significat habitual de conjunt finit d'estats, estat inicial i subconjunt d'estats acceptadors, respectivament. Els axiomes que caracteritzen la funció de transició δ s'expressen, en aquest cas, així:

$$\forall q \in Q \quad \forall m, n \in \mathbb{Z} \qquad \begin{cases} (1) & \delta(q, 0) = q \\ (2) & \delta(q, m+n) = \delta(\delta(q, m), n). \end{cases}$$

Podem mantenir el conveni de representar la funció de transició amb el mateix signe d'operació que s'utilitza per al monoide. En aquest cas, els axiomes anteriors s'escriuen així:

$$q + 0 = q$$
$$q + (m + n) = (q + m) + n.$$

Remarquem que per definir una funció de transició qualsevol només cal especificar el conjunt finit de valors que pren sobre $Q \times \{-1, 1\}$, ja que $\mathbb{Z}$ és generat additivament per $\{-1, 1\}$. L'autòmat finit de la figura 6.1 reconeix el conjunt de nombres que són congruents amb 1 mòdul 4.

És fàcil de constatar que la família de llenguatges recognoscibles sobre aquest monoide, $\mathsf{Rec}(\mathbb{Z}, +)$, està constituïda exactament pels conjunts formats per alguna *reunió finita de progressions aritmètiques* (sense primer ni últim element) de nombres positius i negatius.

En el cas que ens ocupa, el dels llenguatges recognoscibles sobre un monoide de la forma Σ^*, Kleene [Kle56] va caracteritzar $\mathsf{Rec}(\Sigma^*)$, en termes algebraics, com la mínima família de llenguatges que era tancada respecte de les operacions de reunió, concatenació i estrella i que contenia els conjunts finits.

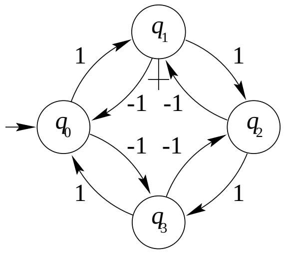

Fig. 6.1 *Un autòmat finit sobre el monoide* $(\mathbb{Z}, +)$

Aquesta definició també pot ser generalitzada a monoides qualssevol. Donat un monoide M, i un subconjunt A de M, podem definir A^* com el *submonoide generat* per A, és a dir, com el mínim submonoide que conté A. En aquestes condicions, s'anomena família de conjunts *racionals* de M, i es representa per $\mathsf{Rat}(M)$, la mínima família que conté els subconjunts finits de M i que és tancada respecte de la reunió, de l'operació pròpia del monoide i de l'operació estrella. En general, per a monoides M qualssevol, $\mathsf{Rec}(M)$ i $\mathsf{Rat}(M)$ són famílies diferents. I cap de les dues inclusions no pot ser establerta amb caràcter general. Així doncs, el que estableix el teorema de Kleene, les bases del qual desenvoluparem en aquest capítol, és que en el cas de llenguatges sobre algun alfabet Σ, $\mathsf{Rec}(\Sigma^*) = \mathsf{Rat}(\Sigma^*)$.[1]

Exemple 6.2 En el cas del monoide $(\mathbb{Z}, +)$ de l'exemple anterior, no es té la igualtat entre $\mathsf{Rec}(\mathbb{Z})$ i $\mathsf{Rat}(\mathbb{Z})$. N'hi ha prou de constatar que $\mathsf{Rat}(\mathbb{Z})$ inclou, per definició, tots els subconjunts finits de $\mathbb{Z}$, mentre que l'únic subconjunt finit de $\mathsf{Rec}(\mathbb{Z})$ és el buit. Deixem com a exercici per al lector demostrar que $\mathsf{Rec}(\mathbb{Z}) \subset \mathsf{Rat}(\mathbb{Z})$.

Exemple 6.3 Hi ha d'altres monoides, a part dels formats per mots sobre algun alfabet, que també satisfan el teorema de Kleene. Per exemple, el monoide $(\mathbb{N}, +)$, format pels nombres naturals amb l'addició. Es té $\mathsf{Rec}(\mathbb{N}) = \mathsf{Rat}(\mathbb{N})$, cosa que es dedueix directament de l'isomorfisme entre els conjunts de nombres naturals i els llenguatges sobre alfabets uniliterals.

6.2 Expressions regulars

Els llenguatges racionals sobre un alfabet Σ qualsevol, que en aquesta mateixa secció demostrarem que coincideixen amb els llenguatges regulars, admeten descripcions que utilitzen tan sols els símbols de Σ i els de les operacions bàsiques entre llenguatges. Aquestes descripcions, que anomenarem expressions regulars, especifiquen de manera directa la forma en què el llenguatge considerat pot ser descompost en una cadena finita d'operacions elementals.

[1] Un estudi de les famílies $\mathsf{Rec}(M)$ i $\mathsf{Rat}(M)$ per a monoides M qualssevol pot trobar-se al capítol III de la referència [Ber79].

DEFINICIÓ. Sigui Σ un alfabet. Anomenarem *expressió regular* sobre Σ tota cadena que satisfaci la definició recursiva següent:

1. Λ i $\varnothing$ són expressions regulars.
2. Per a cada símbol a de Σ, a és una expressió regular.
3. Si r_1 i r_2 són expressions regulars, també ho són (r_1+r_2) i $(r_1 \cdot r_2)$.
4. Si r és una expressió regular, també ho és (r^*).
5. Totes les expressions regulars sobre Σ s'obtenen aplicant 1, 2, 3 i 4.

DEFINICIÓ. Anomenarem *llenguatge associat a una expressió regular* r el llenguatge $\mathsf{L}(r)$ definit recursivament així:

1. $\{\lambda\}$, $\varnothing$ o $\{a\}$, si r és Λ, $\varnothing$ o a, respectivament.
2. $\mathsf{L}(r_1) \cup \mathsf{L}(r_2)$ o $\mathsf{L}(r_1) \cdot \mathsf{L}(r_2)$, si $r = (r_1+r_2)$ o $r = (r_1 \cdot r_2)$, respectivament.
3. $(\mathsf{L}(r_1))^*$ si $r = (r_1^*)$.

Suposarem una certa precedència sobre les operacions que ens permetrà estalviar una gran quantitat de parèntesis a l'hora d'escriure les expressions regulars. En concret, l'estrella de Kleene serà la més prioritària i la concatenació tindrà precedència sobre la unió. Així, escriurem l'expressió regular

$$(a + (b \cdot (c^*)))\,,$$

com a $a + b \cdot c^*$, sense que hi hagi cap ambigüitat. Com és habitual en les notacions multiplicatives, ometrem usualment el símbol $\cdot$ en l'escriptura de les expressions regulars. L'expressió regular anterior passa a ser, doncs, $a + bc^*$. També per comoditat, escriurem r en comptes de $\mathsf{L}(r)$ per referir-nos al llenguatge associat a una expressió regular r qualsevol; només els diferenciarem explícitament quan el context es presti a confusió.

Les expressions regulars permeten descriure de manera senzilla i entenedora llenguatges regulars com els dels exemples següents (considereu-los tots sobre l'alfabet $\Sigma = \{a, b, c\}$).

Exemple 6.4 Σ^* i Σ^+ són els llenguatges associats a

$$(a+b+c)^* \quad \text{i} \quad (a+b+c)\,(a+b+c)^*\,,$$

respectivament. Habitualment, se sol escriure r^+ en comptes de rr^*, on r és una expressió regular qualsevol.

Exemple 6.5 El llenguatge dels mots de longitud múltiple de tres és el llenguatge associat a l'expressió regular: $(\,(a+b+c)\,(a+b+c)\,(a+b+c)\,)^*$.

En canvi, hi ha d'altres llenguatges regulars per als quals és difícil donar expressions regulars intuïtives.

Exemple 6.6 Un exemple el tenim en el llenguatge dels mots, sobre $\Sigma = \{a, b, c\}$, que no contenen la subcadena abc. El problema prové de la dificultat d'expressar, en termes de reunions, concatenacions i estrelles, l'estructura d'aquest llenguatge. Observeu que, en canvi, el llenguatge complementari de l'anterior sí que es pot descriure fàcilment amb una expressió regular: per exemple, $(a+b+c)^*abc\,(a+b+c)^*$.

Donat un llenguatge regular, sempre existeixen infinites expressions regulars que el tenen com a llenguatge associat; direm que totes elles són *equivalents*. Un dels problemes que resoldrem satisfactòriament en aquesta secció és el de decidir l'equivalència de dues expressions regulars qualssevol.

Exemple 6.7 El llenguatge dels mots que tenen un nombre parell de símbols a és el llenguatge associat a l'expressió regular

$$(b+c)^* + (\,(b+c)^*\,a\,(b+c)^*\,a\,(b+c)^*\,)^*\,,$$

o, equivalentment,

$$(b+c)^*\,(a\,(b+c)^*\,a\,(b+c)^*\,)^*\,,$$

o encara,

$$(b+c+a(b+c)^*a)^*.$$

Un primer fenomen que constatarem és que, mentre que en el marc dels autòmats finits el pas d'un DFA que reconeix un cert llenguatge L al DFA que reconeix el complementari, $\overline{L}$, és una operació immediata, en el marc de les expressions regulars no és així de fàcil. Això no té res d'estrany: mentre que la família $\mathsf{Rec}(M)$ dels llenguatges reconeguts per autòmats finits és tancada respecte de la complementació —sigui quin sigui el monoide M—, no succeeix el mateix amb la família $\mathsf{Rat}(M)$ dels llenguatges racionals sobre M. Només cal considerar el monoide $(\mathbb{N},\cdot)$ dels nombres naturals amb la multiplicació, que és generat pels nombres primers. $\mathbb{N}$, doncs, no és finitament generat i no és un conjunt racional. En canvi, el seu complementari, el conjunt buit, sí que ho és. Veurem a continuació que quan el monoide considerat és Σ^*, es té $\mathsf{Rat}(\Sigma^*) = \mathsf{Rec}(\Sigma^*)$, és a dir, les expressions regulars caracteritzen exactament els llenguatges regulars.

TEOREMA. *Un llenguatge és regular si i només si és el llenguatge associat a una expressió regular.*

DEMOSTRACIÓ. La demostració original d'aquest teorema és deguda a S. Kleene [Kle56]. Per a la implicació en el primer dels sentits seguirem la demostració feta per McNaughton i Yamada [MNY60].

$(\Rightarrow)$ *A tot llenguatge regular li correspon una expressió regular.*
Sigui L un llenguatge regular i sigui $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un DFA que el reconeix. Sigui $Q = \{q_0, q_1, \ldots, q_n\}$. Per a $0 \leqslant i,j \leqslant n$ i $0 \leqslant k \leqslant n+1$, definim $L_{i,j,k}$ com el llenguatge de tots els mots de Σ^* per als quals el càlcul que comença en l'estat q_i acaba en l'estat q_j sense passar per cap estat intermedi indexat amb un nombre més gran o igual que k. Formalment, tenim

$$L_{i,j,k} = \big\{w \in \Sigma^* \;\mid\; q_iw = q_j \;\wedge\; \forall y,z \in \Sigma^+\;\forall l\,((w = yz \wedge q_iy = q_l) \Rightarrow l < k)\big\}\,.$$

En aquesta expressió, tant el prefix y com el sufix z en què considerem dividit el mot w són factors *propis* de w, (diferents de λ i del mateix w). Així doncs, l'estat q_l és estrictament intermedi entre q_i i q_j, de manera que els índexs i i j queden exclosos de la condició de ser menors que k.

Observem que, per a $k = n+1$, tenim $L_{i,j,n+1} = \{w \mid q_i w = q_j\}$. Demostrarem ara, per inducció sobre k, que cada $L_{i,j,k}$ està associat a alguna expressió regular, la qual cosa ens donarà el que volem, ja que és clar que

$$L = \bigcup_{j \mid q_j \in F} L_{0,j,n+1}\,.$$

Per a $k = 0$, la demostració és trivial, ja que

$$L_{i,j,0} = \{a \in \Sigma \mid q_i \cdot a = q_j\} \cup \{\lambda \mid i = j\},^2$$

i aquest llenguatge és un conjunt finit de mots d'un sol símbol (i, eventualment, lambda), $\{a_1, a_2, \cdots, a_m\} \subseteq \Sigma$, que té per expressió regular $a_1 + a_2 + \cdots + a_m$.

Ara observem que

$$L_{i,j,k+1} = L_{i,j,k} \cup L_{i,k,k} \cdot L^*_{k,k,k} \cdot L_{k,j,k},$$

ja que anar de q_i a q_j sense passar per cap estat intermedi d'índex més gran o igual que $k+1$ és equivalent a:

1. anar de q_i a q_j sense passar per cap estat intermedi d'índex més gran o igual que k, o bé
2. anar de q_i a q_k (sense passar per cap estat intermedi d'índex més gran o igual que k), anar de q_k a q_k (amb la mateixa restricció) un determinat nombre de vegades i, finalment, anar de q_k a q_j (sempre amb la mateixa restricció).

Com que per hipòtesi d'inducció els llenguatges que apareixen al membre dret d'aquesta igualtat estan associats a expressions regulars: $r_{i,j,k}$, $r_{i,k,k}$, $r_{k,k,k}$ i $r_{k,j,k}$, el membre esquerre ho estarà a: $r_{i,j,k} + r_{i,k,k} \cdot r^*_{k,k,k} \cdot r_{k,j,k}$.

($\Leftarrow$) *El llenguatge associat a una expressió regular és regular.*

Com que $\{\lambda\}$, $\varnothing$ i $\{a\}$ són llenguatges regulars ($\forall a \in \Sigma$), i com que la classe dels llenguatges regulars és tancada respecte de la reunió, la concatenació i l'estrella de Kleene, la implicació és immediata. □

Observeu que la primera inclusió d'aquest teorema ens dóna, a més, un algorisme per construir expressions regulars per a llenguatges regulars donats. Aquest fet ens permet solucionar el problema, descrit a l'exemple 6.6, de trobar una expressió regular per al llenguatge L dels mots sobre $\Sigma = \{a, b, c\}$ que no contenen la cadena abc. Efectivament, si partim d'un DFA que reconegui L (fàcil de construir) i apliquem l'algorisme de McNaughton-Yamada, arribarem a una expressió regular que correspon a L.

A l'exemple següent veurem com s'aplica aquest algorisme per passar d'un autòmat finit a l'expressió regular corresponent.

Exemple 6.8 Considerem l'autòmat determinista definit a la figura 6.2. El llenguatge reconegut per aquest autòmat és $L = \{w \in (a+b)^* \mid |w|_b = \dot{3} + 1\}$.

[2] L'expressió $\{\lambda \mid i = j\}$ representa el conjunt $\{\lambda\}$ si $i = j$ i, altrament, el conjunt buit.

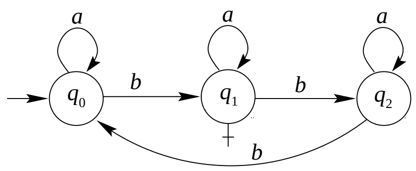

Fig. 6.2 DFA *per al qual construïm una expressió regular*

Com que aquest autòmat té tres estats i només q_1 és acceptador, amb la notació del teorema anterior tenim que $L = L_{0,1,3}$. Ara bé,

$$L_{0,1,3} = L_{0,1,2} \cup L_{0,2,2} \cdot L^*_{2,2,2} \cdot L_{2,1,2}$$

on

$$\begin{aligned} L_{0,1,2} &= L_{0,1,1} \cup L_{0,1,1} \cdot L^*_{1,1,1} \cdot L_{1,1,1} \\ L_{0,2,2} &= L_{0,2,1} \cup L_{0,1,1} \cdot L^*_{1,1,1} \cdot L_{1,2,1} \\ L_{2,2,2} &= L_{2,2,1} \cup L_{2,1,1} \cdot L^*_{1,1,1} \cdot L_{1,2,1} \\ L_{2,1,2} &= L_{2,1,1} \cup L_{2,1,1} \cdot L^*_{1,1,1} \cdot L_{1,1,1} \,. \end{aligned}$$

Utilitzant la regla un altre cop,

$$\begin{aligned} L_{0,1,1} &= L_{0,1,0} \cup L_{0,0,0} \cdot L^*_{0,0,0} \cdot L_{0,1,0} \\ L_{1,1,1} &= L_{1,1,0} \cup L_{1,0,0} \cdot L^*_{0,0,0} \cdot L_{0,1,0} \\ L_{0,2,1} &= L_{0,2,0} \cup L_{0,0,0} \cdot L^*_{0,0,0} \cdot L_{0,2,0} \\ L_{1,2,1} &= L_{1,2,0} \cup L_{1,0,0} \cdot L^*_{0,0,0} \cdot L_{0,2,0} \\ L_{2,2,1} &= L_{2,2,0} \cup L_{2,0,0} \cdot L^*_{0,0,0} \cdot L_{0,2,0} \\ L_{2,1,1} &= L_{2,1,0} \cup L_{2,0,0} \cdot L^*_{0,0,0} \cdot L_{0,1,0} \end{aligned}$$

i com que tots els llenguatges que apareixen a la dreta d'aquestes igualtats són de la forma $L_{i,j,0}$, els podem calcular directament i obtenim

$$\begin{aligned} r_{0,0,0} = r_{1,1,0} = r_{2,2,0} &= a + \Lambda \\ r_{0,1,0} = r_{1,2,0} = r_{2,0,0} &= b \\ r_{1,0,0} = r_{2,1,0} = r_{0,2,0} &= \varnothing. \end{aligned}$$

Substituint en les igualtats anteriors,

$$\begin{aligned} r_{0,1,1} &= b + (a+\Lambda)\,(a+\Lambda)^*\, b = a^* b \\ r_{1,1,1} &= (a+\Lambda)\; + \varnothing\,(a+\Lambda)^*\, b = a + \Lambda \\ r_{0,2,1} &= \varnothing\; + \;(a+\Lambda)\,(a+\Lambda)^*\,\varnothing\; = \varnothing \\ r_{1,2,1} &= b\; + \varnothing\,(a+\Lambda)^*\,\varnothing\; = b \\ r_{2,2,1} &= (a+\Lambda) + b\,(a+\Lambda)^*\,\varnothing\; = a + \Lambda \\ r_{2,1,1} &= \varnothing\; + \; b\,(a+\Lambda)^*\, b = b a^* b \end{aligned}$$

d'on

$$\begin{aligned} r_{0,1,2} &= a^*b + (a^*b)\,(a+\Lambda)^*\,(a+\Lambda) = a^*ba^* \\ r_{0,2,2} &= \varnothing\; + \;(a^*b)\,(a+\Lambda)^*\, b = a^*ba^*b \\ r_{2,2,2} &= (a+\Lambda) + (ba^*b)\,(a+\Lambda)^*\, b = a + \Lambda + ba^*ba^*b \\ r_{2,1,2} &= ba^*b + (ba^*b)\,(a+\Lambda)^*\,(a+\Lambda) = ba^*ba^* \end{aligned}$$

i, finalment, deduïm que L és el llenguatge associat a

$$r = r_{0,1,3} = a^*ba^* + (a^*ba^*b)\,(a + \Lambda + ba^*ba^*b)^*\,(ba^*ba^*).$$

Observeu que, al llarg de tot el procés precedent, hem anat simplificant les expressions que hi apareixien. Convé tenir això en compte perquè el resultat que obtindríem en cas de processar automàticament aquest exemple mitjançant un programa que apliqués simplement les regles de construcció donades podria ser molt més enrevessat.

Per a la segona inclusió del teorema precedent també és fàcil d'imaginar un algorisme per construir un autòmat finit a partir d'una expressió regular: es tracta simplement de començar amb els autòmats elementals que reconeixen els mots formats pels símbols de l'alfabet i combinar-los amb operacions de concatenació, reunió i tancament de Kleene fins arribar a un autòmat que reconegui el llenguatge associat a l'expressió regular de partida.

Observeu que el caràcter constructiu de les dues implicacions del teorema ens permet passar del formalisme dels autòmats finits al de les expressions regulars, i viceversa. Utilitzarem les expressions regulars per descriure llenguatges quan ens interessi posar de manifest l'estructura dels seus mots i el llenguatge concret sigui prou senzill per donar-ne una descripció concisa. Fixeu-vos que les expressions regulars són definicions recursives de llenguatges, de manera similar a com ho són les gramàtiques. De fet, les expressions regulars tenen una traducció directa en termes de gramàtiques. El marc dels autòmats finits, en canvi, és molt més adequat des del punt de vista operatori, ja que és un model de computació efectiu que decideix de manera determinista els mots que pertanyen al llenguatge i els que no. Com a formalisme descriptiu, també pot ser avantatjós quan la definició del llenguatge faci intervenir operacions com ara la complementació o la intersecció, que no existeixen en el marc de les expressions regulars.

Quant al problema de decidir l'equivalència d'expressions regulars, el teorema anterior permet reduir-lo al problema de l'equivalència d'autòmats finits. De fet, aquest és l'únic procediment que tenim per al cas general, ja que en el marc de les expressions regulars no hi ha algorismes generals per poder simplificar expressions qualssevol i comparar-les en una determinada forma canònica única del tipus l'*autòmat mínim* dels autòmats finits. Aquest fet fa que les expressions regulars no siguin un formalisme adequat per a problemes en què calgui fer comparacions entre llenguatges regulars.

6.3 Equacions lineals entre llenguatges. Lema d'Arden

En aquesta secció veurem un resultat, degut a D. N. Arden [Ard60], que permet solucionar equacions lineals de llenguatges i que ofereix un mètode alternatiu per trobar expressions regulars corresponents a autòmats finits. Aquest mètode, tot i requerir un temps de procés del mateix ordre que l'algorisme de McNaughton-Yamada, és preferible quan cal fer els càlculs a mà.

Definició. Una *equació lineal per la dreta* sobre un alfabet Σ és una expressió de la forma

$$X = AX + B$$

on A i B són llenguatges sobre Σ. Una *solució* de l'equació precedent és tot llenguatge L que satisfà $L = A \cdot L \cup B$.

Simètricament, definim les equacions *lineals per l'esquerra* com aquelles que són de la forma $X = XA + B$. Les seves solucions són els llenguatges L que satisfan $L = L \cdot A \cup B$.

Exemple 6.9 Considerem l'equació lineal següent sobre l'alfabet $\Sigma = \{0, 1\}$,

$$X = (0 + 1)X + \Lambda.$$

El llenguatge $L = (0+1)^*$ és solució de la igualtat anterior. En efecte, podem comprovar que: $(0 + 1)(0 + 1)^* + \Lambda = (0 + 1)^+ + \Lambda = (0 + 1)^*$. Però, és la solució proposada l'única solució de l'equació? Com l'hem trobada? El lema següent aportarà els elements necessaris per respondre aquestes qüestions.

Proposició (Lema d'Arden).

1. *El llenguatge A^*B és solució de l'equació $X = AX + B$.*
2. *Qualsevol solució de l'equació $X = AX + B$ conté el llenguatge A^*B.*
3. *Si $\lambda \notin A$, aleshores A^*B n'és l'única solució.*

Demostració.

1. Substituint X per A^*B i utilitzant les propietats d'associativitat i distributivitat de la concatenació de llenguatges, obtenim
$$A(A^*B) + B = A^+B + B = (A^+ + \Lambda)B = A^*B.$$
2. Sigui L una solució qualsevol de $X = AX + B$. Per a tot $k \in \mathbb{N}$ tenim
$$L = AL + B = A^2L + AB + B = \cdots = A^{k+1}L + A^kB + \cdots + A^2B + AB + B$$
i, per tant, $A^kB \subseteq L$, d'on es dedueix inductivament que $A^*B \subseteq L$.
3. Veurem que si suposem l'existència de dues solucions, les dues han de coincidir. Siguin L i M dues solucions de $X = AX + B$.

   3.1 Demostrarem primer que $L \subseteq M$. Com que tots dos llenguatges són solucions, es té
   $$L = AL + B \qquad \text{i} \qquad M = AM + B.$$
   Sigui $w \in L$ i $n = |w|$. Procedim per inducció sobre n.

   - *Base:* $n = 0$. En aquest cas, $w = \lambda \Rightarrow w \notin AL$ (ja que $\lambda \notin A$) $\Rightarrow w \in B \Rightarrow w \in M$.
   - *Pas inductiu:* $n > 0$. Com que $w \in L$, tenim que, o bé $w \in AL$, o bé $w \in B$. En el segon cas, $w \in M$; en el primer, $w = xv$, amb $x \in A$ i $v \in L$. Com que $|x| > 0$, resulta $|v| < n$ i, per la hipòtesi d'inducció, deduïm que $v \in M$. Així, $w \in AM \subseteq M$.

   3.2 La demostració que $M \subseteq L$ es fa de la mateixa manera. Per tant, deduïm que necessàriament $L = M$, i això acaba la demostració. □

Corol·lari. Una demostració idèntica, intercanviant la posició de les constants i les variables, ens permet establir que

1. *El llenguatge BA^* és solució de l'equació $X = XA + B$.*
2. *Qualsevol solució de l'equació $X = XA + B$ conté el llenguatge BA^*.*

3. *Si $\lambda \notin A$, aleshores BA^* n'és l'única solució.*

DEFINICIÓ. Un *sistema d'equacions lineals per la dreta* és un conjunt d'igualtats de la forma

$$\begin{cases} X_1 = A_{11}X_1 + \cdots + A_{1n}X_n + B_1 \\ \quad \vdots \\ X_n = A_{n1}X_1 + \cdots + A_{nn}X_n + B_n \,. \end{cases}$$

on $\forall i, j$ A_{ij} i B_i són llenguatges sobre algun alfabet Σ.

Una *solució* d'aquest sistema és un n-tuple de llenguatges $(L_1, \ldots, L_n)$ que satisfan les equacions. El sistema s'anomena *estricte* si no hi ha cap A_{ij}, per a cap i, j, que contingui λ.

La resolució d'aquests sistemes d'equacions a partir del lema d'Arden es fa seguint un mètode d'aïllament i substitució successiva de variables anàleg al que s'utilitza en àlgebra numèrica. Així, donat un sistema estricte de dues equacions amb dues variables X_1 i X_2, com ara

$$\begin{cases} X_1 = A_{11}X_1 + A_{12}X_2 + B_1 \\ X_2 = A_{21}X_1 + A_{22}X_2 + B_2 \,, \end{cases}$$

fem primer

$$X_1 = A_{11}X_1 + (A_{12}X_2 + B_1)$$

ara, per Arden,

$$= A_{11}^*(A_{12}X_2 + B_1)$$

i ara substituïm X_1 per aquesta expressió a la segona equació. Tenim

$$\begin{aligned} X_2 &= A_{21}A_{11}^*(A_{12}X_2 + B_1) + A_{22}X_2 + B_2 \\ &= (A_{21}A_{11}^*A_{12} + A_{22})X_2 + A_{21}A_{11}^*B_1 + B_2 \end{aligned}$$

i, novament per Arden,

$$= (A_{21}A_{11}^*A_{12} + A_{22})^*(A_{21}A_{11}^*B_1 + B_2) \,.$$

Això ens dóna *la* solució per a X_2. La corresponent a X_1 s'obté substituint X_2 per la seva solució a l'expressió anterior de X_1, cosa que dóna

$$X_1 = A_{11}^*(A_{12}(A_{21}A_{11}^*A_{12} + A_{22})^*(A_{21}A_{11}^*B_1 + B_2) + B_1) \,.$$

Si bé l'extensió d'aquest mètode a un nombre qualsevol d'equacions no presenta cap dificultat, la seva justificació quan passem d'una sola equació a dues equacions o més requereix demostrar que es manté l'existència i la unicitat de la solució. Una demostració formal del manteniment d'aquestes condicions en el cas de sistemes estrictes es troba a la referència [Sal69].

6.4 Sistemes d'equacions lineals associats a un NFA

Veurem ara un mètode alternatiu a l'algorisme de McNaughton i Yamada per construir una expressió regular que tingui com a llenguatge associat el llenguatge reconegut per un autòmat finit. La idea bàsica parteix d'associar a cada estat de l'autòmat finit un llenguatge determinat, el qual es podrà expressar en termes dels llenguatges associats a estats successors o predecessors d'aquest. Aquestes relacions seran equacions lineals de llenguatges, les quals formaran un sistema estricte d'equacions. La resolució d'aquest sistema ens portarà a l'expressió regular corresponent al llenguatge reconegut per l'autòmat. El mètode té dues versions, segons si considerem la linealitat de les equacions per la dreta o per l'esquerra. Els dos enfocaments, que descriurem a continuació, són equivalents i completament simètrics.

Sigui $M = \langle Q, \Sigma, \delta, I, F\rangle$ un NFA qualsevol.

Equacions per l'esquerra

Associem a cada estat, $q \in Q$, el conjunt de mots que van des d'algun estat inicial fins a aquest estat,

$$\mathsf{L}(q) = \{w \in \Sigma^* \mid \exists p \in I \ q \in p\cdot w\},$$

llenguatge que és reconegut pel NFA

$$\langle Q, \Sigma, \delta, I, \{q\}\rangle.$$

Podem descompondre aquest llenguatge en funció dels que corresponen als estats predecessors de q,

$$\mathsf{L}(q) = \bigcup_{p,a \mid q \in p\cdot a} \mathsf{L}(p)\cdot a \ \cup \ \{\lambda \mid q \in I\}.$$

Amb la notació precedent el llenguatge reconegut per l'autòmat és:

$$\mathsf{L}(M) = \bigcup_{q \in F} \mathsf{L}(q).$$

Equacions per la dreta

Associem a cada estat, $p \in Q$, el conjunt de mots que van des d'aquest estat fins a algun estat acceptador,

$$\mathsf{L}(p) = \{w \in \Sigma^* \mid \exists q \in F \ q \in p\cdot w\},$$

llenguatge que és reconegut pel NFA

$$\langle Q, \Sigma, \delta, \{p\}, F\rangle.$$

Podem descompondre aquest llenguatge en funció dels que corresponen als estats successors de p,

$$\mathsf{L}(p) = \bigcup_{q,a \mid q \in p\cdot a} a\cdot \mathsf{L}(q) \ \cup \ \{\lambda \mid p \in F\}.$$

Amb la notació precedent el llenguatge reconegut per l'autòmat és:

$$\mathsf{L}(M) = \bigcup_{p \in I} \mathsf{L}(p).$$

No és necessari partir d'autòmats deterministes per poder formular aquests sistemes d'equacions. És més, en general, evitar el pas al determinisme permetrà treballar amb sistemes més curts i arribar, per tant, a donar expressions regulars més curtes i senzilles com a solució del llenguatge reconegut per l'autòmat finit. Fixeu-vos que, parlant informalment, per construir l'equació associada a cada estat, en el cas de linealitat per la dreta cal tenir en compte "les fletxes que surten" d'aquest estat, mentre que en el cas simètric cal fixar-se en "les fletxes que hi arriben".

Observeu que es tracta, en els dos casos, de sistemes estrictes, ja que els coeficients de les variables en les diferents equacions estan formats per un o més mots d'un sol símbol, que són els que defineixen la transició d'un estat cap a un altre en l'autòmat, sense que això es vegi afectat pel caràcter eventualment indeterminista de l'autòmat.

Si bé la complexitat de resoldre els dos sistemes considerats és equivalent en el cas pitjor, a la pràctica els dos sistemes tenen lleugeres diferències que depenen, en cada cas concret, del graf de l'autòmat tractat, i que poden fer preferible un sistema o un altre.

Veurem, en un exemple, l'aplicació dels resultats precedents.

Exemple 6.10 Considereu el DFA, M, de la figura següent.

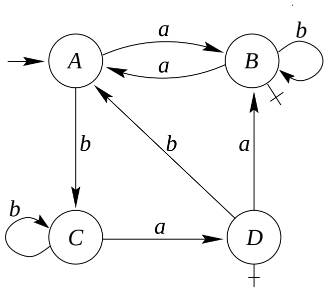

Fig. 6.3 *Exemple de* DFA *per al càlcul d'expressions regulars*

Els sistemes d'equacions lineals a esquerra i dreta que s'obtenen del DFA anterior, utilitzant, per raons de simplicitat, els mateixos noms dels estats com a noms dels llenguatges associats a ells, són els següents:

	Esquerra		*Dreta*
(1)	$A = Ba + Db + \Lambda$	(1)	$A = aB + bC$
(2)	$B = Aa + Bb + Da$	(2)	$B = aA + bB + \Lambda$
(3)	$C = Cb + Ab$	(3)	$C = aD + bC$
(4)	$D = Ca$	(4)	$D = aB + bA + \Lambda\,.$

Si partim del primer sistema tenim $\mathsf{L}(M) = B + D$, mentre que per al segon tenim $\mathsf{L}(M) = A$. Resoldrem els dos sistemes per separat i comprovarem aquesta igualtat.

– *Esquerra*

Resolent per Arden les equacions 2 i 3, obtenim $B = (Aa + Da)b^*$ i $C = Abb^*$, respectivament. Substituir la nova expressió de C a l'equació 4 ens porta a $D = Abb^*a$, i si substituïm aquesta darrera a l'expressió precedent de B obtenim $B = (Aa + Abb^*aa)b^* = A(ab^* + bb^*aab^*)$. A continuació substituïm B i D a l'equació 1 i obtenim

$$A = A(ab^* + bb^*aab^*)a + Abb^*ab + \Lambda = A(ab^*a + bb^*aab^*a + bb^*ab) + \Lambda\,.$$

Resolent per Arden, obtenim l'expressió regular del llenguatge A

$$A = (ab^*a + bb^*aab^*a + bb^*ab)^*\,.$$

Així, finalment,

$$\begin{aligned}\mathsf{L}(M) &= B + D = A(ab^* + bb^*aab^*) + Abb^*a = A(ab^* + bb^*aab^* + bb^*a)\\ &= (ab^*a + bb^*aab^*a + bb^*ab)^*(ab^* + bb^*aab^* + bb^*a)\,.\end{aligned}$$

– *Dreta*

Resolent per Arden les equacions 2 i 3 obtenim

$$B = b^*(aA + \Lambda) = b^*aA + b^*$$

i

$$C = b^*aD,$$

respectivament. Substituint la nova expressió de B a l'equació 4 obtenim

$$D = ab^*(aA + \Lambda) + bA + \Lambda = (ab^*a + b)A + (ab^* + \Lambda)\,.$$

Substituint aquesta expressió de D a la de C anterior obtenim

$$C = (b^*aab^*a + b^*ab)A + (b^*aab^* + b^*a)\,,$$

i substituint B i C a l'equació 1 obtenim l'equació lineal següent:

$$\begin{aligned} A &= ab^*aA + ab^* + (bb^*aab^*a + bb^*ab)A + (bb^*aab^* + bb^*a) \\ &= (ab^*a + bb^*aab^*a + bb^*ab)A + (ab^* + bb^*aab^* + bb^*a)\,. \end{aligned}$$

Finalment, per Arden, obtenim l'expressió regular buscada

$$\mathsf{L}(M) = A = (ab^*a + bb^*aab^*a + bb^*ab)^*(ab^* + bb^*aab^* + bb^*a)\,.$$

Observeu la relació entre l'expressió regular que descriu els mots del llenguatge reconegut per l'autòmat M i els camins sobre aquest autòmat que porten de l'estat inicial als estats acceptadors. L'expressió $(ab^*a + bb^*aab^*a + bb^*ab)^*$ descriu el llenguatge dels mots que buclegen sobre l'estat A de l'autòmat; l'expressió $ab^* + bb^*aab^*$ descriu el conjunt de mots que porten de l'estat A a l'estat B sense tornar a passar per A, i l'expressió bb^*a denota el conjunt de mots que porten de l'estat A a l'estat D sense tornar a visitar A. Per tant, l'expressió regular completa descriu mots w que es poden escriure com a concatenació de dos mots x i y, el primer dels quals comença i acaba el càlcul sobre l'autòmat a l'estat A i el segon ens porta d'aquest estat a algun dels dos estats acceptadors, B o D.

Tot i que a l'exemple anterior els dos sistemes d'equacions han donat com a solució la mateixa expressió regular, és freqüent que els resultats obtinguts siguin diferents. Fins i tot partint d'un mateix sistema, el resultat pot dependre del camí seguit en el procés de substitucions. Ho podem veure a l'exemple que segueix.

Exemple 6.11 Considerem de nou el llenguatge definit a l'exemple 6.8. Per tal de facilitar l'escriptura del sistema d'equacions, representarem per les lletres A, B i C els llenguatges associats als estats q_0, q_1 i q_2, respectivament. El sistema d'equacions per l'esquerra associat a l'autòmat és el següent:

$$\begin{aligned} A &= Aa + Cb + \Lambda \\ B &= Ab + Ba \\ C &= Bb + Ca\,, \end{aligned}$$

i el llenguatge reconegut per l'autòmat és l'associat a l'estat acceptador, que correspon a la variable B.

Una forma de resoldre aquest sistema consisteix a aplicar Arden a cada una de les tres equacions. N'obtenim els resultats següents:

$$\begin{aligned} A &= (\Lambda + Cb)a^* = a^* + Cba^* \\ B &= Aba^* \\ C &= Bba^*\,. \end{aligned}$$

Si ara substituïm successivament aquest valor de C a l'expressió de A, i el valor resultant de A a l'expressió de B, tenim el següent:

$$\begin{aligned} A &= a^* + Bba^*ba^* \\ B &= a^*ba^* + Bba^*ba^*ba^*\,, \end{aligned}$$

i novament per aplicació d'Arden obtenim el resultat

$$\mathsf{L}(M) = B = a^*ba^*(ba^*ba^*ba^*)^* \,.$$

Una alternativa a la resolució d'aquest mateix sistema d'equacions consisteix a començar aplicant Arden només a les equacions de B i C, cosa que dóna els valors ja obtinguts. Però ara podem substituir successivament aquest valor de B a l'expressió de C i el resultat de C a l'expressió de A. Cal seguir els passos següents:

$$\begin{aligned} &B = Aba^* \\ &C = Bba^* = Aba^*ba^* \\ &A = \Lambda + Aa + Cb = \Lambda + A(a + ba^*ba^*b) \,. \end{aligned}$$

Ara podem trobar, per Arden, que el valor de A és $(a + ba^*ba^*b)^*$, el qual, substituït a l'expressió de B, dóna finalment

$$\mathsf{L}(M) = B = (a + ba^*ba^*b)^*ba^* \,.$$

Podem constatar que es tracta d'un resultat diferent a l'obtingut anteriorment, tot i correspondre al mateix llenguatge. Observeu també que tant un resultat com l'altre són diferents a l'obtingut a l'exemple 6.8.

6.5 Gramàtiques regulars

En aquesta secció i en la següent veurem que és possible caracteritzar els llenguatges regulars mitjançant un tipus restringit de gramàtiques incontextuals. Aquest resultat ens donarà molta flexibilitat a l'hora de treballar amb els llenguatges regulars, ja que en cada cas podrem optar per un autòmat o per una gramàtica com a descriptor del llenguatge, depenent de quin dels dos ens doni més facilitat. Més endavant, al capítol 8, farem una caracterització recíproca dels llenguatges incontextuals, és a dir, definirem un tipus més potent d'autòmats que reconeixen exactament aquests llenguatges.

Com es posarà de manifest de seguida, hi ha un paral·lel formal entre les gramàtiques regulars i els sistemes d'equacions lineals associats a autòmats que acabem d'estudiar. També aquí convé desenvolupar, a la vegada, dos models simètrics de gramàtiques.

Definició. Una gramàtica

$$G = \langle V, \Sigma, P, S \rangle$$

és *lineal per l'esquerra* quan totes les seves produccions són d'una de les dues formes següents:

$$\left.\begin{aligned} X &\to Yw \\ X &\to w \end{aligned}\right\} \quad \text{on } X, Y \in V \text{ i } w \in \Sigma^*.$$

Definició. Una gramàtica

$$G = \langle V, \Sigma, P, S \rangle$$

és *lineal per la dreta* quan totes les seves produccions són d'una de les dues formes següents:

$$\left.\begin{aligned} X &\to wY \\ X &\to w \end{aligned}\right\} \quad \text{on } X, Y \in V \text{ i } w \in \Sigma^*.$$

FORMA NORMAL. *Per a tota gramàtica lineal per l'esquerra, que no contingui produccions unàries,*[3] *existeix una gramàtica equivalent en què totes les produccions són d'una de les dues formes següents:*

$$\left.\begin{array}{l} X \to Ya \\ X \to \lambda \end{array}\right\} \quad \textit{on } X,Y \in V \textit{ i } a \in \Sigma.$$

FORMA NORMAL. *Per a tota gramàtica lineal per la dreta, que no contingui produccions unàries,*[3] *existeix una gramàtica equivalent en què totes les produccions són d'una de les dues formes següents:*

$$\left.\begin{array}{l} X \to aY \\ X \to \lambda \end{array}\right\} \quad \textit{on } X,Y \in V \textit{ i } a \in \Sigma.$$

DEMOSTRACIÓ. Farem la demostració per al cas de les gramàtiques lineals per la dreta. L'altre cas admet una construcció simètrica.

Considerem una producció de la forma $X \to wY$, en què $|w| > 1$. Sigui $w = a_1 \dots a_n$. Substituïm cada producció d'aquest tipus pel conjunt de n produccions següent:

$$\begin{aligned} X &\to a_1 Z_1 \\ Z_1 &\to a_2 Z_2 \\ &\vdots \\ Z_{n-2} &\to a_{n-1} Z_{n-1} \\ Z_{n-1} &\to a_n Y\,, \end{aligned}$$

cosa que comporta afegir les $n-1$ noves variables $Z_1, \dots, Z_{n-1}$ al conjunt V.

Anàlogament, cada producció de la forma $X \to a_1 \dots a_n$, amb $n \geqslant 1$, és substituïda pel conjunt de produccions

$$\begin{aligned} X &\to a_1 Z_1 \\ Z_1 &\to a_2 Z_2 \\ &\vdots \\ Z_{n-1} &\to a_n Z_n \\ Z_n &\to \lambda\,, \end{aligned}$$

i ara les variables que cal afegir són $Z_1, \dots, Z_n$. Remarquem que les noves variables introduïdes per cada producció eliminada han de ser diferents de les introduïdes per les altres eliminacions.

Sobre la base d'aquestes substitucions, és fàcil constatar que la gramàtica resultant és equivalent a la primitiva. □

DEFINICIÓ. Anomenem *gramàtica regular* tota gramàtica que sigui lineal per l'esquerra o lineal per la dreta.

La raó de donar-los aquest nom és que, com veurem a la secció següent, la família de llenguatges generats per les gramàtiques regulars és precisament la dels llenguatges regulars. Cal advertir, tanmateix, que perquè una gramàtica sigui regular ha de ser *enterament* lineal per l'esquerra o enterament lineal per la dreta. Una gramàtica que

[3] La possibilitat d'eliminar sempre les produccions unàries es demostra seguint un camí idèntic a l'utilitzat al capítol 3, sense que en aquest cas calgui procedir a eliminar prèviament les produccions nul·les.

contingui els dos tipus de produccions no és una gramàtica regular i pot, doncs, generar un llenguatge no regular. El següent n'és un exemple.

Exemple 6.12 La gramàtica

$$\begin{aligned} S &\to aX \mid \lambda \\ X &\to Sb \end{aligned}$$

té una producció lineal per la dreta i una altra de lineal per l'esquerra. El llenguatge generat és, evidentment,

$$\{a^n b^n \mid n \geqslant 0\}$$

que no és regular, com ja vam veure a l'última secció del capítol 4.

6.6 Correspondència entre gramàtiques regulars i autòmats finits

En aquesta secció veurem la forma de construir, a partir d'una gramàtica regular (tant si és lineal per l'esquerra com si ho és per la dreta), un autòmat finit que reconeix el llenguatge generat per la gramàtica. També farem la construcció inversa, és a dir, la de les gramàtiques lineals per l'esquerra i per la dreta que corresponen a un autòmat donat. La idea subjacent a totes aquestes transformacions és la d'una doble identificació. D'una banda, identifiquem els estats de l'autòmat amb les variables de la gramàtica corresponent. D'altra banda, identifiquem les transicions de l'autòmat amb les produccions de la gramàtica. Per tal d'aconseguir que aquestes construccions siguin simètriques (a dreta i esquerra) i reversibles (de gramàtica a autòmat i d'autòmat a gramàtica), és interessant començar generalitzant la definició de gramàtica, de manera que una gramàtica pugui tenir més d'una variable inicial.

Al llarg d'aquesta secció, considerarem que una gramàtica és una estructura de la forma $G = \langle V, \Sigma, P, H\rangle$, on el caràcter dels tres primers components —variables, alfabet i produccions— és el mateix de sempre, però l'últim component ja no és un símbol de V —la variable inicial— sinó que ara és un *subconjunt* de símbols de V, que anomenarem conjunt de variables inicials. El llenguatge generat per la gramàtica es defineix ara de la manera següent:

$$\mathsf{L}(G) = \{w \in \Sigma \mid \exists X \in H\ X \overset{*}{\Rightarrow} w\}.$$

Aquesta generalització de la definició de gramàtica no amplia la família dels llenguatges generats, que continua essent la dels CFL. En efecte, per cada gramàtica $G = \langle V, \Sigma, P, H\rangle$ del nou tipus podem construir-ne una del tipus antic $G' = \langle V', \Sigma, P', S\rangle$, on $S \notin V$ és l'única variable nova, $V' = V \cup \{S\}$, i on P' s'obté afegint a P les noves produccions $\{S \to X \mid X \in H\}$. És immediat verificar que $\mathsf{L}(G') = \mathsf{L}(G)$.

Remarquem que aquesta modificació és compatible amb la definició de gramàtiques regulars, que continuen essent les que tenen totes les produccions lineals per la dreta o per l'esquerra.

Transformació de gramàtica a autòmat

Sigui $G_{\mathrm{e}} = \langle V, \Sigma, P_{\mathrm{e}}, F\rangle$ una gramàtica lineal per l'esquerra, on les produccions de P_{e} estan en la forma normal definida a la secció precedent, i on F representa el subconjunt de variables inicials.
Definim el NFA $M = \langle V, \Sigma, \delta, I, F\rangle$, on

$$I = \{X \in V \mid X \to \lambda \in P_{\mathrm{e}}\}$$

i on $\forall X \in V\ \forall a \in \Sigma$

$$\delta(X, a) = \{Y \mid Y \to Xa \in P_{\mathrm{e}}\}.$$

Sigui $G_{\mathrm{d}} = \langle V, \Sigma, P_{\mathrm{d}}, I\rangle$ una gramàtica lineal per la dreta, on les produccions de P_{d} estan en la forma normal definida a la secció precedent, i on I representa el subconjunt de variables inicials.
Definim el NFA $M = \langle V, \Sigma, \delta, I, F\rangle$, on

$$F = \{X \in V \mid X \to \lambda \in P_{\mathrm{d}}\}$$

i on $\forall X \in V\ \forall a \in \Sigma$

$$\delta(X, a) = \{Y \mid X \to aY \in P_{\mathrm{d}}\}.$$

Transformació d'autòmat a gramàtica

Sigui $M = \langle V, \Sigma, \delta, I, F\rangle$ un NFA qualsevol.

Definim $G_{\mathrm{e}} = \langle V, \Sigma, P_{\mathrm{e}}, F\rangle$, on

$$P_{\mathrm{e}} = \{Y \to Xa \mid Y \in \delta(X, a)\} \cup \{X \to \lambda \mid X \in I\}.$$

Definim $G_{\mathrm{d}} = \langle V, \Sigma, P_{\mathrm{d}}, I\rangle$, on

$$P_{\mathrm{d}} = \{X \to aY \mid Y \in \delta(X, a)\} \cup \{X \to \lambda \mid X \in F\}.$$

Pas de gramàtica esquerra a gramàtica dreta, i viceversa

Sigui $G_{\mathrm{e}} = \langle V, \Sigma, P_{\mathrm{e}}, F\rangle$ una gramàtica lineal per l'esquerra, on les produccions de P_{e} estan en la forma normal definida a la secció precedent, i on F representa el subconjunt de variables inicials.
Definim $G_{\mathrm{d}} = \langle V, \Sigma, P_{\mathrm{d}}, I\rangle$, on

$$P_{\mathrm{d}} = \{X \to aY \mid Y \to Xa \in P_{\mathrm{e}}\} \cup \{X \to \lambda \mid X \in F\}$$

i on

$$I = \{X \in V \mid X \to \lambda \in P_{\mathrm{e}}\}.$$

Sigui $G_{\mathrm{d}} = \langle V, \Sigma, P_{\mathrm{d}}, I\rangle$ una gramàtica lineal per la dreta, on les produccions de P_{d} estan en la forma normal definida a la secció precedent, i on I representa el subconjunt de variables inicials.
Definim $G_{\mathrm{e}} = \langle V, \Sigma, P_{\mathrm{e}}, F\rangle$, on

$$P_{\mathrm{e}} = \{Y \to Xa \mid X \to aY \in P_{\mathrm{d}}\} \cup \{X \to \lambda \mid X \in I\}$$

i on

$$F = \{X \in V \mid X \to \lambda \in P_{\mathrm{d}}\}.$$

Demostració conjunta de les tres transformacions

LEMA 1. $\forall w \in \Sigma^* \ \forall X, Y \in V$ *els tres enunciats següents són equivalents:*

(1) $Y \in \delta(X, w)$ (2) $Y \overset{*}{\underset{G_{\mathrm{e}}}{\Rightarrow}} Xw$ (3) $X \overset{*}{\underset{G_{\mathrm{d}}}{\Rightarrow}} wY$.

DEMOSTRACIÓ. Immediata, per inducció sobre $|w|$. □

Lema 2.(e)

$$Y \overset{*}{\underset{G_e}{\Rightarrow}} w \iff \exists X \in I \; Y \in \delta(X, w).$$

Demostració. Resulta del lema anterior i del fet que

$$X \in I \iff (X \to \lambda) \in P_e.$$

□

Lema 2.(d)

$$X \overset{*}{\underset{G_d}{\Rightarrow}} w \iff \exists Y \in F \; Y \in \delta(X, w).$$

Demostració. Resulta del lema anterior i del fet que

$$Y \in F \iff (Y \to \lambda) \in P_d.$$

□

Proposició. $\mathsf{L}(M) = \mathsf{L}(G_e)$

Demostració. $w \in \mathsf{L}(M) \iff$

$$\iff \exists X \in I \; \exists Y \in F \quad Y \in \delta(X, w)$$
$$\iff \exists Y \in F \quad Y \overset{*}{\underset{G_e}{\Rightarrow}} w$$
$$\iff w \in \mathsf{L}(G_e).$$

□

Proposició. $\mathsf{L}(M) = \mathsf{L}(G_d)$

Demostració. $w \in \mathsf{L}(M) \iff$

$$\iff \exists X \in I \; \exists Y \in F \quad Y \in \delta(X, w)$$
$$\iff \exists X \in I \quad X \overset{*}{\underset{G_d}{\Rightarrow}} w$$
$$\iff w \in \mathsf{L}(G_d).$$

□

Aquest parell de proposicions simètriques que acabem de demostrar ens permeten formular en forma de teorema la propietat fonamental amb què hem introduït el tema de les gramàtiques regulars.

Teorema. *La família de llenguatges generats per les gramàtiques regulars és la família dels llenguatges regulars.*

Com que les gramàtiques regulars són un cas particular de les gramàtiques incontextuals, podem enunciar el corol·lari següent del teorema precedent.

Teorema. *La família dels llenguatges regulars està inclosa en la família dels llenguatges incontextuals.*

De nou el llenguatge $\{a^n b^n \mid n \geqslant 0\}$, exposat a l'exemple 4.25, serveix per posar de manifest que aquesta inclusió dels regulars en els incontextuals és *estricta*.

Exemple 6.13 Considerem novament l'autòmat definit a l'exemple 6.10. La construcció de la gramàtica per l'esquerra es fa, com hem vist, associant a cada variable una producció per cada fletxa *que arriba* a l'estat corresponent. En canvi, les produccions per la dreta corresponen a les fletxes *que surten* de l'estat corresponent. És a dir, el *modus operandi* és idèntic a l'utilitzat en la construcció dels sistemes d'equacions lineals associats a l'autòmat. Les gramàtiques regulars són, en aquest cas, les següents:

Esquerra	*Dreta*
$A \to Ba \mid Db \mid \lambda$	$A \to aB \mid bC$
$B \to Aa \mid Bb \mid Da$	$B \to aA \mid bB \mid \lambda$
$C \to Cb \mid Ab$	$C \to aD \mid bC$
$D \to Ca$	$D \to aB \mid bA \mid \lambda\,.$

En el cas de la gramàtica de l'esquerra, les variables inicials són B i D, mentre que en el cas de la gramàtica de la dreta la variable inicial és A. Remarquem, doncs, la identitat formal que resulta entre sistemes lineals d'equacions i gramàtiques regulars, on els únics canvis són el del signe igual per la fletxa i el del signe additiu per la barra vertical.

Si es volgués passar la gramàtica de l'esquerra a la forma usual amb una única variable inicial, n'hi hauria prou d'afegir el nou símbol inicial S i les produccions $S \to B$ i $S \to D$. En cas de voler mantenir la forma normal sense produccions unàries, substituiríem aquestes noves produccions per

$$S \to Aa \mid Bb \mid Da \mid Ca\,.$$

Amb vista a facilitar el record de les transformacions anteriors, pot ser útil limitar-se a retenir les regles que regeixen la conversió d'autòmat a gramàtica dreta, i viceversa. Es tracta de la conversió més natural, ja que tant l'autòmat com la gramàtica dreta processen els mots d'esquerra a dreta, mentre que una gramàtica esquerra genera els mots de dreta a esquerra. Aleshores, per construir una gramàtica esquerra a partir d'un autòmat podem seguir el camí indirecte següent. Primer, revessar l'autòmat, és a dir, construir a partir de l'autòmat donat l'autòmat que reconeix el llenguatge revessat (aquesta construcció es troba a la secció 7 del capítol 4). Segon, construir, seguint el procediment que acabem d'exposar, la gramàtica dreta que correspon a aquest autòmat revessat. Tercer, revessar aquesta gramàtica, és a dir, construir a partir d'ella la gramàtica que genera el llenguatge revessat (aquesta construcció es troba a la secció 5 del capítol 2). El resultat final és una gramàtica esquerra que genera el llenguatge original. Aquesta gramàtica serà exactament la que hauríem obtingut aplicant directament la transformació d'autòmat a gramàtica esquerra. Quant a la construcció recíproca, de gramàtica esquerra a autòmat, cal seguir els tres passos anteriors en l'ordre invers, amb l'únic canvi que ara el pas segon consistirà a construir un autòmat a partir d'una gramàtica dreta, i no a l'inrevés com abans.

Inambigüitat dels llenguatges regulars

Quan la construcció d'una gramàtica regular, seguint les regles donades en aquesta mateixa secció, es fa a partir d'un autòmat finit que és determinista, la gramàtica obtinguda, tant si és lineal per la dreta com per l'esquerra, resulta no ambigua. Aquest fet és una conseqüència immediata de la correspondència biunívoca que hem establert entre les transicions de l'autòmat i les produccions de la gramàtica. En efecte, d'això resulta una altra correspondència, també biunívoca, entre camins recorreguts en processar un mot en el graf de l'autòmat i arbres de derivació d'aquest mateix mot en les gramàtiques considerades. L'esquema d'aquesta correspondència ha estat dibuixat a la figura 6.4.

Així doncs, la unicitat de l'arbre de derivació d'un mot qualsevol, tant en gramàtiques dretes com en gramàtiques esquerres, és un resultat de la unicitat del camí d'acceptació d'aquest mot quan l'autòmat donat és determinista. Podem, doncs, enunciar el teorema següent.

TEOREMA. *Tots els llenguatges regulars són inambigus.*

No hem de confondre la unicitat de l'arbre de derivació amb la possibilitat d'identificar pas a pas quina és la producció que cal elegir per generar un mot donat. Aquesta possibilitat, que existeix en les gramàtiques lineals per la dreta que provenen d'autòmats deterministes, no existeix en general en les gramàtiques per l'esquerra.

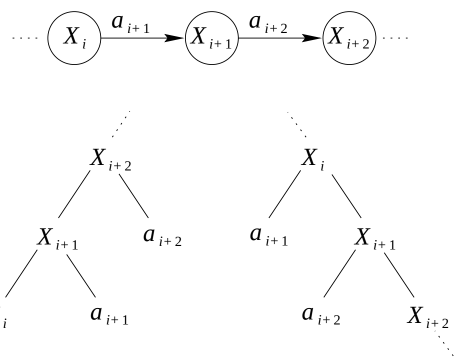

Fig. 6.4 *Correspondència entre camins i arbres de derivació*

6.7 Morfismes i substitucions de llenguatges regulars

Hem diferit fins a aquest capítol la demostració que els llenguatges regulars són tancats respecte dels morfismes i les substitucions, perquè aquestes són propietats lligades a la *racionalitat* dels llenguatges regulars i no a la seva *recognoscibilitat*. És a dir que requereixen disposar d'especificacions *generatives* dels llenguatges, com ho són tant les gramàtiques com les expressions regulars.

Morfisme d'un llenguatge regular

La construcció de la gramàtica que genera la imatge per un morfisme d'un llenguatge regular es fa aplicant la regla general donada al capítol 2 per a gramàtiques qualssevol. És clar que el procediment allí exposat conserva la regularitat de la gramàtica. I aquest fet es fa encara més manifest si partim d'una gramàtica en la forma normal definida en aquest mateix capítol.

Sigui L_1 un llenguatge regular qualsevol i sigui $G_1 = \langle V, \Sigma_1, P_1, S\rangle$ una gramàtica lineal per la dreta en forma normal que el generi. Sigui $h\colon \Sigma_1^* \longrightarrow \Sigma_2^*$ un morfisme. La gramàtica $G_2 = \langle V, \Sigma_2, P_2, S\rangle$ que genera $h(L_1)$, construïda seguint la regla general per a CFGs, manté en P_2 totes les produccions de la forma $X \to \lambda$ que hi havia a P_1 i converteix cada producció de P_1 de la forma $X \to aY$ en la producció $X \to h(a)Y$ de P_2. Així, les produccions de P_2 resulten totes lineals per la dreta (encara que no quedin en forma normal).

Com que es tracta d'un cas particular de la construcció general, no cal tornar a demostrar que $\mathsf{L}(G_2) = h(L_1)$. I com que la gramàtica G_2 obtinguda és regular, en resulta el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de l'aplicació de morfismes directes.*

Exemple 6.14 Una de les aplicacions principals de la conservació de la regularitat quan s'aplica un morfisme és facilitar la demostració de la no-regularitat de determinats llenguatges. Al capítol 7 desenvoluparem aquesta idea més extensament. Un exemple senzill d'aquesta aplicació és la utilització dels morfismes per *esborrar* símbols dels mots d'un llenguatge i transformar-lo en un altre de més simple. Així, l'aplicació del morfisme $h(a) = a$, $h(b) = b$, $h(c) = h(d) = \lambda$ transforma el llenguatge $L_1 = \{c(ad)^n db^n cc \mid n \geqslant 0\}$ en el llenguatge $L_2 = \{a^n b^n \mid n \geqslant 0\}$. El llenguatge L_1 no pot ser regular, ja que si ho fos també ho seria L_2. I és més fàcil demostrar directament la no-regularitat de L_2 que la de L_1.

Substitució regular d'un llenguatge regular

L'existència d'expressions regulars associades a qualsevol llenguatge regular ens permet fer una construcció de les substitucions basada en aquest fet. Però, a diferència del que acabem de fer en el cas dels morfismes, no podem aprofitar, per als llenguatges regulars, la construcció general de les substitucions feta per a CFL. La raó d'això és que encara que partíssim de gramàtiques regulars per a tots els llenguatges implicats en la substitució, el procés que se segueix no preservaria aquesta regularitat.

Sigui L' un llenguatge regular definit sobre un determinat alfabet Σ' i sigui

$$\sigma \colon \Sigma'^* \longrightarrow \mathsf{Reg}$$

una substitució on la imatge de cada mot és un llenguatge regular. Direm que σ és una *substitució regular*. Considerem que aquesta substitució ve definida per les imatges dels símbols de Σ'. Sigui $\Sigma' = \{a_1, \ldots, a_n\}$, i sigui $L_i = \sigma(a_i)$ per a cada i entre 1 i n. Considerem que a cada L_i li podem associar una expressió regular r_i. Sigui r' l'expressió regular associada al llenguatge L'.

Reemplacem cada ocurrència d'un símbol a_i a r' per l'expressió regular r_i. Sigui r l'expressió regular que resulta de fer aquests reemplaçaments.

PROPOSICIÓ. $\mathsf{L}(r) = \sigma(\mathsf{L}(r'))$.

DEMOSTRACIÓ. La demostració es basa en les propietats de les substitucions estudiades al capítol 1. Recordem que la substitució d'una reunió, concatenació o estrella de llenguatges és la reunió, concatenació o estrella de la substitució d'aquests llenguatges. Una senzilla inducció sobre el nombre d'operadors de r' completa la prova. □

Podem concloure finalment amb el teorema següent.

TEOREMA. *La família dels llenguatges regulars és tancada respecte de les substitucions regulars.*

Un raonament idèntic al fet per als llenguatges incontextuals al final del capítol 2 permet afirmar que, també en el cas dels llenguatges regulars, les propietats de tancament respecte de les operacions de reunió, concatenació i estrella s'haurien pogut deduir del tancament respecte de les substitucions. Només cal tenir en compte que els conjunts $\{a_1, a_2\}$, $\{a_1 a_2\}$ i a_1^* que serveixen de base a les substitucions corresponents a la reunió, concatenació i estrella, són llenguatges regulars.

Exercicis

6.1 Trobeu directament, sense construir-ne prèviament els autòmats finits, expressions regulars dels llenguatges definits als apartats 1 a 4 de l'exercici 4.1.

6.2 Trobeu expressions regulars a partir dels autòmats finits que corresponen als llenguatges dels apartats 5 a 9 de l'exercici 4.1.

6.3 Trobeu gramàtiques regulars per a tots els llenguatges definits a l'exercici 4.1.

6.4 Doneu una gramàtica incontextual inambigua que generi el conjunt d'expressions regulars ben formades sobre l'alfabet $\{\cdot, +, *, (,), \Lambda, \varnothing, a, b\}$.

6.5 Construïu DFA que reconeguin els llenguatges associats a les expressions regulars següents:
1. $((ab^*ab)^*a^*b)^*$
2. $((a^*b)^*a^*c)^*b^*(ab^*)^*$
3. $b^*a(bb+bab^*a)^*ba(bb)^*$
4. $\Lambda+(a+bb+ba(aa+b)^*ab)^*a$
5. $(a+b(ab^*a)^*b)^*$
6. $(a+b(bb+bab+ab^*a)^*baa)^*$

6.6 Demostreu que l'expressió regular següent, definida sobre l'alfabet $\{0,1\}$, descriu el llenguatge format per λ i pels mots que, interpretats com a nombres binaris, tenen un valor múltiple de cinc:

$$(0+1(10+(0+11)(01^*01)^*01^*00)^*(0+11)(01^*01)^*1)^*.$$

6.7 Demostreu les equivalències següents entre expressions regulars:
1. $a^*(b+ca^*)^* \;=\; (a+b^*c)^*b^*$.
2. $(bb+ba+a)^*baa^* \;=\; a^*b(aa^*b+ba^*b)^*aa^*$.
3. $(\Lambda+(a+b)^*baa)a^* = \Lambda+a+(a+b)^*aa$.
4. $(\Lambda+b)a^*(b+bba^*)^* = b^*(a+bb+bbb)^*b^*$.

6.8 Considereu l'expressió regular següent, sobre l'alfabet $\Sigma=\{a,b,c\}$:

$$r = c^*(\Lambda+a(a+b+c)^*+(a+b+c)^*b)c^*.$$

Trobeu un contraexemple que provi que $\mathsf{L}(r)\neq\Sigma^*$. Demostreu que $\mathsf{L}(r^2)=\Sigma^*$.

6.9 Doneu una gramàtica regular per a l'únic llenguatge sobre l'alfabet $\{a\,,b\}$ que satisfà l'equació $L=\overline{L}a$.

6.10 Trobeu una condició necessària i suficient que ha de complir qualsevol llenguatge L que satisfaci l'equació $(La+b)^*=(bL+a)^*$.

6.11 Demostreu l'equivalència següent, en la qual r i s designen expressions regulars:

$$(\forall s\ (rs)^*r=(r+s)^*) \iff r=r^*.$$

6.12 Considereu els dos llenguatges següents:
1. $L_1=\{w\in\{a\,,b\}^* \mid |w|=\dot{4}\}$.
2. $L_2=\{w\in\{a\,,b\}^* \mid \exists x,y\in\{a\,,b\}^*\ w=xabay\}$.

Definim, a partir d'aquests, el llenguatge L sobre l'alfabet $\{a,b,c\}$

$$L=\{xcy \mid x\in L_1 \ \wedge\ y\in L_2 \ \wedge\ |x|=|y|\}.$$

Construïu una gramàtica incontextual per a L. Generalitzeu aquesta construcció a llenguatges regulars L_1 i L_2 qualssevol.

6.13 Construïu una gramàtica regular per al llenguatge format pels mots que no tenen cap prefix que pertanyi al llenguatge associat a l'expressió regular següent:

$$b^*aba^*b(bb + bab^*a)^*b\,.$$

6.14 Sabem que la família dels llenguatges regulars és la generada a partir dels conjunts finits per les operacions de reunió, concatenació i estrella. Afegir-hi l'operació de complementació no amplia aquesta família. En canvi, ens demanem quines famílies de llenguatges s'obtenen quan se suprimeix l'operació estrella. Suposarem, per tal de simplificar-ho, que ens referim a llenguatges sobre l'alfabet $\{a\,,b\}$.

1. Demostreu que la família de llenguatges generada a partir dels conjunts $\{a\}$, $\{b\}$, $\varnothing$ i $\{\lambda\}$ per les operacions de reunió i concatenació és la família dels conjunts finits.
2. La consideració de la complementació comporta l'aparició de conjunts infinits. Demostreu que la família de llenguatges generada a partir dels conjunts finits per les operacions de reunió i complementació està formada exactament pels conjunts finits i cofinits. Doneu algun exemple de llenguatge regular que no sigui ni finit ni cofinit.
3. Considereu la família de llenguatges generada pels conjunts finits i les operacions de reunió, concatenació i complementació. Demostreu que el llenguatge $(ababa)^*$ pertany a aquesta família. És a dir, trobeu una expressió d'aquest llenguatge en què intervinguin els símbols a, b i els operadors de reunió, concatenació i complementació, però no l'estrella de Kleene.
4. Trobeu algun llenguatge regular que no pertanyi a aquesta última família.

Capítol 7 Propietats d'iteració

Els models estudiats fins ara que caracteritzen les famílies de llenguatges regulars i incontextuals —FA i CFG, respectivament— són clarament de naturalesa o bé iterativa o bé recursiva. Aquest fet es tradueix en una estructura necessàriament repetitiva dels mots acceptats o generats, respectivament, pels models considerats. En aquest capítol estudiarem una sèrie de proposicions, anomenades *lemes de bombament*, que descriuen propietats estructurals dels llenguatges regulars i incontextuals que posen de manifest precisament aquesta característica repetitiva dels seus mots. El fet que aquestes propietats siguin condicions necessàries i de verificació senzilla les converteix en eines d'una gran potència i simplicitat per decidir la no-pertinença de certs llenguatges a aquestes famílies. Addicionalment també ens permetran atacar altres problemes com, per exemple, la demostració de l'ambigüitat inherent d'alguns llenguatges incontextuals.

7.1 Lema de bombament de llenguatges regulars

El resultat següent, conegut com a lema de bombament (*pumping lemma*) per a llenguatges regulars, és degut a Bar-Hillel, Perles i Shamir [BPS61] i descriu la naturalesa repetitiva dels mots dels llenguatges regulars. Aquesta estructura està en correspondència, bàsicament, amb l'aparició de bucles en la computació de mots suficientment llargs sobre autòmats finits deterministes.

PROPOSICIÓ (LEMA DE BOMBAMENT). *En un llenguatge regular L, tot mot w de longitud no inferior a un determinat valor N, propi de L, admet una factorització $w = xyz$ que satisfà*

1. $|xy| \leqslant N$
2. $|y| \geqslant 1$
3. $\forall i \geqslant 0 \;\; xy^iz \in L$

DEMOSTRACIÓ. Considerem un DFA qualsevol, $M = \langle Q, \Sigma, \delta, q_0, F\rangle$, que reconegui el llenguatge L. Sigui N el nombre d'estats de Q i $w \in L$ un mot qualsevol de longitud igual o superior a N (observem que si el llenguatge és finit no existeix cap mot que satisfaci aquesta condició). Escrivint aquest mot com a successió de símbols tenim $w = a_1a_2\cdots a_m$, amb $m \geqslant N$. Com que el mot w és suficientment llarg, el procés dels N primers símbols ha de passar, com a mínim, dues vegades per un mateix estat (hi ha un bucle). Si representem per q_i l'estat $q_0{\cdot}a_1a_2\cdots a_i$, és a dir, l'estat al qual arriba l'autòmat després de processar els i primers símbols de w, la propietat anterior es pot formalitzar dient que existeixen dos naturals j, k $0 \leqslant j < k \leqslant N$ tals que $q_j = q_k$. A més, com que $w \in L$, es té $q_0 w = q_m \in F$. Vegeu la figura 7.1, que il·lustra la computació de w sobre M.

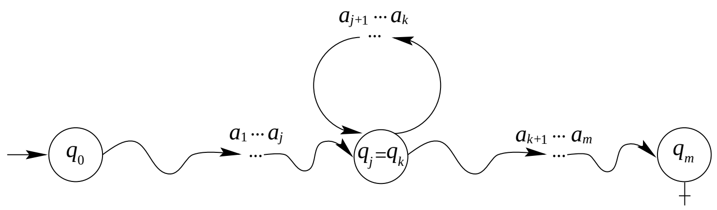

Fig. 7.1 *Computació del mot $w = a_1 a_2 \cdots a_m$ sobre un* DFA

Aquest bucle en la computació del mot sobre l'autòmat ens permet considerar la partició de w en tres trossos, $x = a_1 a_2 \cdots a_j$, $y = a_{j+1} \cdots a_k$ i $z = a_{k+1} \cdots a_m$, que compleixen $|y| \geqslant 1$ i $|xy| \leqslant N$. El submot y és el que provoca el bucle de l'autòmat sobre q_j, és a dir que $q_j \cdot y = q_j$. Trivialment, per inducció es té que per a tot natural $i \geqslant 0$ $q_j y^i = q_j$ i, en aquest cas,

$$q_0 \cdot x y^i z = q_j \cdot y^i z = q_j \cdot z = q_m \in F\,,$$

és a dir que $x y^i z \in L$. □

Informalment, la propietat que hem provat és que, donat un mot w suficientment llarg pertanyent a un llenguatge regular L donat, podem trobar un submot de w a prop del començament (no més enllà de l'N-èsim símbol) tal que els mots resultants de *bombar-lo* (repetir-lo) tantes vegades com es vulgui són tots del llenguatge L.

El lema de bombament s'utilitza per demostrar la no-regularitat d'un llenguatge proposat. Per fer això, n'hi ha prou de provar que el llenguatge donat no satisfà aquesta condició. És a dir, hem de partir del contrarecíproc de l'enunciat anterior.

CONTRARRECÍPROC. *Si L és un llenguatge que satisfà la condició següent:*

"Per a qualsevol natural $N \geqslant 1$ existeix un mot $w \in L$ de longitud igual o superior a N tal que per a tota factorització $w = xyz$ que satisfà les condicions $|xy| \leqslant N$ i $|y| \geqslant 1$ existeix un natural $i \geqslant 0$ tal que $x y^i z \notin L$",

aleshores L no és regular.

Amb vista a la seva aplicació en demostracions, ens interessarà tenir formalitzada la condició de no-regularitat:

$$\forall N \geqslant 1 \;\; \exists w \in L$$
$$|w| \geqslant N \;\wedge\; \forall x, y, z\,((w = xyz \;\wedge\; |xy| \leqslant N \;\wedge\; |y| \geqslant 1) \;\Rightarrow\; \exists i \geqslant 0 \;\; x y^i z \notin L)\,.$$

Vegem a continuació alguns exemples d'aplicació per tal de demostrar la no-regularitat d'una sèrie de llenguatges.

Exemple 7.1 Demostrarem, utilitzant el lema de bombament, que el llenguatge $L = \{a^n b^n \mid n \geqslant 0\}$ no és un llenguatge regular.

Donat un natural $N \geqslant 1$ qualsevol, prenem $w = a^N b^N \in L$, que satisfà $|w| = 2N \geqslant N$. Sigui una factorització qualsevol $w = xyz$ amb $|xy| \leqslant N$ i $|y| \geqslant 1$. S'ha de tenir que

$x = a^j$, $y = a^k$ i $z = a^{N-j-k}b^N$, amb $k \geqslant 1$ i $j + k \leqslant N$. Així doncs, és obvi que per a tot natural $i \neq 1$ el mot xy^iz no és del llenguatge L. Per exemple, si prenem $i = 0$ es té: $xy^0z = xz = a^{N-k}b^N \notin L$. En conseqüència, hem demostrat que L no és regular.

Observeu que la restricció $|xy| \leqslant N$ imposada sobre la factorització de w en x, y i z fa que el submot y a *bombar* (repetir) estigui contingut en els N primers símbols del mot w. Aquest fet és el que ens permet fer una tria adequada de w per poder trencar fàcilment l'estructura dels mots del llenguatge L. En l'exemple precedent, això s'ha concretat prenent un mot w amb un prefix de a's suficientment llarg.

Exemple 7.2 Ens proposem de demostrar que el llenguatge dels palíndroms de longitud parella no és un llenguatge regular.

Sigui $L = \{uu^{\mathrm{R}} \mid u \in \{a, b\}^*\}$. Per a qualsevol natural $N \geqslant 1$, prenem el mot del llenguatge $w = a^N bba^N$ de longitud superior a N. Per a qualsevol factorització $w = xyz$ sota les condicions del lema de bombament s'ha de complir que $x = a^j$, $y = a^k$ i $z = a^{N-j-k}bba^N$. En aquest cas, qualsevol bombament $i \neq 1$ resulta en $xy^iz \notin L$. Per exemple, per a $i = 2$, es té $xy^2z = a^{N+k}bba^N \notin L$. Queda demostrat, doncs, el caràcter no regular del llenguatge L.

Fixeu-vos que, en aquest cas, el bombament modifica el nombre de a's d'abans de la primera b sense tocar el nombre de a's de després de la segona b, de manera que desplaça del centre del mot la subcadena bb. Com que l'única manera que el mot resultant es pugui escriure com la concatenació d'un mot u i el seu revessat u^{R} és que les dues b's formin la subcadena central, el mot resultant del bombament no és del llenguatge L.

Exemple 7.3 Veurem que el llenguatge $L = \{a^n b^m \mid n \neq m\}$ no és regular.

Donat un natural qualsevol $N \geqslant 1$, triarem un mot del llenguatge amb menys a's que b's i mitjançant un bombament positiu adequat igualarem el nombre de a's i de b's i convertirem el mot en un que no sigui del llenguatge. El mot ha de tenir un prefix de N a's com a mínim, ja que volem que el bombament només afecti els símbols a, però amb quantes b's de més hem de partir? D'entrada, posem un nombre $C > 0$ indeterminat i, més tard, ja veurem quin ha de ser el seu valor i quina relació té amb N. Així doncs, triem el mot $w = a^N b^{N+C} \in L$ de longitud superior a N. Sigui una factorització qualsevol $w = xyz$ amb $|xy| \leqslant N$ i $|y| \geqslant 1$. Es té que $x = a^j$, $y = a^k$ i $z = a^{N-j-k}b^{N+C}$, amb $j + k \leqslant N$ i $k \geqslant 1$. Per tant, $xy^iz = a^j a^{ik} a^{N-j-k} b^{N+C} = a^{N+k(i-1)} b^{N+C}$. Si volem tenir tantes a's com b's només hem d'igualar $N + k(i-1)$ a $N + C$ i veure que la i que cal triar és $(C/k) + 1$. Ara bé, i ha de ser un nombre natural i, per tant, hem de garantir que la divisió C/k sigui entera. Com que k és un natural fitat entre 1 i N, hauríem de triar un nombre C que tingui com a mínim tots els divisors entre 1 i N, per exemple, $C = N!$. Ara ja estem en condicions d'escriure la demostració un altre cop des del principi: $w = a^N b^{N+N!} \in L$, i prenent un bombament $i = (N!/k) + 1$ es té $xy^iz = a^{N+k((N!/k)+1-1)} b^{N+N!} = a^{N+N!} b^{N+N!} \notin L$.

Sobre el recíproc del lema de bombament

El lema de bombament descriu una condició necessària de pertinença a la família dels llenguatges regulars, però el seu recíproc és fals. A continuació veurem l'exemple d'un llenguatge que satisfà la condició del lema i que, tanmateix, no és regular.

Exemple 7.4 Demostrarem que el llenguatge $L = \{uu^{\mathsf{R}}v \mid u,v \in \{a\,,b\}^+\}$ compleix la condició del lema de bombament. Prenem la constant $N = 4$. Sigui w un mot del llenguatge L de longitud igual o superior a N. Considerem una descomposició qualsevol de w en la forma $uu^{\mathsf{R}}v$ (el fet que n'hi pugui haver més d'una no afecta la correctesa de la demostració) i representem per u_i l'i-èsim símbol del prefix u i per v_i l'i-èsim símbol del sufix v. D'aquesta manera, $w = u_1u_2\cdots u_nu_nu_{n-1}\cdots u_1v_1v_2\cdots v_m$ per a certs $n,m \geqslant 1$. Distingirem dos casos,

1. $|u| \geqslant 2$. Considerem la factorització $w = xyz$, amb $x = \lambda$, $y = u_1$ i $z = u_2\cdots u_nu^{\mathsf{R}}v$, que satisfà les condicions: $|y| \geqslant 1$ i $|xy| \leqslant N$. Es compleix que, per a tot natural i, $xy^iz \in L$. Vegem-ho. Si $i = 0$, aleshores $xy^0z = z \in L$, ja que és la concatenació del prefix palíndrom $u_2\cdots u_nu_n\cdots u_2$ amb el sufix u_1v, ambdós de longitud superior a un. Si $i = 1$, aleshores $xyz = w \in L$. Finalment, si $i \geqslant 2$, es té $xy^iz = {u_1}^iz \in L$, ja que conté el prefix palíndrom u_1u_1 i el sufix ${u_1}^{i-2}z$ de longitud no nul·la.
2. $|u| = 1$. En aquest cas considerem la factorització $w = xyz$ amb $x = uu^{\mathsf{R}}$, $y = v_1$ i $z = v_2\cdots v_m$. És evident que aquesta factorització satisfà les condicions $|y| \geqslant 1$ i $|xy| \leqslant N$. Observeu que, a més, se satisfà $|z| \geqslant 1$ perquè la longitud de w és més gran o igual que 4. Així, per a tot natural i es té $xy^iz = uu^{\mathsf{R}}{v_1}^iz \in L$, ja que es continua mantenint el prefix palíndrom uu^{R} i el sufix $v' = {v_1}^iz$ no és buit, ni tan sols quan $i = 0$.

Exemple 7.5 Demostrarem ara que el llenguatge

$$L = \{uu^{\mathsf{R}}v \mid u,v \in \{a\,,b\}^+\}$$

no és regular per la via de demostrar que la seva intersecció amb un llenguatge regular tampoc no ho és. Sigui L' el llenguatge intersecció de L i el conjunt regular aba^*bba^*bab. Es compleix que

$$L' = L \cap aba^*bba^*bab = \{aba^nbba^nbab \mid n \geqslant 0\}\,,$$

ja que no és possible construir cap prefix de la forma uu^{R} sense que la subcadena central sigui bb. Això obliga a prendre el mateix nombre de a's a esquerra i dreta de bb. Ara demostrar que L' no és regular és senzill.

Per a qualsevol $N \geqslant 1$ prenem $w = aba^Nbba^Nbab \in L'$, de longitud superior a N. Sigui $w = xyz$ una partició qualsevol amb $|xy| \leqslant N$ i $|y| \geqslant 1$. Necessàriament xy serà un prefix de aba^{N-2} i qualsevol bombament diferent de la unitat alterarà l'estructura de la cadena total i ocasionarà un mot que no pertany al llenguatge L'. En conseqüència, L' no és regular i, per tant, L tampoc ja que, en cas contrari, també ho seria L'.

Verificació de la no-regularitat de llenguatges

Hi ha llenguatges no regulars, com el dels exemples 7.4 i 7.5 anteriors, sobre els quals és impossible aplicar directament el lema de bombament. En d'altres casos, la utilització directa del lema pot ser difícil o simplement laboriosa. Si ens fixem en el procediment que hem seguit per demostrar la no-regularitat del llenguatge de l'exemple 7.5, veurem que ha consistit a obtenir per intersecció amb un llenguatge regular un subconjunt no regular del llenguatge de partida sobre el qual, ara sí, es pot aplicar fàcilment el lema de bombament. Aquest procediment es pot generalitzar de la manera següent:

> *Sigui L el llenguatge de partida i L' el llenguatge resultant d'haver aplicat sobre L una sèrie d'operacions que preserven la regularitat. Si demostrem que L' no és regular, aleshores tenim que L tampoc no ho és, ja que si L fos regular L' també ho seria.*

D'aquesta manera, en molts casos podem disminuir la dificultat del problema i donar demostracions més senzilles i elegants. Algunes transformacions típiques són passar al complementari o al revessat, fer la intersecció amb un llenguatge regular i l'aplicació de morfismes directes o inversos. A continuació veurem exemples en què es posa de manifest la utilitat de cadascuna d'aquestes operacions.

Exemple 7.6 Demostrarem que el llenguatge

$$L=\{a^n \mid n \text{ és un nombre compost}\}$$

no és un llenguatge regular. Si volem atacar el llenguatge L directament amb el lema de bombament ens adonarem que el problema es tradueix a trobar un bombament prou general que ens permeti transformar una cadena de a's de longitud igual a un nombre compost en una cadena de a's de longitud igual a un nombre primer, i que aquest no sembla de cap manera un problema fàcil de resoldre. En canvi, si considerem el llenguatge complementari, el problema es torna prou senzill. Vegem-ho. Sigui

$$\overline{L}=\{a^n \mid n \text{ és un nombre primer}\}\,.$$

Com que el conjunt de nombres primers és infinit, en podem considerar d'arbitràriament grans. Amb això, donat un natural $N \geqslant 1$ qualsevol, podem escollir el mot $w=a^p$, on p és el mínim nombre primer més gran que N. És evident que $w \in \overline{L}$ i que $|w|=p \geqslant N$. Per a qualsevol descomposició de w en x, y, z sota les condicions del lema de bombament tenim que $x=a^j$, $y=a^k$ i $z=a^{p-j-k}$, amb $j+k \leqslant N$ i $k \geqslant 1$. Per tant, $xy^iz=a^ja^{ik}a^{p-j-k}=a^{p+k(i-1)}$. Si prenem el bombament $i=p+1$ ens quedarà $xy^iz=a^{p+kp}=a^{p(k+1)}$, que no és un mot de $\overline{L}$, ja que $p(k+1)$ és un nombre compost. Podem concloure, per tant, que $\overline{L}$ no és un llenguatge regular, i això implica que el seu complementari L tampoc no ho és.

Exemple 7.7 Una manera alternativa de demostrar que el llenguatge presentat a l'exemple 7.3, $L=\{a^nb^m \mid n \neq m\}$, no és un llenguatge regular es basa en el fet que el complementari del llenguatge en qüestió és la reunió d'un llenguatge regular i d'un no regular ja conegut. En concret, el complementari de L és la reunió del llenguatge dels mots de a^*b^* que tenen el mateix nombre de a's i b's amb el conjunt de mots que no estan en a^*b^*. En aquest cas, és evident que

$$L'=\overline{L}\cap a^*b^*=\{a^nb^n \mid n \geqslant 0\}\,.$$

Com que L' és un llenguatge no regular, el llenguatge de partida L tampoc no pot ser-ho, ja que les operacions aplicades (complementació i intersecció amb un llenguatge regular) preserven la regularitat.

Exemple 7.8 Definim els arbres binaris *complets* com aquells que tenen totes les fulles a la mateixa distància de l'arrel. Recordem que a l'exemple 1.11 hem introduït una manera de codificar els arbres binaris. Seguint aquella mateixa notació, definim recursivament el llenguatge dels arbres binaris complets de la manera següent:

$$L = \{0\} \cup \{2ww \mid w \in L\} = \{0, 200, 2200200, 222002002200200, \cdots\}\,.$$

La figura 7.2 conté la representació en forma d'arbre dels quatre primers mots del llenguatge L que corresponen als arbres d'alçària 0, 1, 2 i 3, respectivament.

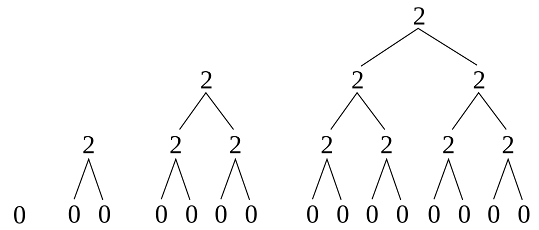

Fig. 7.2 *Els primers mots del llenguatge L en forma d'arbres binaris*

Demostrarem a continuació que aquest llenguatge no és regular. Com a representacions d'arbres binaris complets, els mots de L tenen longituds corresponents a les potències successives de dos menys la unitat (1, 3, 7, 15, 31, etc.). Només per aquest motiu el llenguatge ja no és regular. Amb l'aplicació d'un morfisme podem passar a considerar únicament longituds si confonem els símbols 0 i 2 en un de sol. Vegem-ho. Considerem el morfisme

$$h\colon \{0,2\}^* \longrightarrow \{a\}^*,$$

definit per $h(0)=h(2)=a$. La imatge de L per h és el llenguatge que conté les cadenes de a's de longitud igual al nombre d'elements dels arbres binaris complets,

$$L' = h(L) = \{a^{2^n-1} \mid n \geqslant 1\} = \{a, aaa, aaaaaaa, aaaaaaaaaaaaaaa, \dots\}\,.$$

Donat un natural $N \geqslant 1$ qualsevol, prenem el mot $w = a^{2^N-1} \in L'$, que té longitud $2^N - 1 \geqslant N$. Sigui $w = xyz$ una factorització qualsevol de w que compleixi les condicions del lema de bombament. Necessàriament es té que $x = a^j$, $y = a^k$ i $z = a^{2^N-1-j-k}$, amb $j + k \leqslant N$ i $k \geqslant 1$. Per tant, $xy^iz = a^{2^N-1+k(i-1)}$. Si prenem el bombament $i = 2$, ens quedarà $xy^iz = a^{2^N-1+k}$. Aquest mot no pertany a L' ja que té una longitud que no correspon al nombre d'elements de cap arbre binari complet. Això ho podem veure fitant estrictament la longitud $2^N - 1 + k$ entre dues longituds corresponents al nombre d'elements de dos arbres binaris complets consecutius:

$$2^N - 1 \;<\; 2^N - 1 + k \;<\; 2^{N+1} - 1\,,$$

ja que $1 \leqslant k \leqslant N < 2^N$.

Concloem, doncs, que L' no és regular i, per tant, L tampoc no pot ser-ho, ja que L' és imatge de L via un morfisme.

Exemple 7.9 Suposem que es vol demostrar que el llenguatge

$$L = \{(01)^n(10)^n \mid n \geqslant 0\}$$

no és regular. Si es pren el morfisme $h\colon \{a,b\}^* \longrightarrow \{0,1\}^*$ definit per $h(a)=01$ i $h(b)=10$ i es calcula l'antiimatge del llenguatge L, s'obté

$$L' = h^{-1}(L) = \{a^nb^n \mid n \geqslant 0\}\,.$$

Ja sabem que L' no és un llenguatge regular i, per tant, L tampoc no pot ser-ho, ja que L' és l'antiimatge de L via un morfisme.

Observeu que també podem afirmar que L és la imatge directa per h de L', $L=h(L')$. Però això no ens permet extreure cap conclusió sobre la no-regularitat de L. Tot i que sabem que L' no és regular, la imatge d'un llenguatge no regular pot ser un llenguatge regular.[1]

Exemple 7.10 Si volem demostrar que el llenguatge

$$L=\{baba^2ba^3\cdots ba^n \mid n \geqslant 0\}$$

no és regular ens trobem amb l'inconvenient que els mots de L tenen moltes alternances de a's i b's en els seus primers símbols i això ens obligarà a fer una distinció per casos de les possibles localitzacions del submot y dintre d'un prefix de N símbols. Aquesta dificultat es pot evitar de manera senzilla considerant el revessat

$$L^{\mathsf{R}}=\{a^nba^{n-1}ba^{n-2}\cdots ab \mid n \geqslant 0\}$$

Ara ja tenim prefixos de a's arbitràriament llargs i podem aplicar el lema de bombament de manera senzilla: donat un natural qualsevol $N \geqslant 1$, escollim el mot $w = a^Nba^{N-1}ba^{N-2}\cdots ab \in L^{\mathsf{R}}$. Qualsevol bombament diferent d'1 ens permet veure que $xy^iz \notin L^{\mathsf{R}}$. En conseqüència, L^{R} no és regular i, per tant, L tampoc.

7.2 Lemes de bombament de llenguatges incontextuals

L'estructura repetitiva dels mots dels llenguatges incontextuals ve donada per l'aparició de variables repetides en les cadenes de derivació de mots suficientment llargs. La derivació d'un mot d'aquestes característiques és, en general,

$$S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uvAxy \overset{*}{\Rightarrow} uvwxy\,.$$

És clar que el fragment de derivació $A \overset{*}{\Rightarrow} vAx$ es pot repetir recursivament tantes vegades com es vulgui

$$S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uvAxy \overset{*}{\Rightarrow} uv^2Ax^2y \overset{*}{\Rightarrow} uv^3Ax^3y \;\cdots \overset{*}{\Rightarrow} uv^nAx^ny \overset{*}{\Rightarrow} uv^nwx^ny\,.$$

És a dir que tots els mots de la forma uv^nwx^ny són generats també per la gramàtica i, per tant, pertanyen al mateix llenguatge incontextual considerat.

La proposició següent, deguda a W. Ogden [Ogd68], caracteritza aquesta propietat dels llenguatges incontextuals. Per tal de restringir la zona del mot de partida en la qual s'aplicarà el bombament, el lema d'Ogden considera el *marcatge* d'unes posicions determinades d'aquest mot, les quals tindran una preeminència especial. En aquest context, fer un marcatge d'un mot consisteix a distingir-ne o assenyalar-ne unes quantes posicions. Podem interpretar que distingir una posició en un mot és subratllar el símbol que ocupa aquesta posició dintre de la seqüència total de símbols. És fàcil de veure que el nombre de marcatges possibles d'un mot w donat és $2^{|w|}$. Representem per $\mathcal{M}(w)$ el conjunt de tots els marcatges del mot w. Per a un cert marcatge $\mu \in \mathcal{M}(w)$, representem per $|w|_\mu$ el nombre de símbols marcats (segons μ) de w. Estenem aquesta notació a qualsevol subseqüència de símbols de w.

[1] Considereu l'exemple trivial d'un morfisme h que esborri tots els símbols. La imatge per h de qualsevol llenguatge és sempre el llenguatge regular $\{\lambda\}$.

Exemple 7.11 Donat el mot $w = baaababb$ sobre l'alfabet $\{a\,, b\}$, $\mu_1 = b\underline{a}a\underline{a}\underline{b}a\underline{b}b$ i $\mu_2 = \underline{b}a\underline{a}\underline{a}b\underline{a}b\underline{b}$ són dos marcatges de w. Per a aquests dos marcatges es té $|w|_{\mu_1} = 4$ i $|w|_{\mu_2} = 5$. Considerem ara la factorització $w = xyz$, amb $x = baa$, $y = ab$ i $z = abb$ i el marcatge $\mu = b\underline{a}a\underline{a}\underline{b}a\underline{b}b$. En aquest cas, tenim $|xy|_\mu = 3$ i $|xz|_\mu = 2$.

PROPOSICIÓ (LEMA D'OGDEN). *Per a tota* CFG $G = \langle V, \Sigma, P, S\rangle$, *existeix un natural* $N \geqslant 1$ *tal que, per a tot mot* $z \in \mathsf{L}(G)$ *i per a tot marcatge amb un nombre de posicions distingides igual o superior a* N, *existeix una factorització* $z = uvwxy$ *en què:*

1. *El submot* w *té almenys una posició distingida.*
2. *Els submots* v *i* x *tenen, entre els dos, almenys una posició distingida.*
3. *El submot* vwx *té un màxim de* N *posicions distingides.*
4. *Existeix una variable* $A \in V$ *tal que* $S \stackrel{*}{\Rightarrow} uAy$, $A \stackrel{*}{\Rightarrow} vAx$ *i* $A \stackrel{*}{\Rightarrow} w$. *I com a conseqüència d'això,* $\forall i \geqslant 0\ \ uv^iwx^iy \in \mathsf{L}(G)$.

DEMOSTRACIÓ. Comencem recordant que la relació entre l'alçària d'un arbre i el seu nombre de fulles és exponencial de base igual a l'arietat màxima de l'arbre. Més concretament, per a un arbre k-ari es té que, si l'alçària és h, aleshores el nombre de fulles és menor o igual que k^h (la igualtat es dóna quan l'arbre és complet). Dit amb altres paraules: si un arbre k-ari té un nombre de fulles més gran que k^h, aleshores és que l'arbre té una alçària més gran que h.

Demostrem a continuació el lema d'Ogden. Sigui $G = \langle V, \Sigma, P, S\rangle$ una gramàtica incontextual qualsevol. Representem per H el nombre de variables de V i prenem $k = \mathsf{max}\{|\alpha| \mid X \longrightarrow \alpha \in P\}$. Aquest valor de k, que és la longitud màxima de les parts dretes de les produccions de G, ens dóna l'arietat màxima dels arbres de derivació d'aquesta gramàtica. Prenem com a constant del lema d'Ogden $N = k^H + 1$. Sigui ara un mot $z \in \mathsf{L}(G)$ amb un marcatge qualsevol de N posicions o més. Considerem un arbre de derivació qualsevol de z segons la gramàtica G.

Com que aquest arbre conté *moltes* posicions distingides, segur que és molt profund. Això es tradueix en el fet que almenys conté un camí C des de l'arrel fins a les fulles prou llarg perquè es repeteixin variables en determinats nodes. Dintre d'aquest camí, ens interessen només els nodes *útils* amb vista a la generació de posicions distingides. Anomenem aquests nodes *punts de ramificació* i els definim com els nodes que tenen més d'un fill que o bé és distingit, o bé té descendents distingits.

La definició d'aquest camí es pot fer mitjançant el procediment recursiu següent: el primer node de C és l'arrel de l'arbre. Sigui n el darrer node considerat. Si aquest node és una fulla, aleshores ja hem acabat, altrament n és un node intern de l'arbre de derivació i, en aquest cas, hem d'afegir a C un fill de n que tingui un nombre màxim de descendents distingits i continuar recursivament en aquest punt.

Vegem ara quants punts de ramificació ha de contenir com a mínim aquest camí. Per construcció de C se satisfà:

1. Cada punt de ramificació conserva, com a mínim, una proporció $1/k$ dels descendents distingits que té el punt de ramificació immediatament precedent.
2. Al llarg dels nodes intermedis entre dos punts de ramificació consecutius el nombre de descendents distingits es manté.

Com que el node arrel té $N = k^H + 1$ o més descendents distingits, C ha de tenir, com a mínim, $H+1$ punts de ramificació i, per tant, entre els $H+1$ darrers n'hi ha d'haver dos que repeteixin variable. Sigui A aquesta variable i siguin n_1 i n_2 els punts de ramificació, amb n_2 el de profunditat major. Aquesta situació ens dóna una factorització $z = uvwxy$ del mot original tal com està representat a la figura 7.3.

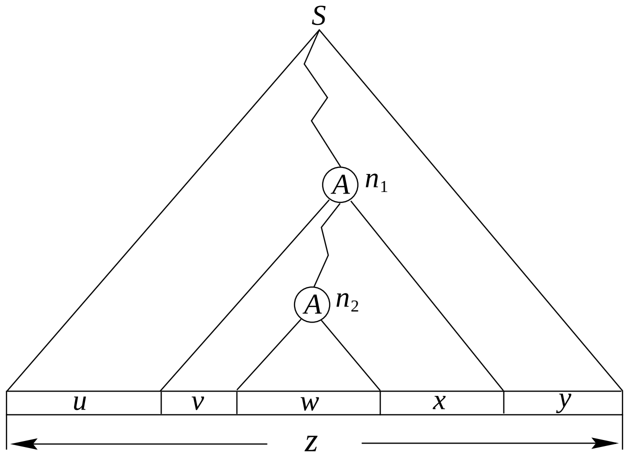

Fig. 7.3 *Factorització del mot z en: u, v, w, x i y*

D'aquesta factorització és clar que

$$S \stackrel{*}{\Rightarrow} uAy, \ \ A \stackrel{*}{\Rightarrow} vAx \text{ i } A \stackrel{*}{\Rightarrow} w\,.$$

Vegem finalment que els submots satisfan les condicions que exigeix el lema:

1. El nombre de posicions distingides de w és no nul ja que l'arrel del subarbre que el genera és el punt de ramificació n_2.
2. El nombre de posicions distingides de vx és no nul ja que, des del punt de ramificació n_1, es genera el submot vwx i, com que n_1 ha de tenir més d'un fill amb descendents distingits, aquests no poden estar concentrats tots en el submot w.
3. El nombre de posicions distingides de vwx és menor o igual que N ja que el punt de ramificació n_1, que és l'arrel d'aquest subarbre, està entre els $H+1$ més profunds.

□

De manera semblant a com passava amb el lema de bombament per a llenguatges regulars, el lema d'Ogden descriu una condició necessària però no suficient d'incontextualitat i, per tant, necessitem treballar amb el contrarecíproc per demostrar la no-pertinença de determinats llenguatges a la família dels incontextuals.

Contrarrecíproc. *Si L és un llenguatge que satisfà la condició següent:*

"Per a tot natural $N \geqslant 1$ existeix un mot $z \in L$ i un marcatge μ de N o més posicions d'aquest mot tal que per a qualsevol factorització $z = uvwxy$ que satisfaci les condicions $|vwx|_\mu \leqslant N$, $|w|_\mu \geqslant 1$ i $|vx|_\mu \geqslant 1$ existeix un natural $i \geqslant 0$ amb $uv^iwx^iy \notin L$",

aleshores L no és un llenguatge incontextual.

Amb vista a la seva aplicació en demostracions, ens interessarà tenir formalitzada aquesta condició de no-incontextualitat. Utilitzant la notació introduïda al començament d'aquesta secció es té

$$\forall N \geqslant 1 \ \exists z \in L \ \exists \mu \in \mathcal{M}(z) \ (|z|_\mu \geqslant N \ \wedge \ \forall u, v, w, x, y \ ((z = uvwxy \ \wedge \\ |vwx|_\mu \leqslant N \ \wedge \ |w|_\mu \geqslant 1 \ \wedge \ |vx|_\mu \geqslant 1) \Rightarrow \exists i \geqslant 0 \ uv^iwx^iy \notin L)) .$$

En cas que el marcatge considerat faci referència a totes les posicions del mot, s'obté un cas particular (més feble) del lema d'Ogden en el qual les consideracions sobre el nombre de posicions distingides passen a ser consideracions sobre longituds en nombre de símbols. Aquest resultat, anterior al Lema d'Ogden, és degut a Bar-Hillel, Perles i Shamir [BPS61].

COROL·LARI (LEMA DE BAR-HILLEL). *Si L és un llenguatge que satisfà la condició*

$$\forall N \geqslant 1 \ \exists z \in L \ (|z| \geqslant N \ \wedge \ \forall u, v, w, x, y \ ((z = uvwxy \ \wedge \\ |vwx| \leqslant N \ \wedge \ |w| \geqslant 1 \ \wedge \ |vx| \geqslant 1) \Rightarrow \exists i \geqslant 0 \ uv^iwx^iy \notin L)) ,$$

aleshores L no és un llenguatge incontextual.

Vegem amb uns quants exemples l'aplicació d'aquests lemes i quina relació hi ha entre la forma feble i la forma forta. Comencem per un cas que es pugui resoldre aplicant el lema de Bar-Hillel.

Exemple 7.12 Considerem el llenguatge $L = \{a^nb^nc^n \mid n \geqslant 0\}$. Donat un natural $N \geqslant 1$ qualsevol, prenem el mot $z = a^Nb^Nc^N$ de longitud superior a N. Considerem ara qualsevol factorització $z = uvwxy$ que compleixi les condicions del lema. Com que la darrera a de z està a distància $N+1$ de la primera c i $|vwx| \leqslant N$ es té que el submot vwx que inclou la part que s'ha de bombar no pot contenir alhora símbols a, b i c. Si hi afegim que $|vx| \geqslant 1$, resulta que qualsevol bombament diferent de la unitat trencarà la igualtat entre el nombre de símbols a, b i c.

En detall, els diferents casos surten de considerar totes les possibilitats de situar els submots per bombar v i x dintre de z: 1) Si $vx \in a^+$, aleshores, per a un bombament $i = 0$, es té $uv^0wx^0y = uwy \notin L$ ja que té menys a's que b's i c's. 2) Els dos casos, $vx \in b^+$ i $vx \in c^+$, es tracten manera anàloga. 3) Si vx conté a's i b's, aleshores, per a un bombament $i = 0$, es té $uwy \notin L$ ja que té més c's que a's i b's. 4) El darrer cas que es dóna quan vx conté b's i c's es resol anàlogament. En conseqüència, L no és un llenguatge incontextual.

La idea essencial per aplicar el lema de Bar-Hillel és tenir en compte el fet que els submots v i x no poden estar gaire separats l'un de l'altre (en concret, no més de N símbols). A l'exemple anterior, hem aprofitat aquest fet per assegurar que el bombament no podia afectar símbols a, b i c alhora.

El fet de marcar algunes posicions concretes de z utilitzant el lema d'Ogden permet restringir el nombre de casos de bombament, ja que podem assegurar que almenys un dels submots v i x ha d'estar localitzat en una zona concreta de z. De tota manera, perdem la condició sobre la longitud màxima del submot vwx que teníem en el lema de Bar-Hillel, ja que ara només estem autoritzats a suposar que el nombre de posicions distingides de vwx (però no la seva longitud) és menor o igual que N. Vegem-ho en un exemple.

Exemple 7.13 Considerem el llenguatge $L=\{a^ib^jc^k \mid i \leqslant j \leqslant k\}$. Apliquem el lema d'Ogden per demostrar que no és un llenguatge incontextual. Sigui un natural $N \geqslant 1$ qualsevol. Elegim el mot $z = a^N b^{N+1} c^{N+2} \in L$ i el marcatge $\mu = \underline{a}^N b^{N+1} c^{N+2}$ que distingeix les N primeres posicions. Considerem qualsevol factorització $z=uvwxy$ que compleixi les condicions del lema d'Ogden i vegem quins són els casos possibles per als submots v i x.

El fet que $|w|_\mu \geqslant 1$ obliga que $v \in a^*$. A més x ha de ser uniliteral, altrament prenem un bombament $i = 2$ i tenim $uv^2wx^2y \notin L$ ja que ni tant sols pertany a $a^*b^*c^*$ (per exemple, si x conté a's i b's aleshores en uv^2wx^2y apareixerà alguna b abans que alguna a). Estudiem els casos que es presenten:

1. Si $v=\lambda$, aleshores $x \in a^+$ i un bombament per $i=2$ produirà $uv^2wx^2y \notin L$ ja que contindrà un nombre de a's més gran o igual que el nombre de b's.
2. Si $v \in a^+$, aleshores apareixen quatre casos per al submot x. Si $x=\lambda$, $x \in a^+$ o $x \in c^+$, triem com abans un bombament $i=2$ i podem assegurar que el mot uv^2wx^2y té un nombre de a's igual o superior al nombre de b's. Finalment, si $x \in b^+$ i triem també $i=2$, es compleix que el mot uv^2wx^2y té un nombre de b's més gran o igual que el nombre de c's i tampoc no pertany a L.

Posarem ara de manifest que el lema d'Ogden és més fort i permet tractar més casos que no el de Bar-Hillel, presentant l'exemple d'un llenguatge no incontextual sobre el qual no és aplicable el lema de Bar-Hillel i sí, en canvi, el d'Ogden.

Exemple 7.14 Considerem el llenguatge

$$L=\{a^ib^jc^kd^l \mid i=0 \ \vee \ j=k=l\}\,.$$

Demostrarem que no podem aplicar a L el lema de Bar-Hillel. Observeu que L es pot expressar com a reunió dels llenguatges disjunts

$$b^*c^*d^* \ \cup \ a^+\{b^nc^nd^n \mid n \geqslant 0\}\,.$$

En tenim prou a prendre $N=2$. Sigui z un mot qualsevol de L de longitud més gran o igual que dos. Hi ha dos casos:

1. Si $z \in b^*c^*d^*$, aleshores es tracta de triar una factorització que permeti un bombament uniliteral (per exemple, $v \in b^+$ i $x=\lambda$ si el mot conté b's) i segur que, per a tot $i \geqslant 0$, es té $uv^iwx^iy \in b^*c^*d^* \subseteq L$.
2. Si $z \in a^+\{b^nc^nd^n \mid n \geqslant 0\}$, aleshores segur que podem triar una factorització de l'estil $v=a$ i $x=\lambda$. Si el nombre de a's de z és superior a 1, aleshores, per a tot $i \geqslant 0$, es té $uv^iwx^iy \in a^+\{b^nc^nd^n \mid n \geqslant 0\} \subseteq L$. Si el mot z només tenia una a, aleshores, per al bombament $i=0$, es té $uwy \in b^*c^*d^* \subseteq L$ mentre que qualsevol bombament positiu $i \geqslant 1$ dóna $uv^iwx^iy \in a^+\{b^nc^nd^n \mid n \geqslant 0\} \subseteq L$.

Exemple 7.15 Considerem el mateix llenguatge L que en l'exemple precedent. Demostrarem que no és un llenguatge incontextual utilitzant el lema d'Ogden.

Donat un natural qualsevol $N \geqslant 1$, prenem el mot $z = ab^N c^N d^N \in L$ i el marcatge $\mu = ab^N c^N \underline{d^N}$ de N posicions. Considerem ara qualsevol factorització $z = uvwxy$ que compleixi les condicions del lema d'Ogden. Si v o x no són uniliterals, aleshores, per a qualsevol bombament amb $i \geqslant 2$, es té $uv^iwx^iy \notin L$ ja que $uv^iwx^iy \notin a^*b^*c^*d^*$. Si v i x són uniliterals, aleshores per força $x \in d^*$ ja que si la x no contingués cap d, aleshores v tampoc no contindria posicions distingides i, per tant, es tindria $|vx|_\mu = 0$.

1. Si $x = \lambda$, aleshores $v \in d^+$ i un bombament amb $i = 0$ produirà $uwy \notin L$ ja que contindrà el mateix prefix no nul de a's però menys d's que b's i c's.
2. Si $x \in d^+$, aleshores cal considerar les cinc possibilitats per a v. Si $v = \lambda$, $v \in a^+$ o $v \in d^+$, aleshores un bombament per $i = 2$ ens assegura que el mot uv^2wx^2y continua tenint un prefix no nul de a's i un nombre de d's estrictament superior al nombre de b's i c's. Els dos casos restants, $v \in b^+$ i $v \in c^+$, es poden resoldre també amb un bombament $i = 2$, ja que s'obtenen mots uv^2wx^2y que conserven un prefix no nul de a's i que tenen un nombre de d's estrictament superior al nombre de c's i b's, respectivament.

7.3 Llenguatges inherentment ambigus

En aquesta secció posarem de manifest l'existència de llenguatges incontextuals inherentment ambigus. Veurem que el lema d'Ogden és també una eina adequada per raonar sobre l'ambigüitat de les gramàtiques incontextuals i això serà la base de la demostració que el llenguatge incontextual $\{a^ib^jc^k \mid i = j \ \vee \ j = k\}$, presentat a la secció 2.3 del capítol 2, és inherentment ambigu.

PROPOSICIÓ. *El llenguatge $L = \{a^ib^jc^k \mid i = j \ \vee \ j = k\}$ és inherentment ambigu.*

DEMOSTRACIÓ. Sigui $G = \langle V, \Sigma, P, S\rangle$ una gramàtica incontextual qualsevol que generi L. Sigui N' la constant del lema d'Ogden per a aquesta gramàtica. Considerem N el màxim entre N' i 3. A continuació, provarem que existeixen dues derivacions diferents del mot $a^{N+N!}b^{N+N!}c^{N+N!} \in L$ per la gramàtica G. Dividim la demostració en tres parts.

- *Primera part*

Considerem el mot $z = a^N b^N c^{N+N!} \in L$ i el marcatge $\mu = \underline{a^N} b^N c^{N+N!}$ que en distingeix les N primeres posicions. El lema d'Ogden ens assegura l'existència d'una factorització $z = uvwxy$ que satisfà determinades condicions.

En primer lloc, pel fet que el submot w ha de contenir almenys una posició distingida, és necessari que $u, v \in a^*$.

En segon lloc, x no pot contenir dos símbols diferents ja que, en aquest cas, en fer un bombament per $i = 2$, el mot resultant uv^2wx^2y no seria de $a^*b^*c^*$ i, per tant, no pertanyeria a L. En conseqüència, x ha de ser un mot de $a^* + b^* + c^*$. Ara bé, x no pot ser de a^*, ja que, en aquest cas, tindríem $vx = a^p$ amb $p \geqslant 1$ i, per tant, $uv^2wx^2y = a^{N+p}b^N c^{N+N!} \notin L$. Tampoc no es pot donar la possibilitat $x \in c^*$, ja que si $x = c^q$, aleshores necessàriament $v = a^p$, amb $p \geqslant 1$, perquè x no té cap posició distingida. En aquest cas, un bombament per $i = 2$ dóna com a resultat $uv^2wx^2y = a^{N+p}b^N c^{N+N!+q} \notin L$.

En conseqüència, $x \in b^*$. Sigui $x = b^q$, amb $q \leqslant N$, i sigui $v = a^p$, amb $p \geqslant 1$. Amb un bombament per $i = 2$ obtenim $uv^2wx^2y = a^{N+p}b^{N+q}c^{N+N!}$. Com que aquest mot ha de pertànyer a L, i com que $q \neq N!$, ja que $q \leqslant N$ i $N \geqslant 3$, resulta necessàriament que $p=q$. La factorització que estem considerant és, per tant, de la forma:

$$z = \underbrace{a^{N-p-r}}_{u}\,\underbrace{a^p}_{v}\,\underbrace{a^r b^s}_{w}\,\underbrace{b^p}_{x}\,\underbrace{b^{N-p-s}c^{N+N!}}_{y}$$

amb $1 \leqslant p \leqslant N$.

Sabem, per aplicació del lema d'Ogden, que existeix una variable $A \in V$ associada a aquesta factorització tal que, per a tot $i \geqslant 0$, es té una derivació del tipus

$$S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uv^iwx^iy \in L\,.$$

Si prenem $i=(N!/p)+1$ (valor sempre enter pel fitament de p), en resulta la derivació

$$S \overset{*}{\Rightarrow} a^{N-p-r}Ab^{N-p-s}c^{N+N!} \overset{*}{\Rightarrow} a^{N+N!}b^{N+N!}c^{N+N!}$$

on la derivació de la dreta correspon a

$$A \overset{*}{\Rightarrow} a^{N!+p+r}b^{N!+p+s}\,.$$

- *Segona part*

Considerem ara el mot $z'=a^{N+N!}b^Nc^N \in L$ del qual en distingim les N darreres posicions. Un raonament simètric al de l'apartat 1 ens portaria a establir l'existència d'una derivació del tipus

$$S \overset{*}{\Rightarrow} a^{N+N!}b^{N-p'-s'}Bc^{N-p'-r'} \overset{*}{\Rightarrow} a^{N+N!}b^{N+N!}c^{N+N!}$$

on la derivació de la dreta correspon a

$$B \overset{*}{\Rightarrow} b^{N!+p'+s'}c^{N!+p'+r'}\,.$$

- *Tercera part*

Es tracta de demostrar que les dues derivacions del mot $a^{N+N!}b^{N+N!}c^{N+N!} \in L$, construïdes a les parts primera i segona, no poden correspondre a un mateix arbre de derivació i que, per tant, la gramàtica G és ambigua.

Suposem que les dues derivacions anteriors corresponguessin a un mateix arbre de derivació. Com que la variable A genera a's i b's però no c's i, en canvi, la variable B genera b's i c's però no a's, els nodes interns de l'arbre corresponents a aquestes variables no poden ser un d'ells antecessor de l'altre. Per tant, l'arbre considerat seria de la forma representada a la figura 7.4.

Però, en aquest cas, el nombre de símbols b generats ha de ser, com a mínim,

$$N!+p+s\;+\;N!+p'+s'\,,$$

valor que és forçosament superior al total resultant de $N+N!$. Aquesta contradicció invalida la suposició que les dues derivacions corresponen a un únic arbre de derivació.

□

Exercicis

7.1 Els llenguatges següents són tots regulars. Doneu subconjunts no regulars de cadascun d'ells.

1. $((a+b)^*c)^*(a+b)^*$
2. $\{w \in \{a\,,b\}^* \mid |w| = \dot{2}\}$

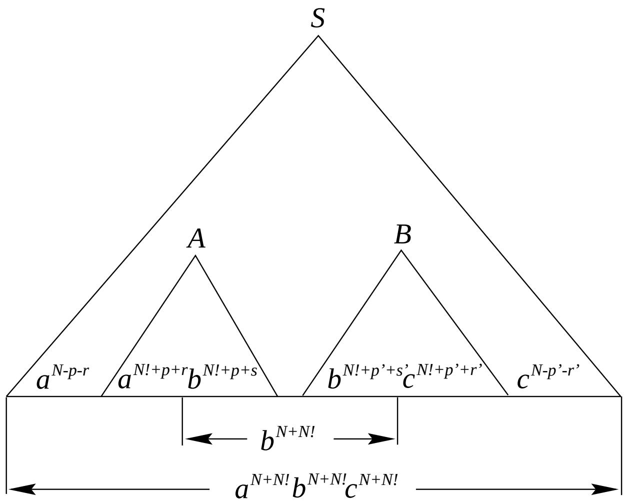

Fig. 7.4 *Les dues derivacions del mot* $a^{N+N!}b^{N+N!}c^{N+N!}$ *en un sol arbre*

3. $\{xcy \mid x, y \in \{a\,, b\}^*\}$

7.2 Demostreu que hi ha llenguatges incontextuals lineals que no són regulars.

7.3 Demostreu que els llenguatges següents no són regulars:

1. El llenguatge de les expressions regulars sobre l'alfabet $\{\cdot, +, *, (,), \Lambda, \varnothing, a, b\}$.
2. $\{a^i b^j c^k \mid j = \mathsf{max}(i,k)\}$
3. $\{a^i b^j \mid \mathsf{mcd}(i,j) = 1\}$
4. $\{a^i b^j \mid i \neq j \ \wedge \ i \neq 2j\}$
5. $\{a^i b^j c^k \mid i \neq j \ \vee \ j \neq k\}$
6. $\{a^i b^j c^k d^l \mid i = k \ \vee \ j = l\}$
7. $\{w \in (a+b+c)^* \mid |w|_a \leqslant |w|_b \ \vee \ |w|_a \leqslant |w|_c\}$
8. $\{w \in (a+b+c)^* \mid \exists u, v \in (a+b+c)^* \ \ w = uv \ \wedge \ |u| = |v|^2\}$
9. $\{a^n ww \mid n \geqslant 0 \ \wedge \ w \in (a+b)^*\}$
10. $\{xcy \mid x, y \in (a+b)^* \ \wedge \ x^{\mathsf{R}}$ és submot de $y\}$
11. $\{xcy \mid x, y \in (a+b)^* \ \wedge \ (|x| < |y| \ \vee \ |x| = \dot{2})\}$
12. $\{xcy \mid x, y \in (a+b)^* \ \wedge \ (x = y \ \vee \ |x| \neq \dot{5})\}$
13. $\{aba^2ba^3b \cdots a^{n-1}ba^nb \mid n > 1\}$
14. $\{ab^{i_1}ab^{i_2} \cdots ab^{i_n} \mid n \geqslant 1 \ \wedge \ i_1, \ldots, i_n \geqslant 1 \ \wedge \ (\exists j \ 1 \leqslant j \leqslant n \ \ i_j = j)\}$

7.4 Doneu un subconjunt infinit i regular del llenguatge $\{ww^{\mathsf{R}} \mid w \in (a+b)^*\}$. Demostreu que, en canvi, cap subconjunt infinit del llenguatge $\{wcw^{\mathsf{R}} \mid w \in (a+b)^*\}$ no pot ser regular.

7.5 Sigui f una funció numèrica definida sobre els naturals. Preneu el llenguatge $L = \{a^n b^{f(n)} \mid n \geqslant 0\}$ i discutiu la seva regularitat en cadascun dels casos següents:

1. f constant.
2. f constant per trams i definida només per un nombre finit de trams.
3. f bijectiva.
4. $\mathsf{Imatge}(f)$ finit.

7.6 Siguin f i g dues funcions funcions numèriques *creixents* definides sobre els naturals. Considereu els dos llenguatges següents, que suposarem regulars:

$$A = \{a^{f(n)} \mid n \geqslant 0\} \ \text{ i } \ B = \{a^{g(n)} \mid n \geqslant 0\}.$$

Considereu també els llenguatges

$$L_1 = \{a^{f(n)+g(n)} \mid n \geqslant 0\} \ \text{ i } \ L_2 = \{a^{f(n)\times g(n)} \mid n \geqslant 0\}.$$

Demostreu que L_1 ha de ser regular i doneu un contraexemple que provi que L_2 pot no ser-ho.

7.7 Siguin $L_1 = \{a^{n^2} \mid n \geqslant 0\}$ i $L_2 = \{a^{2^n} \mid n \geqslant 0\}$ dos llenguatges definits sobre l'alfabet uniliteral $\Sigma = \{a\}$.

1. Demostreu que L_1 i L_2 no són regulars.
2. Demostreu que L_1 i L_2 no són incontextuals.

7.8 En el supòsit de considerar només alfabets Σ uniliterals, demostreu les dues afirmacions següents:

1. El lema de bombament de Bar-Hillel és una condició necessària i *suficient* de regularitat.
2. Per a tot llenguatge L sobre Σ, L és incontextual si i només si L és regular.

7.9 Demostreu la versió següent, més potent, del lema de bombament dels llenguatges regulars:

LEMA. Si L és un llenguatge regular, aleshores

$\exists N > 0 \ \ \forall w \in L \ \ \forall w_1, w_2, w_3 \in \Sigma^*$

$$(w = w_1 w_2 w_3 \wedge |w_2| = N) \implies \exists x, y, z \in \Sigma^* \ \ w_2 = xyz \wedge |y| \geqslant 1 \ \wedge \ (\forall i \geqslant 0 \ \ w_1 x y^i z w_3 \in L).$$

Es demana:

1. Utilitzeu el lema per demostrar directament la no-regularitat del llenguatge
$$\{r r^{\mathsf{R}} s \mid r, s \in \{a, b\}^*\}.$$
2. Demostreu que el lema de Bar-Hillel es dedueix de l'anterior.

7.10 Demostreu que els llenguatges següents no són incontextuals:

1. $\{w \in (a+b+c)^* \mid |w|_a = |w|_b = |w|_c\}$
2. $\{ww \mid w \in (a+b)^*\}$
3. $\{a^i b^j \mid j = i^2\}$
4. $\{a^i b^i c^j \mid i \neq j\}$
5. $\{a^i b^j c^k \mid i \neq j \ \wedge \ j \neq k\}$
6. $\{a^i b^j c^k \mid i \neq j \ \wedge \ j \neq k \ \wedge \ i \neq k\}$
7. $\{a^i b^j c^k \mid j = \mathsf{max}(i,k)\}$
8. $\{a^i b^j c^i d^j \mid i, j \geqslant 1\}$

7.11 Demostreu el *lema de bombament* següent per a llenguatges *lineals*:

LEMA DE BOMBAMENT. Si L és un CFL lineal, llavors

$$\exists N > 0 \ \ \forall z \in L \quad |z| > N \implies \exists u, v, w, x, y \ \ z = uvwxy \ \wedge \ |uvxy| \leqslant N \ \wedge \ |vx| \geqslant 1 \ \wedge \ (\forall i \geqslant 0 \ \ uv^i w x^i y \in L).$$

7.12 Demostreu, per aplicació del lema de bombament precedent, que els llenguatges definits als exercicis 2.8 i 2.9 no són lineals.

Capítol 8 Autòmats amb pila

8.1 Introducció

En aquest capítol presentarem un nou model de computació: l'*autòmat amb pila* o PDA (de l'anglès *push-down automaton*). Informalment, un PDA és un autòmat finit que disposa d'una memòria amb estructura de pila on emmagatzema símbols de treball. La utilització d'aquesta pila permet als PDA fer determinats comptatges i comparacions sobre els mots d'entrada i, en general, possibilita guardar més informació que la que estrictament es pot codificar en els estats d'un autòmat finit. Un PDA, com el que està representat esquemàticament a la figura 8.1, és doncs un model reconeixedor amb un control finit i una memòria potencialment infinita.

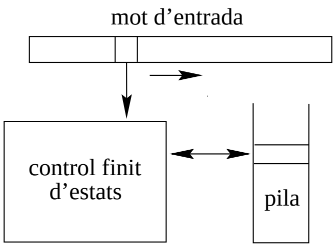

Fig. 8.1 *Esquema d'un autòmat amb pila*

Començarem, com al capítol 4, definint el model determinista, que posteriorment estendrem al cas indeterminista. Veurem que, en la seva versió indeterminista, els PDA són equivalents a les CFGs i que, per tant, suposen una manera alternativa de caracteritzar la família dels llenguatges incontextuals. D'altra banda, els PDA deterministes constitueixen una eina eficient d'anàlisi sintàctica per a una extensa gamma de llenguatges incontextuals i són molt importants en l'àrea de la compilació de llenguatges de programació. Els PDAdeterministes, però, caracteritzen només un subconjunt estricte dels llenguatges incontextuals. De tota manera, aquesta família és prou àmplia per abastar la definició dels llenguatges de programació habituals.

Convé advertir al lector que, per tal de no carregar excessivament el text, les demostracions corresponents a l'equivalència entre gramàtiques incontextuals i autòmats amb pila han estat desplaçades en forma d'annexos al final del capítol. Remarquem, també, que la demostració de la propietat de tancament dels DCFL respecte de la complementació és presentada només en línies generals, sense incidir en els aspectes més tècnics que poden dificultar la comprensió d'una construcció conceptualment senzilla.

8.2 Autòmats amb pila deterministes

Un *autòmat amb pila determinista* (abreujadament un DPDA, de l'anglès *deterministic push-down automaton*) és una estructura de la forma

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle \,,$$

els components de la qual es defineixen a continuació

- Q és un conjunt finit no buit d'estats.
- Σ és un alfabet (anomenat d'*entrada*).
- Γ és un alfabet (anomenat de *pila*).
- δ és la funció de transició que definirem tot seguit.
- $q_0 \in Q$ és l'estat inicial.
- $Z_0 \in \Gamma$ és el símbol de *fons de pila*.
- $F \subseteq Q$ és el conjunt d'estats *acceptadors*.

La funció de transició és una aplicació de la forma

$$\delta\colon Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \longrightarrow Q \times \Gamma^* \,,$$

que és *parcial*, és a dir que no està necessàriament definida per a tots els punts del conjunt de partida. Satisfà, a més, la condició següent, anomenada *condició de determinisme*: per a tot estat p i tot símbol de pila Z, si $\delta(p, \lambda, Z)$ està definida, aleshores $\delta(p, a, Z)$ no ho està per a cap símbol a de Σ.

Expressarem les transicions $\delta(p, a, Z) = (q, \alpha)$ en la forma $Zpa \vdash \alpha q$,[1] on els tres components del domini de definició apareixen simplement concatenats, sense cap separació entre ells. La funció de transició es pot veure, per tant, com un subconjunt finit del producte cartesià

$$\Gamma Q(\Sigma \cup \{\lambda\}) \times \Gamma^* Q \,,$$

que compleix unes condicions determinades.

Els criteris de notació que farem servir són coherents amb els que hem donat en els capítols precedents. Usarem les darreres lletres minúscules de l'alfabet llatí ($\cdots$ w, x, y, z) per representar els mots sobre Σ, les majúscules llatines (A, B, ..., Z) per representar els símbols de l'alfabet de pila i les lletres gregues minúscules (α, β, γ, $\ldots$, ω) per representar els mots sobre l'alfabet de pila Γ. A més, per evitar confusions, procurarem que els alfabets d'entrada i de pila siguin disjunts.

De manera semblant als autòmats finits, els autòmats amb pila es poden descriure mitjançant diagrames de transicions. Es tracta de grafs dirigits que tenen per vèrtexs els estats de l'autòmat i tals que cada transició de la forma $Zpa \vdash \alpha q$ està representada per un arc que va de l'estat p a l'estat q, etiquetat amb $Za|\alpha$ (vegeu la figura 8.2). L'estat inicial el representarem amb una fletxa que hi incideix i els estats acceptadors els identificarem amb una creueta.

[1] Això ens permetrà, com veurem més endavant, expressar una seqüència de transicions d'un PDA com l'aplicació d'un conjunt de regles de reescriptura.

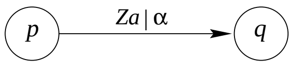

Fig. 8.2 *Una transició* $Zpa \vdash \alpha q$

Exemple 8.1 La figura 8.3 representa un DPDA de dos estats $\{q_0, q_1\}$, que té per estat acceptador l'estat q_0, per alfabet d'entrada els símbols a i b i per alfabet de pila els símbols A, B i Z_0.

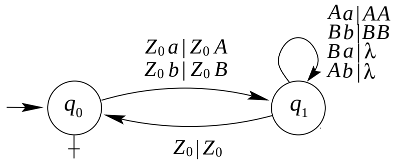

Fig. 8.3 *Un autòmat amb pila determinista*

Observeu que la funció de transició del DPDA precedent és parcial —per exemple, no està definida en (q_0, a, A)— i que compleix la condició de determinisme exigida: com que està definida en (q_1, λ, Z_0), aleshores no està definida en (q_1, a, Z_0) ni en (q_1, b, Z_0).

El funcionament d'un DPDA es pot descriure de la manera següent: el control de l'autòmat comença en l'estat inicial amb el símbol Z_0 com a únic element de la pila i, a partir d'aquest moment, llegeix un a un i d'esquerra a dreta els símbols del mot d'entrada, canviant d'estat i modificant el contingut de la pila segons que indiqui la funció de transició. Cada pas del càlcul depèn únicament de l'estat actual, del símbol llegit en el mot d'entrada i del símbol del cim de la pila. La transició corresponent indica quin ha de ser el nou estat i quin ha de ser el mot que s'ha d'afegir a la pila una vegada eliminat el símbol del cim. En la notació $Zpa \vdash \alpha q$, p és l'estat actual, q el nou, a el símbol llegit de l'entrada, Z el símbol del cim de la pila i α el mot que s'afegeix a la pila, llegit de baix cap a dalt.[2] Una vegada llegit tot el mot de l'entrada, aquest és acceptat si i només si s'accedeix a algun estat acceptador. Si en un moment donat del càlcul la funció de transició es troba indefinida i queden símbols de l'entrada davant o a la dreta del capçal, l'autòmat s'atura i rebutja el mot d'entrada.

La definició que hem donat de la funció de transició d'un DPDA admet transicions de la forma $Zp\lambda \vdash \alpha q$ (que escriurem $Zp \vdash \alpha q$). Aquestes transicions, que anomenem *λ-transicions*, tenen la particularitat de permetre un canvi d'estat i de contingut de la pila sense consumir cap símbol del mot d'entrada. Observeu que l'indeterminisme que podria aparèixer a cada pas de càlcul, per la tria de llegir o no un nou símbol del mot d'entrada, és eliminat per la condició exigida a la funció de transició que força que només es pugui realitzar una λ-transició quan cap altra opció no és possible. Més endavant discutirem la necessitat i les implicacions d'aquestes λ-transicions.

[2] Convé advertir que hi ha textos que segueixen la convenció contrària: el símbol de més a l'esquerra de α és el de la part superior de la pila.

Llenguatge reconegut per un DPDA

Per caracteritzar formalment el llenguatge associat a un autòmat amb pila determinista $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ qualsevol, necessitem definir primer els conceptes de configuració i de transició directa entre configuracions.

DEFINICIÓ. Anomenem *configuració* (o *descripció instantània*) tot mot $\alpha p x \in \Gamma^* Q \Sigma^*$, on p representa l'estat actual, x el sufix de l'entrada comprès entre el símbol que està llegint el capçal i l'extrem dret, i $\alpha \in \Gamma^*$ el contingut de la pila, llegit de baix cap a dalt.

DEFINICIÓ. Diem que existeix una *transició directa* (o un *pas directe*) d'una configuració $\gamma p x$ a una altra $\gamma' q x'$, i ho representem per $\gamma p x \vdash_{\overline{M}} \gamma' q x'$ (o simplement per $\gamma p x \vdash \gamma' q x'$ quan M estigui sobreentès) quan existeix $a \in \Sigma \cup \{\lambda\}$ tal que $x = a x'$, existeixen $Z \in \Gamma$ i $\alpha, \beta \in \Gamma^*$ tals que $\gamma = \alpha Z$, $\gamma' = \alpha\beta$ i la funció δ conté la transició $Zpa \vdash \beta q$.

En altres paraules, el pas directe de la configuració $\alpha Z p a x'$ a la configuració $\alpha\beta q x'$, utilitzant la transició $Zpa \vdash \beta q$, no és res més que reescriure el submot Zpa de la primera configuració en βq per tal d'obtenir la segona. La figura 8.4 representa esquemàticament una transició directa.

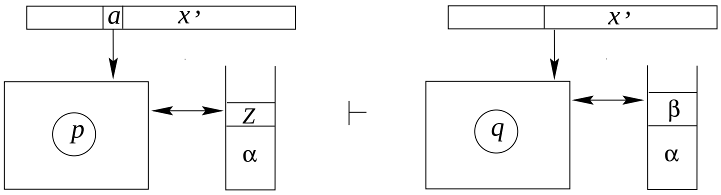

Fig. 8.4 *Transició directa de la configuració $\alpha Z p a x'$ a la configuració $\alpha\beta q x'$*

En tant que $\vdash$ és una relació entre parells de configuracions, podem considerar el seu tancament reflexiu i transitiu, el qual representarem per $\overset{*}{\vdash}$. Així, diem que $k \overset{*}{\vdash} k'$ si es pot passar de la configuració k a la configuració k' en zero o més passos de transició directa. En altres paraules, sempre que existeixin k_0, k_1, ..., k_n configuracions intermèdies tals que

$$k = k_0 \vdash k_1 \vdash \cdots \vdash k_{n-1} \vdash k_n = k' .$$

Utilitzarem la notació $k \overset{n}{\vdash} k'$ per representar el pas de la configuració k a la configuració k' en exactament n transicions directes. Estem ja en condicions de definir el llenguatge reconegut per un autòmat amb pila determinista.

DEFINICIÓ. Donat un DPDA $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ qualsevol, s'anomena llenguatge *reconegut* per aquest autòmat, i es representa per $\mathsf{L}(M)$, el conjunt dels mots que permeten passar de la configuració inicial a una configuració amb estat acceptador després d'haver consumit tots els símbols del mot d'entrada. Formalment,

$$\mathsf{L}(M) = \{ w \in \Sigma^* \mid \exists \alpha \in \Gamma^* \; \exists q \in F \;\; Z_0 q_0 w \overset{*}{\vdash} \alpha q \} .$$

DEFINICIÓ. Anomenarem llenguatge incontextual determinista, abreujadament DCFL, qualsevol llenguatge reconegut per un DPDA. Representarem per DCFL la família dels DCFL.

Veurem més endavant, en aquest mateix capítol, la justificació d'aquesta denominació, és a dir, que els DCFL són tots ells CFL. A continuació presentem alguns exemples de DCFL i la seva caracterització via DPDA.

Exemple 8.2 L'autòmat amb pila donat a l'exemple 8.1 reconeix el llenguatge dels mots de $\{a, b\}^*$ que tenen el mateix nombre de símbols a i b. El seu funcionament és molt senzill: els símbols A i B de l'alfabet de pila s'associen, respectivament, als símbols a i b de l'entrada, i el contingut de la pila reflecteix, a cada pas de càlcul, la diferència de símbols a i b del mot d'entrada que han estat llegits fins a aquest moment. L'estat q_0 correspon als prefixos amb el mateix nombre de a's i b's i, per tant, és estat acceptador, mentre que l'estat no acceptador q_1 correspon als prefixos en què el nombre de a's i b's és diferent. La gestió de la pila en l'estat q_1 és la següent: si hi ha superàvit de a's (apareix un símbol A al cim de la pila), noves a's incrementen aquest superàvit (s'afegeixen més símbols A a la pila) i noves b's el decrementen (s'eliminen A's de la pila). En cas de superàvit de b's (el cim de la pila és un símbol B), el comportament és el simètric. Es retorna a q_0 cada vegada que el comptador de a's i b's s'iguala (apareix Z_0 al cim de la pila); per tant, el mot d'entrada acabarà la seva computació en q_0 –i serà acceptat– si i només si té tantes a's com b's .

Exemple 8.3 Considerem el llenguatge $\{wcw^{\mathrm{R}} \mid w \in (a+b)^*\}$ sobre l'alfabet $\{a, b, c\}$. La figura 8.5 representa un DPDA sense λ-transicions que el reconeix.

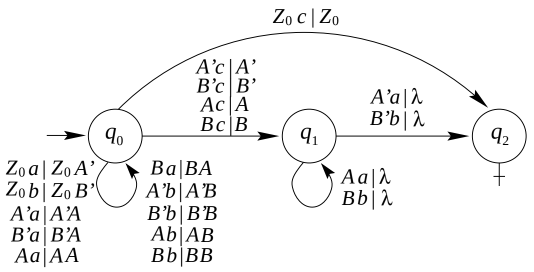

Fig. 8.5 DPDA *que reconeix el llenguatge* $\{wcw^{\mathrm{R}} \mid w \in (a+b)^*\}$

La funció de transició d'aquest mateix autòmat, expressada en forma de regles de reescriptura, es troba a la taula 8.1.

Taula 8.1 *Taula de transicions del* DPDA *de la figura 8.5*

Z_0q_0a	$\vdash$	$Z_0A'q_0$	$B'q_0b$	$\vdash$	$B'Bq_0$
Z_0q_0b	$\vdash$	$Z_0B'q_0$	$B'q_0c$	$\vdash$	$B'q_1$
Z_0q_0c	$\vdash$	Z_0q_2	Bq_0a	$\vdash$	BAq_0
$A'q_0a$	$\vdash$	$A'Aq_0$	Bq_0b	$\vdash$	BBq_0
$A'q_0b$	$\vdash$	$A'Bq_0$	Bq_0c	$\vdash$	Bq_1
$A'q_0c$	$\vdash$	$A'q_1$	$A'q_1a$	$\vdash$	q_2
Aq_0a	$\vdash$	AAq_0	Aq_1a	$\vdash$	q_1
Aq_0b	$\vdash$	ABq_0	$B'q_1b$	$\vdash$	q_2
Aq_0c	$\vdash$	Aq_1	Bq_1b	$\vdash$	q_1
$B'q_0a$	$\vdash$	$B'Aq_0$			

La idea bàsica del funcionament de l'autòmat és que emmagatzema a la pila el prefix w del mot de l'entrada i després de consumir el símbol c compara, símbol a símbol, el contingut de la pila amb el sufix w^{R} que queda per llegir. L'autòmat accepta quan, després de consumir tot el mot de l'entrada, a la pila només hi queda el símbol de fons Z_0. Per exemple, a continuació presentem la seqüència de configuracions que porten l'autòmat descrit a acceptar el mot $w = abacaba$. Observeu que hem subratllat els submots que es reescriuen a cada pas.

$$\underline{Z_0 q_0 a}bacaba \vdash Z_0\underline{A' q_0 b}acaba \vdash Z_0 A'\underline{B q_0 a}caba \vdash Z_0 A' B\underline{A q_0 c}aba$$
$$\vdash Z_0 A' B\underline{A q_1 a}ba \vdash Z_0 A'\underline{B q_1 b}a \vdash Z_0\underline{A' q_1 a} \vdash Z_0 q_2$$

i el mot és acceptat, ja que $q_2 \in F$.

És interessant assenyalar que el fet de marcar de manera especial el primer símbol que guardem a la pila (A' o B') possibilita la construcció de l'autòmat sense utilitzar λ-transicions, ja que permet anticipar en un pas el coneixement que no queden més símbols útils a la pila i passar a l'estat acceptador consumint justament el darrer símbol del mot d'entrada i no després amb una λ-transició. Amb la mateixa idea es podria construir un DPDA sense λ-transicions equivalent al de l'exemple 8.1.

A la vista de l'exemple precedent, ens podem preguntar si és sempre possible evitar les λ-transicions en els DPDA, ja que en tal cas aquestes serien supèrflues. La resposta és negativa. Podem constatar-ho a l'exemple següent.

Exemple 8.4 Sigui el llenguatge $\{a^n b^m 1 c^n \mid n, m \geqslant 1\} \cup \{a^n b^m 2 c^m \mid n, m \geqslant 1\}$ sobre l'alfabet $\Sigma = \{a, b, c, 1, 2\}$. L'autòmat de la figura 8.6 és un DPDA que el reconeix.

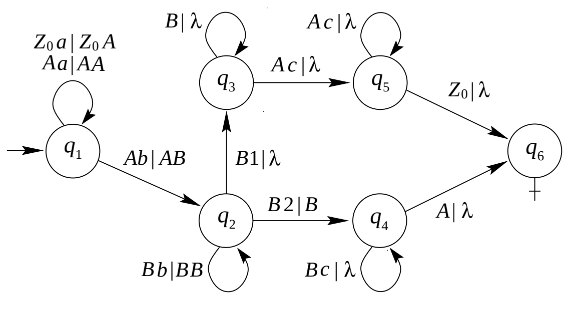

Fig. 8.6 DPDA *que reconeix* $\{a^n b^m 1 c^n \mid n, m \geqslant 1\} \cup \{a^n b^m 2 c^m \mid n, m \geqslant 1\}$

El DPDA presentat funciona bàsicament de la manera següent: els estats q_1 i q_2 s'utilitzen per memoritzar la quantitat de símbols a i b que hi ha al principi del mot d'entrada (guardant els símbols auxiliars corresponents A i B a la pila) i per forçar que l'ordre sigui correcte (primer les a's i després les b's). El símbol 1 o 2 que es llegeix a continuació determina un dels dos possibles camins a seguir sobre l'autòmat: un a través dels estats q_3, q_5 i q_6 (per als mots de la forma $a^n b^m 1 c^n$) i l'altre a través dels estats q_4 i q_6 (per als mots de la forma $a^n b^m 2 c^m$). En el primer cas, es desapilen amb λ-transicions els símbols B que han quedat damunt la pila i que han passat a ser inútils a l'efecte de comptatge (estat q_3) i després es comprova que hi ha tantes c's a l'entrada com símbols A a la pila (estat q_5). En el segon cas, el comptatge dels símbols c es pot fer directament, ja que cal *aparellar-los* amb els símbols B del cim de la pila (estat q_4). En qualsevol dels dos casos, es fa una λ-transició cap a l'estat acceptador quan el comptatge acaba tenint èxit.

Observeu que els parells d'estats q_1, q_2 i q_3, q_5 es poden agrupar en sengles estats únics. No ho hem fet per preservar la claredat i la connexió entre la idea del disseny de l'autòmat i el paper que juga cadascun dels estats. Observeu també que, en l'acceptació dels mots del tipus $a^n b^m 2c^m$, el DPDA construït no es preocupa de buidar els símbols A que han quedat al fons de la pila, producte del processament inicial del prefix a^n. Això no té cap importància, perquè el criteri d'acceptació que hem definit només té en compte l'estat d'acabament i no el contingut final de la pila.

Si mirem detingudament el llenguatge de l'exemple 8.4, ens convencerem que les λ-transicions són inevitables. La clau del problema rau en el fet que el símbol (1 o 2) que ens permet decidir si estem esperant un mot de la forma $a^n b^m 1c^n$ o bé $a^n b^m 2c^m$ no apareix fins després d'haver llegit el primer grup de a's i b's i, per tant, obliga a guardar en la pila els símbols B corresponents a les b's per si es tractés del segon cas (podeu comprovar que, en canvi, el llenguatge $\{a^n 1b^m c^n \mid n,m \geqslant 1\} \cup \{a^n 2b^m c^m \mid n,m \geqslant 1\}$ sí que pot ser reconegut per un DPDA sense λ-transicions). Per tant, el model de DPDA amb λ-transicions pot reconèixer un conjunt de llenguatges més gran que no pas el mateix model sense aquestes transicions buides. Aquesta característica i el fet que el model conserva totes les propietats que fan del DPDA una eina útil i eficient en el camp de la compilació en justifiquen la introducció.

L'existència de les λ-transicions té com a conseqüència que un DPDA pot entrar en bucle sense consumir símbols del mot d'entrada. Si el bucle es produeix abans d'haver llegit completament l'entrada, aquesta entrada no és acceptada, si ens atenim a la definició d'acceptació. Però un mot pot ser acceptat encara que l'autòmat no s'aturi. N'hi ha prou que, un cop llegit enterament, l'autòmat accedeixi a algun estat acceptador, ja sigui abans d'entrar en un bucle de λ-transicions, ja sigui a dins mateix del bucle. Observeu que aquesta situació és impossible en el model d'autòmat finit[3] , ja que un DFA efectua exactament tants passos de transició com símbols tingui el mot d'entrada fins a aturar-se en un estat d'acceptació o de rebuig. Per tant, el seu cost és sempre lineal respecte de la llargada de l'entrada.

Més endavant, en tractar les propietats dels DCFL, veurem que, afortunadament, aquests bucles de λ-transicions són detectables i evitables de manera algorísmica.

Un altre fet interessant d'observar a l'exemple 8.4 precedent és que el símbol 1 o 2 que apareix en els mots del llenguatge és el que converteix en determinista el llenguatge en qüestió. Es pot demostrar que, en canvi, no existeix cap DPDA que reconegui el llenguatge $\{a^n b^m c^n \mid n,m \geqslant 1\} \cup \{a^n b^m c^m \mid n,m \geqslant 1\}$. Informalment, això prové del fet que els mots d'aquest llenguatge no tenen cap característica que permeti distingir, en algun moment del càlcul, si el mot de l'entrada ha de ser del tipus $a^n b^m c^n$ o bé del tipus $a^n b^m c^m$. Això obliga a considerar indeterminísticament les dues possibilitats i a disposar, per tant, d'un model indeterminista de PDA.

[3] Ens referim al model d'autòmat finit determinista unidireccional. Més endavant, al capítol 9, veurem que els autòmats finits bidireccionals també poden entrar en bucle.

8.3 Autòmats amb pila indeterministes

El model indeterminista d'autòmat amb pila que presentarem en aquesta secció és un recurs conceptual que té l'interès teòric de ser equivalent al model de gramàtica incontextual i, per tant, de caracteritzar la família dels llenguatges incontextuals. Els DPDA són un cas particular d'aquests autòmats amb pila indeterministes, però no són models equivalents, ja que la família dels DCFL és un subconjunt estricte de la família dels CFL. En aquest sentit, veurem que un llenguatge clarament incontextual com ara $\{a^n b^m c^n \mid n, m \geqslant 1\} \cup \{a^n b^m c^m \mid n, m \geqslant 1\}$, que hem vist al final de la secció anterior, i que no és DCFL, sí que és reconegut per un autòmat amb pila indeterminista.

DEFINICIÓ. Un *autòmat amb pila indeterminista* (abreujadament NPDA, de l'anglès, *non-deterministic push-down automaton*) és un tuple de set components

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle ,$$

on Q, Σ, Γ, q_0, Z_0 i F tenen el mateix significat que en la definició dels DPDA, però ara la funció de transició δ és *qualsevol* subconjunt finit del producte cartesià

$$\Gamma Q(\Sigma \cup \{\lambda\}) \times \Gamma^* Q .$$

Es defineixen els conceptes de *configuració* i *transició entre configuracions* exactament igual que en el cas determinista. Observeu que l'única diferència respecte del model determinista és que ara una configuració pot anar seguida directament de diferents configuracions. Finalment, el llenguatge reconegut per un NPDA és també el conjunt de mots que permeten passar de la configuració inicial a una configuració amb estat acceptador després d'haver-se consumit tots els símbols. Formalment, donat un NPDA qualsevol $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$, s'anomena llenguatge *reconegut* per aquest autòmat, i es representa per $\mathsf{L}(M)$, el conjunt

$$\mathsf{L}(M) = \{w \in \Sigma^* \mid \exists \alpha \in \Gamma^* \, \exists q \in F \;\; Z_0 q_0 w \overset{*}{\vdash} \alpha q\} .$$

La idea d'indeterminisme presentada és la mateixa que ja hem vist al capítol 4 d'autòmats finits. Intuïtivament, un PDA indeterminista accepta un mot w quan existeix un *camí acceptador*, és a dir, una seqüència de tries que condueixen, al llarg del procés del mot en qüestió, des de la configuració inicial a una configuració acceptadora. L'autòmat rebutja el mot w quan cap de les tries possibles, al llarg del procés de w, no porta a una configuració acceptadora. Vegem-ne alguns exemples.

Exemple 8.5 Considerem el llenguatge dels mots palíndroms de longitud parella, $\{ww^{\mathsf{R}} \mid w \in (a+b)^*\}$. La figura 8.7 és un NPDA que el reconeix.

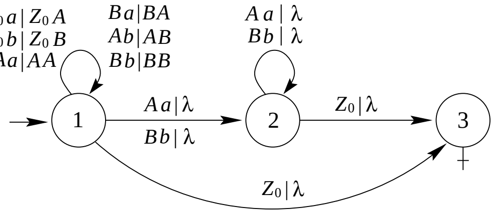

Fig. 8.7 NPDA *que reconeix* $\{ww^{\mathsf{R}} \mid w \in (a+b)^*\}$

Observeu que en aquest autòmat el caràcter indeterminista de la funció de transició ve donat pels parells de transicions següents:

$$\underbrace{A1a \vdash AA1,\ A1a \vdash 2}\ \text{ i }\ \underbrace{B1b \vdash BB1,\ B1b \vdash 2}.$$

El seu funcionament és molt semblant al de l'autòmat de la figura 8.5. Per acceptar un mot, cal llegir i apilar w símbol a símbol i, un cop arribats a la meitat del mot, llegir w^{R} comparant-lo amb el que tenim a la pila, mentre es va desapilant. El punt clau aquí és com es determina aquesta meitat que marca el canvi de comportament de l'autòmat. L'indeterminisme es pot veure com un *ajut extern* que indica en quin moment l'autòmat ha de començar a desapilar. Aquest fet és impossible de decidir de manera determinista sense dotar l'autòmat d'altres recursos, com ara la possibilitat de fer recular el capçal, o la de permetre-li posar marques als símbols de l'entrada.

Exemple 8.6 Sigui el llenguatge $\{a^n b^m c^n \mid n, m \geqslant 1\} \cup \{a^n b^m c^m \mid n, m \geqslant 1\}$, esmentat anteriorment. A la figura 8.8 es representa un NPDA que el reconeix.

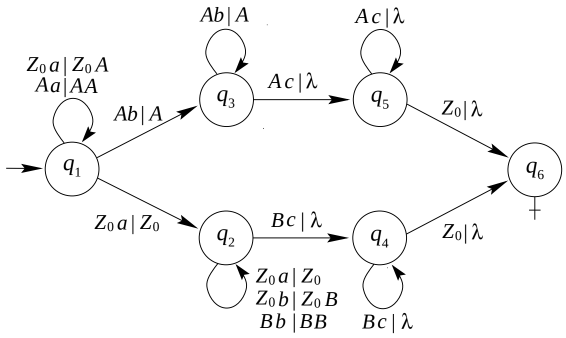

Fig. 8.8 NPDA *que reconeix* $\{a^n b^m c^n \mid n, m \geqslant 1\} \cup \{a^n b^m c^m \mid n, m \geqslant 1\}$

El graf d'estats descriu bàsicament dos camins possibles per arribar a l'estat acceptador: un a través dels estats q_1, q_3, q_5 i q_6, per acceptar els mots de la forma $a^n b^m c^n$, i l'altre seguint els estats q_1, q_2, q_4 i q_6, per acceptar els mots de la forma $a^n b^m c^m$. La tria indeterminista entre aquests dos camins es fa en l'estat q_1 amb el primer símbol del mot d'entrada. En qualsevol dels dos casos, la seqüència d'estats i el tipus de transicions garanteixen l'ordre dels símbols i el seu nombre mínim, és a dir, mots de $a^+b^+c^+$. Finalment, la pila s'utilitza per comptar el mateix nombre de a's i c's en el primer camí i el mateix nombre de b's i c's en el segon.

Remarquem que, a diferència dels DPDA, els NPDA poden prescindir de les λ-transicions sense perdre potència de càlcul. La demostració d'això, que deixem per al lector, no és difícil.

Exemple 8.7 L'autòmat de la figura 8.9 és un NPDA sense λ-transicions que reconeix el llenguatge $\{a^n b^m 1 c^n \mid n, m \geqslant 1\} \cup \{a^n b^m 2 c^m \mid n, m \geqslant 1\}$, presentat a l'exemple 8.4 com a prototipus de DCFL que no pot ser reconegut per cap DPDA sense λ-transicions. Podeu comparar aquest autòmat amb el DPDA que apareix a la figura 8.6.

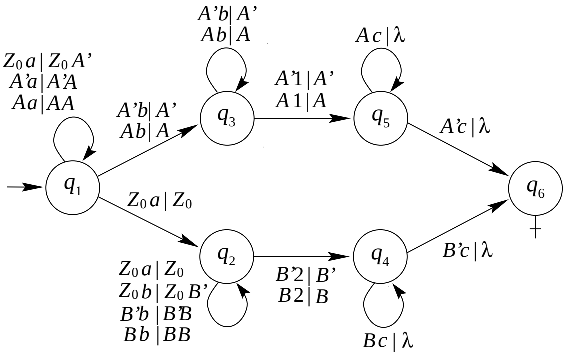

Fig. 8.9 NPDA *sense λ-transicions*

8.4 Equivalència entre autòmats amb pila i gramàtiques incontextuals

En aquesta secció veurem que els autòmats amb pila indeterministes i les gramàtiques incontextuals caracteritzen la mateixa família de llenguatges. Una conseqüència d'aquest resultat és que, atès que tot DPDA és un cas particular de NPDA, els DCFL són efectivament CFL (com el seu nom suggeria). Farem la demostració en dos passos: en primer lloc, demostrarem que a tota CFG li correspon un NPDA que reconeix el llenguatge generat per ella, i després provarem la correspondència inversa. En ambdós casos, explicitarem les construccions involucrades.

Construcció d'un NPDA a partir d'una CFG

Informalment, el NPDA resultant d'aquesta construcció concentra tot el procés en un únic estat i utilitza la pila per fer un recorregut, en preordre (*top-down*), d'un arbre de derivació seguint les produccions de la gramàtica. L'indeterminisme del PDA serà essencial per poder optar entre les diferents produccions de la gramàtica que són aplicables a cada pas. Si algun dels camins possibles porta l'autòmat a reproduir en la pila el recorregut d'un arbre de derivació del mot d'entrada, aleshores el mot és acceptat, altrament és rebutjat. L'alfabet de pila ha de contenir, per tant, variables i símbols terminals de la gramàtica. Precisament, el tipus de símbol que apareix al cim de la pila determina, a cada pas del càlcul, el comportament que ha de seguir l'autòmat: 1) Si el cim de la pila és una variable X, l'autòmat ha de tenir un camí possible per cada producció de la gramàtica del tipus $X \to \alpha$. Escollir-ne un significa fer una λ-transició que substitueixi, a la pila, la variable X per la part dreta de la producció, α. 2) Si el cim de la pila és un símbol terminal a, l'autòmat ha de verificar que sigui igual al símbol en curs del mot d'entrada i avançar el capçal de lectura cap al símbol següent. Calen, doncs, dos tipus de transicions: unes per reflectir aquesta tria entre les diferents produccions aplicables (indeterminisme) i unes altres per comprovar que el mot que es va construint a la pila coincideix exactament amb el de l'entrada. Finalment, la condició d'acceptació és que l'autòmat hagi consumit tots els símbols de l'entrada i a la pila hi resti únicament el símbol Z_0, és a dir, que algun dels camins possibles hagi permès reproduir, a la pila, un arbre de derivació del mot d'entrada. Formalitzem aquestes idees.

Donada una gramàtica incontextual $G = \langle V, \Sigma, P, S\rangle$ qualsevol, definim el NPDA $M_G = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \{\mathsf{f}\}\rangle$, on

- $Q = \{q_0, \mathsf{p}, \mathsf{f}\}$. El conjunt d'estats està format per un estat inicial, un estat acceptador i un estat que anomenarem *de procés*.
- $\Gamma = \Sigma \cup V \cup \{Z_0\}$ on $Z_0 \notin V \cup \Sigma$. L'alfabet de pila conté les variables, els terminals i el símbol Z_0.
- El conjunt de transicions està format per:
  1. $Z_0 q_0 \vdash Z_0 S\mathsf{p}$. Una λ-transició inicial que permet començar el procés, a base de posar la variable inicial de la gramàtica al cim de la pila i evolucionar cap a l'estat p.
  2. $\{Z\mathsf{p} \vdash \alpha^{\mathsf{R}}\mathsf{p} \mid Z \to \alpha \in P\}$. Amb la variable Z al cim de la pila hi ha d'haver una λ-transició per a cada producció de la gramàtica associada a la mateixa variable. Observeu que la convenció que hem adoptat d'escriure la pila de baix cap a dalt obliga a posar α^{R} a la part dreta de la transició.
  3. $\{a\mathsf{p}a \vdash \mathsf{p} \mid a \in \Sigma\}$. Les transicions que comproven que els terminals de l'arbre es corresponen amb els símbols del mot d'entrada.
  4. $Z_0\mathsf{p} \vdash \mathsf{f}$. Una λ-transició que permet passar a l'estat acceptador f quan tot el mot ha estat processat.

PROPOSICIÓ. $\mathsf{L}(M_G) = \mathsf{L}(G)$.

DEMOSTRACIÓ. La demostració d'aquesta proposició es troba a l'annex A, al final del capítol. $\square$

Exemple 8.8 Considerem el llenguatge $L = \{w \in (a+b)^* \mid |w|_a = |w|_b\}$. Les produccions següents descriuen una gramàtica incontextual G que genera L

$$S \to aSbS \mid bSaS \mid \lambda\,.$$

Sigui el mot $w = abbaba \in L$. L'arbre de la figura 8.10 és un possible arbre de derivació de w segons la gramàtica G.

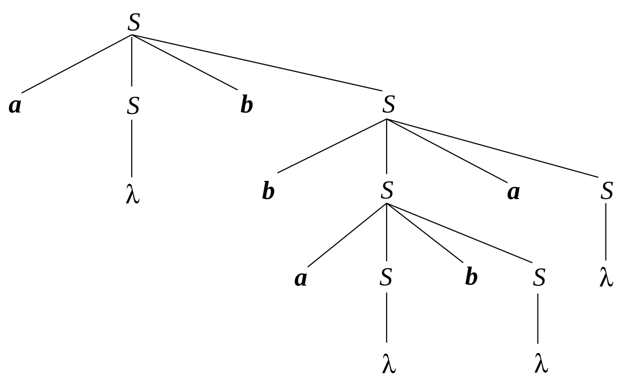

Fig. 8.10 *Un arbre de derivació per al mot abbaba*

L'autòmat M_G, construït segons la definició precedent, contindrà les transicions següents.

1. La transició inicial: $Z_0 q_0 \vdash Z_0 S\mathsf{p}$.
2. Les transicions associades a cada producció:
$$S\mathsf{p} \vdash SbSa\mathsf{p}\ ,\ \ S\mathsf{p} \vdash SaSb\mathsf{p}\ \ \text{i}\ \ S\mathsf{p} \vdash \mathsf{p}.$$
3. Les transicions de comprovació dels símbols de l'entrada amb els terminals de la pila: $a\mathsf{p}a \vdash \mathsf{p}$ i $b\mathsf{p}b \vdash \mathsf{p}$.
4. La transició de pas a l'estat acceptador: $Z_0\mathsf{p} \vdash \mathsf{f}$.

Aquest NPDA que hem construït reconeix L. Com a exemple, donem a continuació una seqüència de configuracions que prova que M_G accepta el mot w.[4] Observeu que els símbols que van apareixent al cim de la pila a cada pas del procés són els que corresponen al recorregut en preordre de l'arbre de derivació de la figura 8.10.

$$\begin{array}{rlll}
Z_0 q_0\, abbaba & \vdash\ Z_0 S\mathsf{p}abbaba & \vdash & Z_0 SbSa\mathsf{p}abbaba \\
& \vdash\ Z_0 SbS\mathsf{p}bbaba & \vdash & Z_0 Sb\mathsf{p}bbaba \\
& \vdash\ Z_0 S\mathsf{p}baba & \vdash & Z_0 SaSb\mathsf{p}baba \\
& \vdash\ Z_0 SaS\mathsf{p}aba & \vdash & Z_0 SaSbSa\mathsf{p}aba \\
& \vdash\ Z_0 SaSbS\mathsf{p}ba & \vdash & Z_0 SaSb\mathsf{p}ba \\
& \vdash\ Z_0 SaS\mathsf{p}a & \vdash & Z_0 Sa\mathsf{p}a \\
& \vdash\ Z_0 S\mathsf{p} & \vdash & Z_0\mathsf{p} \\
& \vdash\ \mathsf{f} & &
\end{array}$$

i el mot és acceptat, ja que f és un estat acceptador.

Remarquem, una vegada més, que la construcció del NPDA a partir de la CFG ens dóna com a resultat un autòmat amb tres estats, dels quals l'inicial i l'acceptador tenen una funció rutinària, mentre que tot el procés específic es realitza sobre un únic estat. Això és tant com dir que no és mitjançant els canvis d'estat que l'autòmat fa la seva funció, sinó a través del procés de modificar la pila (en efecte, cada modificació d'una variable al cim de la pila correspon a l'aplicació d'una producció), la qual cosa representa una traducció exacta del procés de derivació de la gramàtica. Així, el diagrama de transicions de l'autòmat es redueix a tornar a escriure la gramàtica en una forma diferent però anàloga.

Aquest fet posa clarament de manifest la pèrdua del caràcter *automàtic* dels PDA indeterministes. Conceptualment, un NPDA és molt més proper a una gramàtica incontextual que no pas a un model reconeixedor. Els únics autòmats "veritables" són, des d'aquest punt de vista, els DPDA, els quals permeten tractar l'acceptació de mots de manera eficient.

Construcció d'una CFG a partir d'un NPDA

Amb vista a facilitar la construcció en aquest sentit, és convenient introduir un nou model d'autòmat amb pila, que anomenarem NPDA d'*acceptació per pila buida*. Es tracta

[4] Convé assenyalar que el que proposem no és l'únic camí acceptador de w sobre M_G. Cada arbre de derivació de w segons la gramàtica dóna lloc a un camí acceptador diferent.

bàsicament d'un NPDA usual (que en aquest context anomenarem NPDA d'*acceptació per estat final*) en què els estats acceptadors són irrellevants i l'acceptació dels mots es produeix per buidat de la pila.

Formalment, es defineix com una estructura $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \varnothing \rangle$, on Q, Σ, Γ, δ, q_0 i Z_0 tenen el mateix significat que en la definició dels NPDA d'acceptació per estat final. El llenguatge reconegut per aquest autòmat M és:

$$\mathsf{L}(M) = \{w \in \Sigma^* \mid \exists q \in Q \;\; Z_0 q_0 w \overset{*}{\vdash} q\}.$$

La figura 8.11 conté un NPDA d'acceptació per pila buida, equivalent al de la figura 8.3, que reconeix el llenguatge dels mots sobre $\{a, b\}$ que tenen tantes a's com b's.

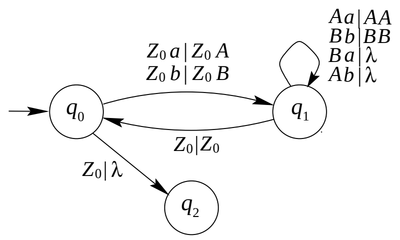

Fig. 8.11 *Un* NPDA *d'acceptació per pila buida*

És senzill demostrar l'equivalència dels dos models d'autòmat amb pila presentats. Donarem aquí només la demostració d'una de les implicacions. L'altra implicació és considerada a l'exercici 8.5.

PROPOSICIÓ. *Per a tot llenguatge L, si L és reconegut per un* NPDA *d'acceptació per estat final, llavors també és reconegut per un* NPDA *d'acceptació per pila buida.*

DEMOSTRACIÓ. Sigui $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ un NPDA d'acceptació per estat final que reconeix L. Es tracta senzillament d'incorporar un nou estat q_{f}, accessible des dels estats acceptadors, que s'encarregui de buidar la pila. Addicionalment, cal protegir el fons de la pila amb un nou símbol X_0 per sota de Z_0, per acceptar correctament aquells mots que porten M a buidar la pila en un estat acceptador (serà necessari afegir un nou estat inicial, q_{i}, per inicialitzar adequadament la pila).

Formalment, la construcció del NPDA d'acceptació per pila buida equivalent és la següent: $M' = \langle Q', \Sigma, \Gamma', \delta', q_{\mathrm{i}}, Z_0, \varnothing \rangle$, on $Q' = Q \cup \{q_{\mathrm{i}}, q_{\mathrm{f}}\}$, $\Gamma' = \Gamma \cup \{X_0\}$ i

$$\delta' = \delta \cup \{Z_0 q_{\mathrm{i}} \vdash X_0 Z_0 q_0\} \cup \{X q \vdash q_{\mathrm{f}} \mid q \in F \wedge X \in \Gamma'\} \cup \{X q_{\mathrm{f}} \vdash q_{\mathrm{f}} \mid X \in \Gamma'\}.$$

És clar que els dos autòmats reconeixen el mateix llenguatge. □

Ja estem en condicions de presentar el procés de construcció, a partir d'un NPDA M d'acceptació per pila buida, d'una gramàtica G_M que generi el mateix llenguatge que reconeix M. Sigui $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \varnothing \rangle$ un NPDA d'acceptació per pila buida. Construïm la gramàtica $G_M = \langle V, \Sigma, P, S \rangle$, on S és un símbol nou, de la manera següent:

- $V = \{S\} \cup Q \times \Gamma \times Q$. Representarem les variables del tipus $(p, A, q) \in Q \times \Gamma \times Q$ en la forma $[pAq]$, i el seu significat serà el següent: la gramàtica G_M podrà generar un mot w a partir d'una variable $[pAq]$ si i només si l'autòmat M, amb contingut de pila A, passa de l'estat p al q consumint w i buidant la pila.

- El conjunt de produccions està format per:
  1. $S \to [q_0 Z_0 q]$ per a cada $q \in Q$.
  2. $[pAq] \to a$ per a cada $(Apa \vdash q) \in \delta$ amb $a \in \Sigma \cup \{\lambda\}$.
  3. $[pAq_{i_m}] \to a[qB_1q_{i_1}][q_{i_1}B_2q_{i_2}] \cdots [q_{i_{m-1}}B_m q_{i_m}]$
     per a cada $(Apa \vdash B_m \cdots B_1 q) \in \delta$ i cada $(q_{i_1}, \ldots, q_{i_m}) \in Q^m$.

PROPOSICIÓ. $\mathsf{L}(G_M) = \mathsf{L}(M)$.

DEMOSTRACIÓ. La demostració d'aquesta proposició es troba a l'annex B, al final del capítol. □

La construcció donada és general per a NPDA. Com que els DPDA en són casos particulars, la manera de construir la gramàtica equivalent segueix exactament els mateixos passos. De tota manera, aquestes contruccions donen lloc, en general, a un nombre de produccions de la gramàtica que és exponencial amb respecte la talla del PDA de partida. És fàcil veure[5] que podem fer passar el cost d'aquesta construcció d'exponencial a cúbic si prenem la precaució de transformar prèviament les transicions que afegeixen un nombre arbitrari de símbols a la pila en seqüències de transicions que substitueixen el símbolo de cim de pila per dos símbols com a màxim. Vegeu l'exercici 8.14.

Exemple 8.9 Tornem a considerar el llenguatge L dels mots que tenen tantes a's com b's. Es tracta de construir una gramàtica incontextual per a L, a partir del NPDA d'acceptació per pila buida de la figura 8.11, el qual anomenem M.

La gramàtica G_M estarà descrita per les produccions següents.

1. Les produccions lligades a la variable inicial:
$$S \to [q_0 Z_0 q_0] \mid [q_0 Z_0 q_1] \mid [q_0 Z_0 q_2].$$
2. Les produccions derivades de les transicions que esborren símbols de la pila, és a dir, $Aq_1 b \vdash q_1$, $Bq_1 a \vdash q_1$ i $Z_0 q_0 \vdash q_2$:
$$[q_1 A q_1] \to b, \quad [q_1 B q_1] \to a \quad \text{i} \quad [q_0 Z_0 q_2] \to \lambda\,.$$
3. Les produccions derivades de les transicions que no fan decréixer el nombre de símbols de la pila. Per exemple, la transició $Z_0 q_0 a \vdash Z_0 A q_1$ origina totes les produccions del tipus
$$[q_0 Z_0 q'] \to a[q_1 A q''][q'' Z_0 q']\,,$$
on q' i q'' són estats qualssevol de l'autòmat (q_0, q_1, o q_2).És a dir,
$$\begin{aligned}
&[q_0 Z_0 q_0] \to a[q_1 A q_0][q_0 Z_0 q_0] \mid a[q_1 A q_1][q_1 Z_0 q_0] \mid a[q_1 A q_2][q_2 Z_0 q_0]\\
&[q_0 Z_0 q_1] \to a[q_1 A q_0][q_0 Z_0 q_1] \mid a[q_1 A q_1][q_1 Z_0 q_1] \mid a[q_1 A q_2][q_2 Z_0 q_1]\\
&[q_0 Z_0 q_2] \to a[q_1 A q_0][q_0 Z_0 q_2] \mid a[q_1 A q_1][q_1 Z_0 q_2] \mid a[q_1 A q_2][q_2 Z_0 q_2]\,.
\end{aligned}$$
Observeu que el seguiment estricte del procés de construcció comporta una explosió exponencial del nombre de produccions respecte del nombre d'estats de l'autòmat. De tota manera, moltes d'aquestes produccions generades no seran necessàries, ja que contindran símbols inútils. En el cas present, raonaments molt simples sobre el NPDA M ens permeten reduir dràsticament el conjunt de produccions a considerar.

[5] Vegeu [HMU01], pàgs. 294 i 295.

Per exemple, s'observa que M només utilitza Z_0 en el fons de la pila i que només pot buidar la pila amb la transició cap a l'estat q_2 (ja que és l'única que esborra el símbol Z_0). Per tant, podem deixar de considerar les variables del tipus $[pZ_0q]$ amb q diferent de q_2, en particular $[q_0Z_0q_0]$ i $[q_0Z_0q_1]$. Observeu també que, del fet que l'estat q_2 no tingui cap transició definida, se'n desprèn que totes les variables del tipus $[q_2Xq]$ seran inútils en la gramàtica. Amb aquestes simplificacions, les nou produccions presentades anteriorment es redueixen a:

$$[q_0Z_0q_2] \to a[q_1Aq_0][q_0Z_0q_2] \mid a[q_1Aq_1][q_1Z_0q_2]\,.$$

Els criteris de simplificació ja exposats i d'altres de semblants ens permeten reduir la resta de produccions al conjunt següent.

- $Z_0q_0b \vdash Z_0Bq_1$ origina:
  $[q_0Z_0q_2] \to b[q_1Bq_0][q_0Z_0q_2] \mid b[q_1Bq_1][q_1Z_0q_2]\,.$
- $Aq_1a \vdash AAq_1$ origina:
  $[q_1Aq_0] \to a[q_1Aq_1][q_1Aq_0]\,,\quad [q_1Aq_1] \to a[q_1Aq_1][q_1Aq_1]$ i
  $[q_1Aq_2] \to a[q_1Aq_1][q_1Aq_2]\,.$
- $Bq_1b \vdash BBq_1$ origina:
  $[q_1Bq_0] \to a[q_1Bq_1][q_1Bq_0]\,,\quad [q_1Bq_1] \to a[q_1Bq_1][q_1Bq_1]$ i
  $[q_1Bq_2] \to a[q_1Bq_1][q_1Bq_2]\,.$
- Finalment, $q_1Z_0 \vdash Z_0q_0$ origina: $[q_1Z_0q_2] \to [q_0Z_0q_2]\,.$

Observeu també que de les produccions associades a la variable S només té sentit $S \to [q_0Z_0q_2]$. Identificarem aquestes dues variables en una de sola i rebatejarem la resta de variables amb noms més senzills:

$$S = [q_0Z_0q_2]\,,\ \ A = [q_1Z_0q_2]\,,\ \ B = [q_1Aq_0]\,,\ \ C = [q_1Aq_1]\,,$$
$$D = [q_1Aq_2]\,,\ \ E = [q_1Bq_0]\,,\ \ F = [q_1Bq_1]\ \text{ i }\ G = [q_1Bq_2].$$

Escrivim la gramàtica resultant:

$$\begin{aligned}
&S \to \lambda \mid aBS \mid aCA \mid bES \mid bFA\\
&A \to S\\
&B \to aCB\\
&C \to b \mid aCC\\
&D \to aCD\\
&E \to bFE\\
&F \to a \mid bFF\\
&G \to bFG\,.
\end{aligned}$$

Observeu que les variables B, D, E i G són símbols no fecunds, que les variables D i G són no accessibles i que la variable A és la "mateixa" que la S. L'eliminació d'aquests símbols no útils i redundants dóna com a resultat la gramàtica següent amb només tres variables i set produccions:

$$\begin{aligned}
&S \to \lambda \mid aCS \mid bFS\\
&C \to b \mid aCC\\
&F \to a \mid bFF.
\end{aligned}$$

La gramàtica construïda a l'exemple precedent és equivalent a la de l'8.8, però a més aquesta és inambigua. Aquest fet no s'ha donat per casualitat. Fixeu-vos que el NPDA d'acceptació per pila buida original, tot i ser indeterminista, té un únic camí acceptador per a cada mot que reconeix, i que aquest fet es tradueix en un únic arbre de derivació possible en la gramàtica. La unicitat del camí acceptador és immediata quan el PDA de partida és determinista i es fa la construcció canònica per a passar a un NPDA d'acceptació per pila buida. Aquest resultat ens proveeix d'un mecanisme per construir gramàtiques inambigues per als DCFL (sempre que coneguem els DPDA corresponents). Addicionalment, ens permet afirmar que tots els DCFL són inambigus.

A la vista de les dues proposicions presentades en aquesta secció, podem enunciar el teorema següent.

TEOREMA. *Un llenguatge és incontextual si i només si és el llenguatge reconegut per algun autòmat amb pila.*

En conseqüència, a partir d'ara podem estudiar els CFL a través dels autòmats amb pila que els reconeixen. Això serà útil perquè sovint els PDA ajuden a comprendre fàcilment si un llenguatge donat és o no CFL, per la via de concentrar-se en l'argument principal —quines tasques es poden resoldre amb una pila?— i prescindir dels secundaris.

Considereu per exemple, el llenguatge $L = \{w \in \{a\,,b\}^* \mid |w|_a = 2|w|_b\}$. Justificar l'existència d'un PDA que el reconegui és prou senzill. Conceptualment és el mateix DPDA que el de l'exemple 8.1, el qual acceptava els mots amb el mateix nombre de a's i b's. En aquell cas, la pila es feia servir per comptar la diferència entre símbols a i b que es portaven llegits a cada moment i el PDA acceptava només quan el contingut de la pila en acabar el processament del mot d'entrada era exactament Z_0 (diferència igual a zero). Ara volem que la pila reflecteixi la diferència entre el nombre de a's i el doble del nombre de b's. Per fer això, només cal que els símbols b *comptin doble* en el recompte a la pila. Més concretament, a l'inici el PDA carrega a la pila una A o dues B's segons que el primer símbol sigui a o b, respectivament. A partir d'aquest moment, si el PDA llegeix una nova b i la diferència és favorable a les b's (al cim de la pila hi ha una B), es carreguen dos símbols B més a la pila, mentre que si la diferència és de signe contrari (al cim de la pila hi ha una A), el que cal fer és eliminar dues A's de la pila (això s'haurà de fer en dos passos, utilitzant un estat addicional i una λ-transició). En el cas que només quedi una A a la pila i el símbol llegit sigui una b, l'autòmat haurà d'esborrar aquesta A i carregar una B a la pila (novament fent ús d'un estat intermedi). Finalment, el comportament davant de nous símbols a del mot d'entrada no ha variat respecte al DPDA referit anteriorment. Com en aquell cas, també el PDA haurà d'acceptar quan, després d'haver processat completament el mot d'entrada, a la pila només hi quedi el símbol Z_0.

Com que aquest PDA esbossat és determinista, tenim la seguretat que existeix una gramàtica inambigua per a L i, a més, la podem construir automàticament a partir de l'autòmat. Adoneu-vos que, en canvi, construir directament una CFG no ambigua per a L no és una tasca gens fàcil.

En el cas contrari podem veure, per exemple, que el llenguatge dels mots quadrats, $\{xx \mid x \in (a+b)^*\}$, no és incontextual. Un hipotètic PDA que acceptés L hauria de fer servir la pila per memoritzar els símbols de la primera meitat del mot, per poder comprovar després que coincideixen amb els de la segona. Decidir on està la meitat del mot no és un problema, ja que podem disposar d'indeterminisme, però el que serà impossible de fer és comparar, sense perdre informació, els símbols de la segona meitat amb els que han estat carregats prèviament a la pila. El motiu és que els símbols llegits d'esquerra a dreta es guarden de baix cap a dalt en la pila i, per tant, queden emmagatzemats en l'ordre invers al que seria útil: en el moment de llegir el primer símbol de la segona meitat, aquest s'hauria de comparar amb el que es troba a sota de tot en la pila, i aquest accés és impossible en una estructura d'aquest tipus.[6]

Evidentment, aquests darrers raonaments no constitueixen demostracions formals, sinó simples justificacions intuïtives del caràcter dels llenguatges considerats. Fer una demostració formal requeriria, en el primer cas, la construcció efectiva del PDA i la seva verificació, mentre que en el segon cas ens podria servir l'aplicació del lema de bombament per a llenguatges incontextuals que ja hem descrit al capítol 7.

8.5 Propietats de tancament dels CFL i dels DCFL

Intersecció d'un CFL i un llenguatge regular

Ja coneixem, del capítol 2, que la intersecció de dos CFL qualssevol no és necessàriament un CFL. Ara bé, si almenys un dels dos és regular, aleshores podem assegurar que el llenguatge intersecció és un llenguatge incontextual. Per demostrar això farem la construcció d'un PDA que reconeix el llenguatge intersecció a partir del PDA i del FA que reconeixen els llenguatges considerats.

Donats un PDA, $M = \langle Q_M, \Sigma, \Gamma, \delta_M, q_0, Z_0, F_M \rangle$, que reconeix un determinat llenguatge L, i un DFA, $A = \langle Q_A, \Sigma, \delta_A, p_0, F_A \rangle$, que reconeix un llenguatge R, es tracta de construir un PDA M' que reconegui el llenguatge intersecció $L \cap R$. Per fer-ho, ens basarem en la idea, utilitzada també al capítol 4, de simular el funcionament dels dos autòmats simultàniament. Estendrem la memòria fent el producte cartesià d'estats $Q_A \times Q_M$ i la funció de transició δ_A actuarà sobre el primer component, mentre que δ_M ho farà sobre el segon. La pila de M' reproduirà exactament la de M, l'estat inicial de M' serà el parell format pels estats inicials de cadascun dels autòmats i els estats acceptadors seran els parells del producte cartesià $F_A \times F_M$, ja que M' ha d'acceptar aquells mots que siguin acceptats per tots dos autòmats. Remarquem que, en el cas particular en què l'autòmat M realitzi una λ-transició, M' ha de simular la transició corresponent de M, actualitzant la pila i els possibles estats següents, sense variar l'estat actual de A. Aquest fet donarà lloc a dos tipus de transicions en M': les que consumeixen efectivament un nou símbol de l'entrada i reprodueixen el comportament dels dos autòmats i les λ-transicions, que reflecteixen el comportament de M i deixen A *momentàniament aturat*.

Formalment, l'autòmat construït és $M' = \langle Q, \Sigma, \Gamma, \delta, (p_0, q_0), Z_0, F \rangle$, els components del qual, encara no definits, són

- $Q = Q_A \times Q_M$,
- $F = F_A \times F_M$,

i δ conté les transicions

1. $\{Z(p,q)a \vdash \alpha(p',q') \mid a \in \Sigma \ \wedge \ \delta_A(p,a) = p' \ \wedge \ Zqa \vdash \alpha q' \in \delta_M\}$.

[6] Algunes modificacions en el model d'autòmat amb pila, com per exemple la que veurem al capítol 9, que consisteix a dotar-lo d'un capçal bidireccional (que pugui avançar i retrocedir en la cinta d'entrada), permetrien superar aquesta limitació.

2. $\{Z(p,q) \vdash \alpha(p,q') \mid Zq \vdash \alpha q' \in \delta_M\}$.

Exemple 8.10 Suposem que volem construir un PDA que accepti el llenguatge $L' = \{ww^{\mathsf{R}} \mid w \in (a+b)^* \wedge |w| = \dot{3}\}$. La definició d'aquest llenguatge combina una condició típicament incontextual (la condició de palíndrom) amb una condició típicament regular (longitud múltiple de tres). Podem considerar-lo, doncs, com la intersecció d'un llenguatge regular $R = \{w \in (a+b)^* \mid |w| = \dot{3}\}$ i un d'incontextual $L = \{ww^{\mathsf{R}} \mid w \in (a+b)^*\}$. La figura 8.12 representa un senzill DFA que reconeix R, mentre que anteriorment, a la figura 8.7, ha estat exposat un NPDA que reconeix L.

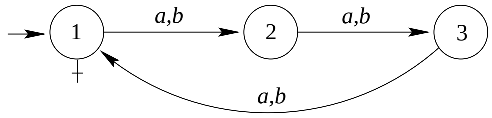

Fig. 8.12 DFA *que reconeix els mots de longitud múltiple de tres*

El NPDA que resulta del procés d'intersecció d'aquests dos darrers autòmats està dibuixat a la figura 8.13, on t_1 resumeix el conjunt de transicions

$$Z_0 a|Z_0 A\,,\ Z_0 b|Z_0 B\,,\ Aa|AA\,,\ Ba|BA\,,\ Ab|AB\,,\ Bb|BB\,,$$

mentre que t_2 resumeix les transicions $Aa|\lambda$ i $Bb|\lambda$, i el símbol t_3 significa $Z_0|Z_0$.

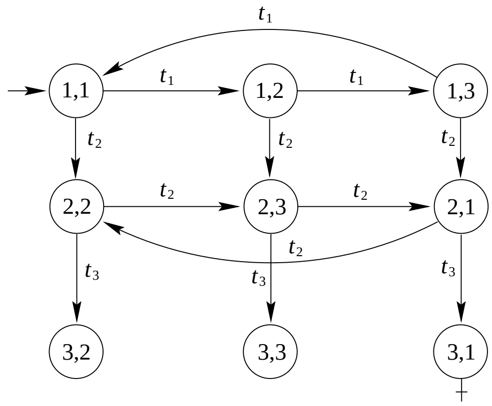

Fig. 8.13 NPDA *que reconeix el llenguatge* $\{ww^{\mathsf{R}} \mid w \in (a+b)^* \wedge |w| = \dot{3}\}$

PROPOSICIÓ. $\mathsf{L}(M') = R \cap L$.

DEMOSTRACIÓ. Es demostra fàcilment, per inducció sobre el nombre n de passos de procés, que en M' hi ha la transició $Z_0(p_0,q_0)w \overset{n}{\vdash} \alpha(p,q)$ si i només si en A es dóna $p_0 w = p$ i en M hi ha la transició $Z_0 q_0 w \overset{n}{\vdash} \alpha q$. Així doncs, per a tot mot $w \in \Sigma^*$

$$\begin{aligned} w \in \mathsf{L}(M') &\iff \exists \alpha \in \Gamma^* \ \exists (p,q) \in F_A \times F_M \quad Z_0(p_0,q_0)w \overset{*}{\underset{M'}{\vdash}} \alpha(p,q) \\ &\iff \exists p \in F_A \ p_0 w = p \ \wedge \ \exists \alpha \in \Gamma^* \ \exists q \in F_M \ Z_0 q_0 w \overset{*}{\underset{M}{\vdash}} \alpha q \\ &\iff w \in \mathsf{L}(A) \ \wedge \ w \in \mathsf{L}(M) \\ &\iff w \in L \cap R \end{aligned}$$

□

Com a conseqüència de la proposició anterior, podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de la intersecció amb llenguatges regulars.*

Observeu que si l'autòmat amb pila de partida és determinista, la construcció considerada manté aquest determinisme, i dóna com a resultat un DPDA. Per tant, també podem enunciar el corol·lari següent.

COROL·LARI. *La família dels DCFL és tancada respecte de la intersecció amb llenguatges regulars.*

Morfisme invers d'un CFL

Demostrarem que la família dels CFL és tancada respecte de l'aplicació de morfismes inversos. De la mateixa manera que passava amb els llenguatges regulars (capítol 4), els autòmats —en aquest cas amb pila— seran una eina adequada per fer-ho.

Sigui $h\colon \Sigma_1^* \longrightarrow \Sigma_2^*$ un morfisme qualsevol i $M_2 = \langle Q_2, \Sigma_2, \Gamma, \delta_2, q_0'', Z_0, F_2\rangle$ un NPDA que reconeix un cert llenguatge $L_2 \subseteq \Sigma_2^*$. L'objectiu és construir un NPDA M_1 que reconegui el llenguatge $h^{-1}(L_2)$. Essencialment, el comportament de M_1 davant d'un símbol $a \in \Sigma_1$ de l'entrada i d'un determinat contingut de la pila serà simular el comportament de M_2 davant del mot $h(a)$ i el mateix contingut de pila. D'aquesta manera M_1 acceptarà un mot w si i només si M_2 accepta $h(w)$, és a dir que $\mathsf{L}(M_1) = h^{-1}(\mathsf{L}(M_2)) = h^{-1}(L_2)$. Aquesta idea és la mateixa que utilitzàvem al capítol 4 per demostrar que l'aplicació de morfismes inversos també preserva el caràcter regular dels llenguatges, amb l'únic canvi que els autòmats involucrats eren FA i no PDA. Recordem que en aquell cas la construcció era molt senzilla perquè la simulació de M_2 sobre el mot $h(a)$ es pot resumir en un sol pas de càlcul de M_1. Ara la simulació és una mica més complicada. Fixeu-vos que el PDA M_1 pot desapilar un nombre no determinat de símbols de la pila durant el càlcul sobre el mot $h(a)$. Com que en una sola transició un PDA només pot modificar el símbol del cim de la pila, la simulació de M_1 sobre tota la cadena $h(a)$ pot requerir més d'un pas. Plantejarem la solució d'aquest problema simulant el comportament de M_1 símbol a símbol del mot $h(a)$ i utilitzant tants estats com faci falta.

Funcionalment, la solució emprada es pot interpretar com la introducció al control de M_1 d'una memòria intermèdia que permet emmagatzemar el mot $h(a)$ i que proporciona, un a un, els símbols d'aquest mot al control de l'autòmat M_2. A cada pas de la simulació del mot $h(a)$ en la memòria intermèdia, necessitem tenir constància de l'estat en què es trobaria l'autòmat M_2 i de la posició en què estaria el capçal de M_2. Podem interpretar aquesta simulació com si hi hagués un cursor virtual que va recorrent el mot $h(a)$ d'esquerra a dreta en la memòria intermèdia. L'esquema de la figura 8.14 il·lustra aquesta idea.

Com que només la posició del cursor i els símbols que hi ha a la seva dreta són rellevants, podem identificar el conjunt de situacions possibles amb el conjunt de sufixos dels mots $h(a)$ per a cada a de Σ_1, és a dir,

$$S = \{y \in \Sigma_2^* \mid \exists a \in \Sigma_1\ \exists x \in \Sigma_2^* \quad h(a) = xy\}\,.$$

Així doncs, cada canvi d'estat i de posició del capçal en l'autòmat M_2 pot ser representat per un parell de $Q_2 \times S$. De fet, com que Q_2 i S són disjunts, és més senzill identificar aquests parells amb mots de $Q_2 S\$$ (cadenes formades per un estat de Q_2 seguit d'un

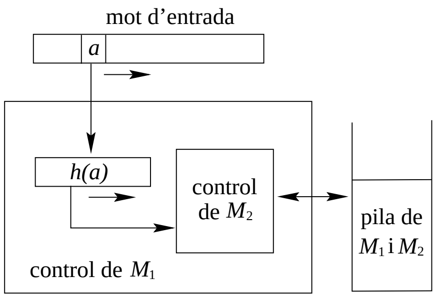

Fig. 8.14 *Construcció d'un* PDA *per al morfisme invers*

sufix de $h(a)$ per a alguna a, i d'un símbol separador \$ que no pertany ni a Σ_1 ni a Σ_2). Aquest conjunt, que és evidentment finit, serà el conjunt d'estats de l'autòmat M_1 i el representarem per Q_1. Així la informació del contingut de la memòria intermèdia queda representada en els estats de Q_1. Les configuracions de l'autòmat M_1 seran del tipus $\alpha qy\$w$, on $\alpha \in \Gamma^*$ és el contingut de la pila, $qy\$ \in Q_1$ és l'estat actual i $w \in \Sigma_1^*$ és el sufix del mot d'entrada que queda a la dreta del capçal. S'observa, per tant, que el símbol \$ fa només la funció de separador de l'estat actual (a l'esquerra) i del mot d'entrada (a la dreta). Això és necessari perquè els mots y i w es podrien confondre en el cas que els alfabets Σ_1 i Σ_2 no fossin disjunts. Observeu també que les situacions en què el cursor virtual hagi recorregut tota la memòria intermèdia vindran donades per $y = \lambda$, és a dir, per estats de $Q_2\$$. Amb tot això, el funcionament de M_1 serà el següent: començant en l'estat inicial $q_0' = q_0''\$$, carregarà a la memòria intermèdia la imatge del primer símbol del mot d'entrada. A partir d'aquest moment simularà el funcionament de M_2 sobre aquest mot imatge, i cada vegada que el cursor virtual hagi recorregut el contingut sencer de la memòria intermèdia llegirà un nou símbol de l'entrada i carregarà novament la memòria intermèdia amb la imatge corresponent. Finalment, M_1 acceptarà un mot quan la seva computació, després de consumir tots els símbols d'entrada, acabi en un estat $q\$$ amb $q \in F_2$.

Formalment, el PDA construït és

$$M_1 = \langle Q_1, \Sigma_1, \Gamma, \delta_1, q_0', Z_0, F_1 \rangle ,$$

on els components nous són

- $Q_1 = Q_2 S\$$,
- $q_0' = q_0''\$$,
- $F_1 = F_2\$$,

i δ_1 conté les transicions següents:

1. $\{Zp\$a \vdash_{\overline{M_1}} Zph(a)\$ \mid a \in \Sigma_1 \wedge Z \in \Gamma \wedge p \in Q_2\}$. Quan el cursor virtual arriba al final de la memòria intermèdia, es consumeix un nou símbol a de l'entrada i es carrega la memòria intermèdia amb $h(a)$.
2. $\{Zpy\$ \vdash_{\overline{M_1}} \alpha qy\$ \mid Zp \vdash_{\overline{M_2}} \alpha q \in \delta_2 \wedge y \in S\}$. Simulació de les λ-transicions de M_2. No es consumeix cap símbol de l'entrada ni tampoc de la memòria intermèdia.
3. $\{Zpby\$ \vdash_{\overline{M_1}} \alpha qy\$ \mid Zpb \vdash_{\overline{M_2}} \alpha q \in \delta_2 \wedge by \in S\}$. Simulació dels moviments de M_2 amb entrada b de la memòria intermèdia. El cursor virtual avança una posició. El capçal que llegeix l'entrada no es mou.

Exemple 8.11 Tornem a reprendre el llenguatge $L_2 = \{ww^{\mathrm{R}} \mid w \in (a+b)^*\}$ vist en exemples precedents i considerem el morfisme $h\colon \{c\,, d\}^* \longrightarrow \{a\,, b\}^*$, definit per $h(c)=ab$ i $h(d)=ba$. Ara volem caracteritzar, mitjançant un PDA, el conjunt $h^{-1}(L_2)$, és a dir, el llenguatge dels mots de $\{c\,, d\}^*$ tals que la seva imatge pel morfisme h és un palíndrom de longitud parella en $\{a\,, b\}^*$. Recordem que el NPDA de la figura 8.7, que aquí anomenarem M_2, reconeix el llenguatge L_2. Així doncs, no hem de fer res més que construir l'autòmat per al morfisme invers seguint les regles que hem donat en aquesta secció. En aquest cas, el conjunt de sufixos possibles és $S=\{\lambda, a, b, ab, ba\}$ i l'autòmat M_1 resultant, que reconeix el llenguatge $h^{-1}(L_2)$, està representat a la figura 8.15. Observeu que, per tal de simplificar la notació, hem utilitzat el símbol $*$ per representar qualsevol símbol de pila. Així, per exemple, la transició $*1\$c \vdash *1ab\$$ significa que, sigui quin sigui el símbol del cim de la pila, si el símbol en curs del mot d'entrada és una c i l'estat actual és $1\$$, l'autòmat passarà a l'estat $1ab\$$ i deixarà la pila tal com està.

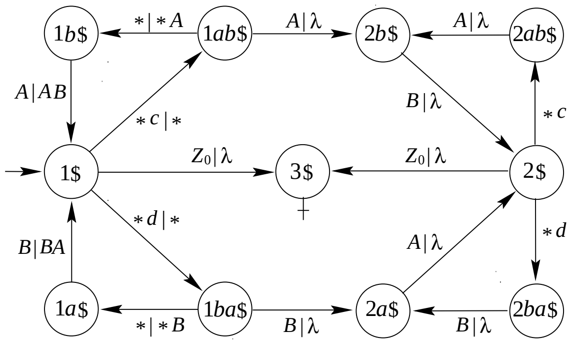

Fig. 8.15 NPDA M_1 *que reconeix el llenguatge* $h^{-1}(L_2)$

Observeu també que, per motius de claredat, en el diagrama de transicions precedent no hem inclòs tots els estats ni tampoc totes les transicions que surten d'aplicar estrictament la construcció general de l'autòmat M_1. En particular, no apareixen els estats $3a\$$, $3b\$$, $3ab\$$ i $3ba\$$, que són *morts* i, per tant, no útils, i pel que fa a les transicions, han estat eliminades totes aquelles que no es podrien aplicar mai, i que són:

$$Z_0 1a\$ \vdash Z_0 A1\$,\quad A1a\$ \vdash AA1\$,\quad A1a\$ \vdash 2\$,$$
$$Z_0 1b\$ \vdash Z_0 B1\$,\quad B1b\$ \vdash BB1\$,\quad B1b\$ \vdash 2\$.$$

Analitzem a continuació l'autòmat que hem construït. Essencialment, M_1 realitza dos tipus de bucles: dos sobre l'estat inicial i dos sobre l'estat $2\$$. Els dos primers —que involucren les ternes d'estats $1\$$, $1ab\$$, $1b\$$ i $1\$$, $1ba\$$, $1a\$$, respectivament— serveixen per carregar a la pila un mot AB per cada c de l'entrada i un mot BA per cada d. És evident que aquesta feina es pot resumir en un sol estat. Els altres dos —que involucren les ternes d'estats $2\$$, $2ab\$$, $2b\$$ i $2\$$, $2ba\$$, $2a\$$, respectivament— fan la feina inversa, és a dir, descarreguen de la pila mots BA o AB en processar, respectivament, símbols c i d del mot d'entrada. En aquest cas, també es pot sintetitzar aquest procés en un nombre menor d'estats, tot i que el fet d'haver d'eliminar dos símbols consecutius de la pila obligarà a tenir sengles estats intermedis.

A la figura 8.16 hi trobem descrit un NPDA equivalent a M_1 que incorpora les simplificacions que acabem d'esmentar. L'estat 1 fa la feina dels estats $1\$$, $1ab\$$, $1b\$$, $1ba\$$ i $1a\$$; els estats 2, 3 i 4 resumeixen la feina de $2\$$, $2ab\$$, $2b\$$, $2ba\$$ i $2a\$$; i finalment l'estat 5 equival al $3\$$.

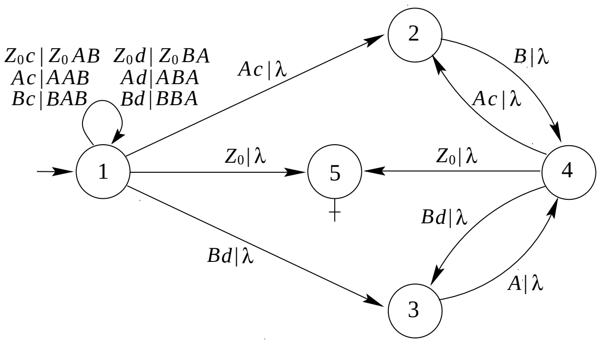

Fig. 8.16 NPDA *equivalent al de la figura 8.15*

És clar que aquest autòmat codifica a la pila cada c per AB i cada d per BA. Si prenem els mots de pila AB i BA com a símbols atòmics C i D, l'autòmat precedent es pot simplificar encara més i donar lloc al PDA de la figura 8.17.

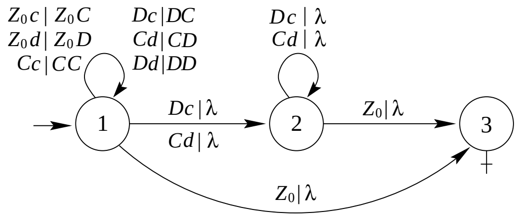

Fig. 8.17 NPDA *equivalent al de la figura 8.16*

No és difícil de veure que el llenguatge $h^{-1}(L_2)$ és

$$L_1 = \{w \mid \exists x \in \{c,d\}^* \ w = x\psi(x^{\mathsf{R}})\},$$

on $\psi(w)$ representa el mot obtingut canviant les c's de w per d's, i viceversa. És a dir, que L_1 és un llenguatge molt proper al dels palíndroms de longitud parella: si el símbol a distància n del principi és una c, aleshores el símbol a la mateixa distància començant pel final ha de ser una d, i viceversa, si el símbol n-èsim és una d, aleshores el símbol corresponent començant pel final ha de ser una c. Observeu que el NPDA de la figura 8.17 és exactament el que hauríem obtingut si haguéssim construït l'autòmat directament a partir del llenguatge L_1 (fixeu-vos que la idea per construir-lo és exactament la mateixa que a l'exemple 8.5 ens portava al PDA representat a la figura 8.7).

PROPOSICIÓ. $\mathsf{L}(M_1) = h^{-1}(L_2)$.

DEMOSTRACIÓ. Es demostra fàcilment que per a tot $\alpha, \beta \in \Gamma^*$, $p, q \in Q$ i $a \in \Sigma_1$ es té

$$\alpha p h(a) \vdash^*_{M_2} \beta q \iff \alpha p\$ a \vdash_{M_1} \alpha p h(a)\$ \vdash^*_{M_1} \beta q\$. \qquad (1)$$

En efecte, l'autòmat M_1 no fa altra cosa que carregar a la memòria intermèdia el mot $h(a)$ (transició del tipus 1 de la definició) i després simular el comportament de M_2 amb aquesta entrada (transicions dels tipus 2 i 3 de la definició).

Considerem ara un mot qualsevol $w = a_1 a_2 \cdots a_n \in \Sigma_1^*$ pertanyent al llenguatge $\mathsf{L}(M_1)$. Segur que existeixen $\alpha \in \Gamma^*$ i $q \in F_1$ tals que

$$Z_0 q_0' w \overset{*}{\underset{M_1}{\vdash}} \alpha q \,.$$

Tenint en compte que q_0' és $q_0''\$$, podem descompondre aquest càlcul en els passos següents:

$$\begin{aligned} Z_0 q_0'' \$ a_1 a_2 \cdots a_n &\underset{M_1}{\vdash} Z_0 q_0'' h(a_1) \$ a_2 \cdots a_n \overset{*}{\underset{M_1}{\vdash}} \alpha_1 q_1 \$ a_2 \cdots a_n \\ &\underset{M_1}{\vdash} \alpha_1 q_1 h(a_2) \$ a_3 \cdots a_n \overset{*}{\underset{M_1}{\vdash}} \alpha_2 q_2 \$ a_3 \cdots a_n \\ &\cdots \\ &\underset{M_1}{\vdash} \alpha_{n-1} q_{n-1} h(a_n) \$ \quad \overset{*}{\underset{M_1}{\vdash}} \alpha_n q_n \$ = \alpha q \,. \end{aligned}$$

En virtut de la fórmula 1, ha d'existir en M_2 el càlcul següent:

$$\begin{aligned} Z_0 q_0'' h(w) = Z_0 q_0'' h(a_1) h(a_2) \cdots h(a_n) &\overset{*}{\underset{M_2}{\vdash}} \alpha_1 q_1 h(a_2) \cdots h(a_n) \\ &\overset{*}{\underset{M_2}{\vdash}} \alpha_2 q_2 h(a_3) \cdots h(a_n) \\ &\cdots \\ &\overset{*}{\underset{M_2}{\vdash}} \alpha_{n-1} q_{n-1} h(a_n) \\ &\overset{*}{\underset{M_2}{\vdash}} \alpha_n q_n = \alpha q_n \,. \end{aligned}$$

Com que $q \in F_1$ aleshores $q_n \in F_2$ i, per tant, $h(w) \in \mathsf{L}(M_2)$, que és el mateix que dir $w \in h^{-1}(\mathsf{L}(M_2)) \subseteq h^{-1}(L_2)$.

Observeu que la cadena d'implicacions que hem seguit és totalment reversible, i aquest fet ens dóna automàticament l'altra inclusió. □

A la vista de la proposició precedent podem enunciar el teorema següent.

TEOREMA. *La família dels llenguatges incontextuals és tancada respecte de l'aplicació de morfismes inversos.*

Tot i que la construcció que hem donat és general per a NPDA, observeu que en el cas particular de partir d'un DPDA, la mateixa construcció ens permet trobar un DPDA que reconeix el llenguatge $h^{-1}(L_2)$. Per tant, també podem enunciar el corol·lari següent.

COROL·LARI. *La família dels* DCFL *és tancada respecte de l'aplicació de morfismes inversos.*

Complementació d'un DCFL

En aquest apartat enunciarem la propietat que els DCFL, a diferència dels CFL, formen una família tancada respecte de la complementació i veurem les línies generals de la demostració.

La idea essencial per a la construcció d'un DPDA que reconegui el llenguatge complementari del que reconeix un altre DPDA donat és la mateixa que en el cas de DFA: intercanviar els estats acceptadors i no acceptadors. De tota manera, els DPDA presenten tres problemes que fan que aquesta construcció no pugui ser tan senzilla com en el cas dels autòmats finits. Vegem-los a continuació. Sigui M un DPDA qualsevol, L el llenguatge que reconeix i w un mot qualsevol sobre l'alfabet d'entrada. Anomenem M' el DPDA que s'obté de M intercanviant els estats acceptadors pels que no ho són, i viceversa. En la computació de M sobre el mot w es poden donar tres situacions conflictives:

1. Abans que el capçal hagi arribat a l'extrem dret del mot d'entrada, el càlcul pot aturar-se perquè en un moment donat la funció de transició pot no estar definida.

2. L'autòmat pot entrar en un bucle de λ-transicions abans d'haver llegit completament l'entrada.
3. Després que el capçal hagi superat l'extrem dret del mot d'entrada, l'autòmat pot realitzar un seguit de λ-transicions que el portin a passar per estats acceptadors i no acceptadors.

En qualsevol dels dos primers casos, el DPDA M no avançarà més enllà d'un símbol determinat en la cadena d'entrada. Representem per x el prefix de w (diferent de w) que M ha deixat a l'esquerra del capçal quan això succeeix. Evidentment, el mot w no pertany al llenguatge L i tampoc no hi pertanyen tots aquells que tinguin x com a prefix propi, els quals són, per tant, mots de $\overline{L}$. Ara bé, fixeu-vos que M' també rebutja tots els mots amb prefix x, ja que el càlcul de l'autòmat sobre aquests mots segueix sense processar més enllà de x en la cadena d'entrada.

En el tercer cas, convé remarcar que la definició que hem donat de llenguatge reconegut no exigeix que el procés del mot *acabi* en un estat acceptador, sinó només que *passi* per algun estat acceptador un cop llegit el mot sencer. Atès que existeix algun estat acceptador en la cua d'estats que M recorre una vegada ha consumit tot el mot d'entrada, l'autòmat accepta w i, per tant, aquest mot pertany a L. Però, precisament per l'existència d'algun estat no acceptador en aquest conjunt d'estats, sabem que M' també acceptarà el mot w.

Observem, doncs, que en cap de les situacions precedents l'autòmat M' no reconeix el llenguatge $\overline{L}$ com nosaltres preteníem. La construcció d'un DPDA per al complementari es basarà a fer successives modificacions a l'autòmat de partida per evitar els inconvenients que hem esmentat prèviament. Veurem a continuació les línies principals d'aquesta construcció, sense incloure'n les demostracions.[7] A partir d'aquest moment representarem per q qualsevol estat de l'autòmat M i per Z i a símbols qualssevol dels alfabets de pila i d'entrada, respectivament.

1. Evitar les indefinicions de la funció de transició és una tasca senzilla: només hem d'afegir un estat *mort* no acceptador que reculli totes les transicions no definides i que s'encarregui d'avançar el capçal de lectura fins a superar l'extrem dret del mot d'entrada. Formalment, si p és aquest estat mort, cal afegir la transició $Zqa \vdash Z\mathsf{p}$ sempre que la funció de transició original no estigui definida per a Zqa ni per a Zq i, pel que fa a l'estat mort, s'han d'incloure les transicions $Z\mathsf{p}a \vdash Z\mathsf{p}$. Una segona font d'indefinició pot provenir del fet que el DPDA buidi completament la pila abans d'haver processat tota l'entrada. Per tal d'impedir aquests possibles buidats de la pila només cal protegir el fons de pila amb un nou símbol X_0 just a sota del símbol Z_0. Així, si en un moment donat del càlcul apareix X_0 al cim de la pila, vol dir que s'hauria arribat a una situació de buidat de pila i podem fer evolucionar l'autòmat cap a l'estat mort. És a dir, haurem d'incorporar les transicions $X_0qa \vdash Z_0\mathsf{p}$. Per afegir aquest símbol X_0, n'hi ha prou a considerar un nou estat inicial q_0' i una transició addicional $Z_0q_0' \vdash X_0Z_0q_0$.
2. Considerem ara la manera d'evitar que el DPDA pugui entrar en un cicle infinit de càlcul. Aquest fet es donarà sempre que existeixi un estat q, un símbol de pila Z, un mot de pila $\alpha \in \Gamma^*$ i un natural $n \geqslant 1$ tals que $Zq \overset{n}{\vdash} \alpha Zq$, o el que és el mateix

[7] En podeu trobar una demostració detallada a [HoU79].

$$Zq = \alpha_1 q_1 \vdash \alpha_2 q_2 \vdash \cdots \vdash \alpha_n q_n = \alpha Z q\,.$$

És a dir, hi ha un bucle de n passos en el càlcul que només involucra λ-transicions (no es consumeix cap més símbol del mot d'entrada) i que no buida mai el contingut de la pila. Fixeu-vos que aquests bucles no depenen per a res del mot d'entrada i que, per tant, es poden detectar amb l'única informació del DPDA. Per evitar-los, cal tenir en compte si el conjunt d'estats $q_1, q_2, \ldots, q_n$, que forma aquest bucle, conté algun estat acceptador o no. Si cap q_i no és estat acceptador, aleshores és clar que cal rebutjar el mot d'entrada i, per tant, substituirem l'antiga transició que iniciava el cicle en q per una nova transició cap a l'estat mort: $Zq \vdash Z\mathsf{p}$. Si algun dels q_i és acceptador, aleshores el mot d'entrada ha de ser acceptat només en el cas en què el DPDA entri en bucle just quan el capçal estigui a la dreta del mot d'entrada. Per tant, considerarem un nou estat acceptador f i les transicions: $Zq \vdash Z\mathsf{f}$ —per acceptar quan no hi ha més símbols per processar— i $Z\mathsf{f} \vdash Z\mathsf{p}$ —si queden més símbols a l'entrada hauran de ser llegits des de l'estat mort—. Amb aquestes transformacions descrites ens assegurarem que, o bé l'autòmat acaba processant tot el mot sense entrar en un cicle de λ-transicions, o bé, si a partir d'un moment donat l'autòmat només aplica λ-transicions, la pila s'haurà d'acabar buidant i, per tant, hi haurà un pas cap a l'estat mort.

La conseqüència d'haver fet les transformacions esbossades als apartats 1 i 2 és que ara disposem d'un nou DPDA M' que reconeix el mateix llenguatge que l'original i tal que, davant de qualsevol mot d'entrada w, segur que acaba el càlcul i deixa el capçal a la dreta del darrer símbol de w. Per construir, a partir de M', un autòmat M'' que accepti el llenguatge complementari només haurem de tenir en compte la tercera de les dificultats.

3. Una solució possible es basa a marcar els estats de M' amb un segon component que indiqui si el DPDA ha passat per algun estat acceptador o no des de la darrera transició que hagi consumit realment un símbol del mot d'entrada. Per cada estat q de M', n'hi ha d'haver tres en M'': dos d'ells associats als valors cert i fals del segon component, que representarem per (q, C) i (q, F), i un tercer d'acceptació, representat per (q, A). Els estats acceptadors de M'' seran precisament aquells estats amb segon component igual a A. Quant a l'estat inicial, o bé serà (q_0, C) si q_0 és acceptador en M' o bé (q_0, F) altrament. Com que M' és determinista i no té transicions indefinides, es té que per a tot símbol de pila Z i tot estat p, o bé M' té definida una λ-transició del tipus $Zp \vdash \alpha q$ o bé té definides transicions $Zpa \vdash \alpha q$ per a tot símbol a de Σ. Amb això, la funció de transició de M' es traduirà de la manera següent en M'':
   - Les λ-transicions $Zp \vdash \alpha q$ de M' donen lloc, en M'', a les transicions següents: $Z(p, \mathsf{C}) \vdash \alpha(q, \mathsf{C})$ i $Z(p, \mathsf{F}) \vdash \alpha(q, x)$, on x val C o F segons que q sigui estat acceptador o no.
   - Les transicions $Zpa \vdash \alpha q$ de M' donen lloc, en M'', a les transicions següents: $Z(p, \mathsf{C})a \vdash \alpha(q, x)$, $Z(p, \mathsf{F}) \vdash Z(p, \mathsf{A})$ i $Z(p, \mathsf{A})a \vdash \alpha(q, x)$, on x val, com abans, C quan q és estat acceptador i F altrament.

Observeu que l'autòmat M'' bàsicament simula el comportament de M' pel que fa a la transició entre estats i que pot evolucionar cap als estats acceptadors (de tipus A) només en unes condicions determinades: quan M' no hauria passat per cap estat acceptador des del darrer moviment del capçal, és a dir, des dels estats de tipus F. Des d'aquests estats de tipus A s'accepta el mot d'entrada si el capçal de lectura ja se'n troba a la dreta, o bé es continua normalment la simulació de M' en el cas que encara quedin símbols per llegir. Observeu que aquest criteri per accedir als estats acceptadors és el que determina pròpiament el pas al complementari. Així, si M' rebutja un mot w, segur que n'acaba la computació en un estat q no acceptador i que, a més, en la seqüència eventual de λ-transicions que hagi pogut realitzar després de consumir el mot d'entrada no haurà passat per cap estat acceptador, per tant, M'' haurà arribat a l'estat (q, F) i existirà una λ-transició cap a l'estat corresponent (q, A) per acceptar w. En canvi, si M' accepta w, aleshores la seva computació inclou algun estat acceptador q en la seva seqüència terminal de λ-transicions, i la computació de M'' acabarà en un estat de la forma (q', C). Com que els estats acceptadors de M'' només són accessibles directament des dels estats de tipus F, M'' no podrà assolir cap estat acceptador i rebutjarà w.

A la vista dels resultats precedents podem enunciar el teorema següent.

TEOREMA. *La família dels* DCFL *és tancada respecte de la complementació.*

Exercicis

8.1 Construïu autòmats amb pila deterministes per als llenguatges següents:

1. El llenguatge de Dyck $\mathsf{D'}_1^*$ (vegeu-ne la definició a l'exercici 2.15).
2. $\{a^{2i}b^{i+2j}c^{3j+1} \mid i, j > 0\}$
3. $\{a^i b^j c^k d^i \mid i > 0 \ \wedge \ j \geqslant k > 0\}$
4. $\{w \in (a+b)^* \mid 2|w|_a = 3|w|_b\}$
5. $\{w \in (a+b)^* \mid |w|_a \neq |w|_b\}$
6. El conjunt de mots sobre l'alfabet $\{a, b\}$ en què no hi ha cap prefix propi que contingui igual nombre de a's que de b's.

8.2 Construïu gramàtiques inambigües per als llenguatges definits als apartats 4, 5 i 6 de l'exercici precedent.

8.3 Construïu autòmats amb pila per als següents llenguatges incontextuals no deterministes:

1. $\{a^i b^j \mid i = j \ \vee \ 2i = j\}$
2. $\{a^i b^j c^k \mid i = j \ \vee \ j = k\}$
3. $\{a^i b^j c^k d^l \mid i = k \ \vee \ j = l\}$
4. $\{a^i b^j c^k d^l \mid (i = j \wedge k = l) \ \vee \ (i = l \wedge j = k)\}$
5. $\{ab^{i_1}ab^{i_2}\cdots ab^{i_n} \mid n \geqslant 1 \ \wedge \ \exists j \ 1 \leqslant j \leqslant n \ \ i_j = n - j\}$
6. $\{ab^{i_1}ab^{i_2}\cdots ab^{i_n} \mid n \geqslant 1 \ \wedge \ \exists j \ 1 \leqslant j \leqslant n \ \ i_j \neq j\}$

8.4 Demostreu que els llenguatges següents són incontextuals:

1. $\{w \in (a+b)^* \mid w = w^{\mathsf{R}} \ \wedge \ \forall x, y \in (a+b)^* \ w \neq xabay\}$
2. $\{w \in a^+b^+c^+ \mid |w|_b \leqslant |w|_c \ \wedge \ |w|_c = \dot{2}\}$
3. $\{xcy \mid x, y \in (a+b)^* \ \wedge \ |x|_a + |y|_b = \dot{2} \ \wedge \ |x| = |y|\}$
4. $\{a^n w \mid w \in (a+b)^* \wedge ((n \text{ parell} \wedge |w|_a > |w|_b) \ \vee \ (n \text{ senar} \wedge |w|_a < |w|_b))\}$

8.5 Demostreu que per a tot llenguatge L:

1. Si L és reconegut per un NPDA d'acceptació per pila buida aleshores també és reconegut per un NPDA d'acceptació per estat final.

Considereu el model de DPDA d'*acceptació per pila buida* definit, a partir dels DPDA d'acceptació per estat final, de manera anàloga al cas indeterminista. Demostreu que per a tot llenguatge L es té:

2. Si L és reconegut per un DPDA d'acceptació per pila buida aleshores també és reconegut per algun DPDA d'acceptació per estat final.
3. Doneu contraexemples per a la inclusió contrària.

Diem que un llenguatge L és *prefixial* (o bé que compleix la propietat del prefix) quan no conté cap mot que sigui prefix propi d'algun altre, és a dir,

$$\forall x \in L \ \forall y \in \Sigma^+ \quad xy \notin L.$$

4. Demostreu que els llenguatges que es poden reconèixer amb DPDA d'acceptació per pila buida són exactament els DCFL prefixials.

8.6 Demostreu que la intersecció de dos CFL pot no ser incontextual fins i tot en el cas en què tots dos llenguatges siguin deterministes.

8.7 Siguin L un DCFL i R un llenguatge regular qualssevol.

1. Demostreu que el llenguatge LR és DCFL.
2. Demostreu que el llenguatge RL pot no ser DCFL.

8.8 Demostreu que si L és un DCFL, aleshores també ho són els llenguatges següents:

1. El conjunt de mots de L que no contenen com a prefix cap altre mot de L.
2. El conjunt de mots de L que no són prefixos de cap altre mot de L.

8.9 Demostreu que les operacions següents (les quals preserven la condició d'incontextualitat) aplicades sobre DCFL poden donar com a resultat llenguatges no deterministes.

1. Reunió.
2. Concatenació.
3. Tancament positiu i de Kleene.
4. Morfisme directe.
5. Revessament.

8.10 Demostreu que la reunió d'un llenguatge no incontextual i un llenguatge finit és un llenguatge no incontextual. Trobeu, en canvi, un contraexemple que posi de manifest que la concatenació d'un llenguatge no incontextual amb un llenguatge finit no buit pot ser un llenguatge incontextual.

8.11 Demostreu que la reunió, la concatenació i l'estrella de Kleene de llenguatges no incontextuals poden ser llenguatges incontextuals.

8.12 Sigui L un DCFL i R un llenguatge regular. Doneu algorismes que permetin respondre les qüestions següents:

1. L és cofinit.
2. L i R són disjunts.
3. L és un subconjunt de R.
4. R és un subconjunt de L.
5. L i R són iguals.

Demostreu que les qüestions 2 i 3 també són decidibles per a CFL no deterministes.

8.13 Demostreu que tot llenguatge incontextual L és reconegut per algun NPDA que utilitza un alfabet de pila amb només dos símbols.

8.14 Demostreu que a tot PDA li podem associar un altre PDA que reconeix el mateix llenguatge i que satisfà la propietat següent: totes les transicions del nou PDA substitueixen el símbol de cim de pila per dos símbols com a màxim. Expressat formalment, si el nou PDA és de la forma $\langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$, es té

$$\forall p, q \in Q \ \forall a \in \Sigma \cup \{\lambda\} \ \forall Z \in \Gamma \ \forall \alpha \in \Gamma^* \quad (q, \alpha) \in \delta(p, a, Z) \implies |\alpha| \leqslant 2.$$

Comproveu que la construcció del nou PDA a partir del donat pot fer-se en temps polinòmic respecte de la longitud de la dada.

8.15 Definim el model d'*autòmat amb comptador* com un NPDA en què s'ha reduït l'alfabet de pila a un sol símbol, a més del Z_0, i en què la funció de transició s'ha restringit de manera que no pugui utilitzar el símbol Z_0 en cap altre lloc de la pila que no sigui el fons. (Observeu que una pila amb només un símbol "útil" és precisament un comptador en unari.) Es demana:

1. Demostreu que tots els llenguatges presentats a l'exercici 8.1són reconeguts per autòmats amb comptador.
2. Doneu llenguatges incontextuals i, en particular, lineals que no puguin ser reconeguts per cap autòmat amb comptador.

Annex A NPDA que correspon a una CFG

DEMOSTRACIÓ. Demostrarem, en primer lloc, que en G hi ha una derivació del tipus $X \stackrel{*}{\Rightarrow} w\alpha$, on $w \in \Sigma^*$ i $\alpha \in V(\Sigma \cup V)^* \cup \{\lambda\}$, si i només si per a tot mot $x \in \Sigma^*$ en l'autòmat M_G hi ha el càlcul $X\mathsf{p}wx \stackrel{*}{\vdash} \alpha^{\mathsf{R}}\mathsf{p}x$. Tractem les dues implicacions per separat.

a) $X \stackrel{*}{\Rightarrow} w\alpha \implies \forall x \ X\mathsf{p}wx \stackrel{*}{\vdash} \alpha^{\mathsf{R}}\mathsf{p}x$.

Procedim per inducció sobre la longitud, n, de la derivació de la gramàtica.

- *Base*: $n=0$. Si $X \stackrel{0}{\Rightarrow} w\alpha$, aleshores forçosament $w=\lambda$ i $\alpha=X$. Per tant, per a tot mot $x \in \Sigma^*$ es compleix trivialment que $X\mathsf{p}x \stackrel{*}{\vdash} X\mathsf{p}x$.
- *Pas inductiu*. Si $X \stackrel{n+1}{\Longrightarrow} w\alpha$, aleshores podem descompondre aquesta derivació en dues parts: $X \stackrel{n}{\Rightarrow} uA\gamma \Rightarrow uv\beta\gamma$, on $uv=w$, $\beta \in V(\Sigma \cup V)^* \cup \{\lambda\}$, $\gamma \in (V \cup \Sigma)^*$, $\alpha=\beta\gamma$ i la variable A depèn del mot α. Si $\alpha=\lambda$, aleshores A és l'últim node intern de la dreta de l'arbre de derivació, mentre que si α és de la forma $Y\eta$ (amb $Y \in V$ i $\eta \in (V \cup \Sigma)^*$), A és el node pare de Y en l'arbre de derivació. Per a qualsevol dels dos casos de A, aplicant la hipòtesi d'inducció sobre la primera part de la derivació es té

  $$\forall x' \ \ X\mathsf{p}ux' \stackrel{*}{\vdash} \gamma^{\mathsf{R}}A\mathsf{p}x' .$$

  D'altra banda, la producció $A \to v\beta$ de la gramàtica ens assegura que hi haurà una transició $A\mathsf{p} \vdash \beta^{\mathsf{R}}v^{\mathsf{R}}\mathsf{p}$ en l'autòmat. Recordem que, per construcció, el NPDA també conté les transicions $a\mathsf{p}a \vdash \mathsf{p}$ per a tot símbol a de Σ. Ajuntant les tres coses i fent $x'=vx$ per a cada x, es té:

  $$\forall x \in \Sigma^* \quad X\mathsf{p}uvx \ \stackrel{*}{\vdash} \ \gamma^{\mathsf{R}}A\mathsf{p}vx \ \vdash \ \gamma^{\mathsf{R}}\beta^{\mathsf{R}}v^{\mathsf{R}}\mathsf{p}vx \ \stackrel{*}{\vdash} \ \gamma^{\mathsf{R}}\beta^{\mathsf{R}}\mathsf{p}x ,$$

  que és el que volíem demostrar.

b) $X\mathsf{p}wx \stackrel{*}{\vdash} \alpha^{\mathsf{R}}\mathsf{p}x \implies X \stackrel{*}{\Rightarrow} w\alpha$.

Procedim per inducció sobre el nombre n de passos de la transició.

- *Base*: $n=0$. Si $X\mathsf{p}wx \stackrel{0}{\vdash} \alpha^{\mathsf{R}}\mathsf{p}x$, aleshores forçosament $w=\lambda$ i $\alpha=X$. En aquest cas, es compleix trivialment la condició $X \stackrel{*}{\Rightarrow} X$.

- *Pas inductiu*. En aquest cas, $X\mathsf{p}wx \overset{n+1}{\vdash} \alpha^{\mathsf{R}}\mathsf{p}x$. Separarem en aquest càlcul els n primers passos del darrer, distingint dos casos que depenen del tipus de la darrera transició aplicada.
  a) L'última transició és del tipus: $A\mathsf{p} \vdash \gamma^{\mathsf{R}}\mathsf{p}$. Això vol dir, d'una banda, que la gramàtica conté la producció $A \to \gamma$ i, d'altra banda, que es pot separar el càlcul sobre l'autòmat de la manera següent:
  $$X\mathsf{p}wx \overset{n}{\vdash} \beta^{\mathsf{R}}A\mathsf{p}x \vdash \underbrace{\beta^{\mathsf{R}}\gamma^{\mathsf{R}}}_{\alpha^{\mathsf{R}}}\mathsf{p}x\,.$$
  Si apliquem la hipòtesi d'inducció sobre la primera part i utilitzem la producció $A \to \gamma$, obtenim:
  $$X \overset{*}{\Rightarrow} wA\beta \Rightarrow w\gamma\beta = w\alpha\,.$$
  b) L'última transició és del tipus: $a\mathsf{p}a \vdash \mathsf{p}$. En aquest cas, el mot w és de la forma $w'a$ i podem expressar el càlcul sobre l'autòmat de la manera següent:
  $$X\mathsf{p}w'ax \overset{n}{\vdash} \alpha^{\mathsf{R}}a\mathsf{p}ax \vdash \alpha^{\mathsf{R}}\mathsf{p}x\,.$$
  Aplicant la hipòtesi d'inducció sobre la primera part, s'obté el resultat esperat,
  $$X \overset{*}{\Rightarrow} w'a\alpha = w\alpha\,.$$

A partir de l'equivalència que acabem de demostrar es té:

$$\begin{aligned}\forall w \in \Sigma^* \quad w \in \mathsf{L}(G) &\Longleftrightarrow S \overset{*}{\Rightarrow} w\\ &\Longleftrightarrow S\mathsf{p}w \overset{*}{\vdash} \mathsf{p}\\ &\Longleftrightarrow Z_0q_0w \vdash Z_0S\mathsf{p}w \overset{*}{\vdash} Z_0\mathsf{p} \vdash \mathsf{f}\\ &\Longleftrightarrow w \in \mathsf{L}(M_G)\,.\end{aligned}$$

Que és el mateix que dir $\mathsf{L}(G) = \mathsf{L}(M_G)$. □

Annex B CFG que correspon a un NPDA

Demostració. Demostrarem primer l'equivalència següent, que descriu la relació que hi ha entre les transicions de l'autòmat i les derivacions de la gramàtica:

$$\forall w \in \Sigma^* \quad Xpw \overset{*}{\vdash} q \iff [pXq] \overset{*}{\Rightarrow} w, \tag{1}$$

on p i q són estats de l'autòmat i X és un símbol de pila. Tractarem les dues implicacions per separat.

a) $Xpw \overset{*}{\vdash} q \implies [pXq] \overset{*}{\Rightarrow} w.$

Procedirem per inducció sobre el nombre, n, de passos de la transició de l'autòmat.

- *Base*: $n = 1$. Si $Xpw \vdash q$, aleshores es té que $w = a \in \{\lambda\} \cup \Sigma$. Per construcció, hi haurà en la gramàtica la producció $[pXq] \to a$. En conseqüència: $[pXq] \overset{*}{\Rightarrow} w$.
- *Pas inductiu*. En aquest cas, $Xpw \overset{n+1}{\vdash} q$. Podem separar el primer pas d'aquesta transició dels n restants i tenim:
  $$Xpw \vdash B_m \cdots B_1p'x \overset{n}{\vdash} q,$$
  on $Xpa \vdash B_m \cdots B_1p'$ és la transició directa aplicada inicialment per l'autòmat i $w = ax$, amb $a \in \{\lambda\} \cup \Sigma$ i $x \in \Sigma^*$. S'observa que després de la primera transició directa l'autòmat té m símbols a la pila i que, al final, la pila és buida; per tant, al llarg del procés de x els símbols inicials B_1, … , B_m hauran de ser esborrats de la pila, un a un i en aquest ordre. Podem considerar, doncs, la factorització de x en $w_1w_2\cdots w_m$ amb el criteri següent: l'autòmat tindrà el capçal apuntant al principi del submot w_i quan *per primer cop* la pila contingui només els símbols B_i, B_{i+1}, … , B_m, és a dir, quan s'acabi d'esborrar el símbol B_{i-1} anterior.

Això origina una descomposició de la transició $B_m \cdots B_1 p' x \overset{n}{\vdash} q$ en els trossos següents:

$$\begin{aligned} B_m \cdots B_2 B_1 p' w_1 w_2 \cdots w_m &\overset{n_1}{\vdash} B_m \cdots B_2 q_{i_1} w_2 \cdots w_m \\ &\overset{n_2}{\vdash} B_m \cdots B_3 q_{i_2} w_3 \cdots w_m \\ &\cdots \\ &\overset{n_{m-1}}{\vdash} B_m q_{i_{m-1}} w_m \\ &\overset{n_m}{\vdash} q\,. \end{aligned}$$

Com que durant el processament de cada submot w_i l'autòmat no utilitza cap símbol de pila per sota de B_i i al final acaba esborrant el propi B_i, és clar que les transicions següents també són vàlides per a l'autòmat

$$B_1 p' w_1 \overset{n_1}{\vdash} q_{i_1},\ \ B_2 q_{i_1} w_2 \overset{n_2}{\vdash} q_{i_2},\ \ \ldots,\ \ B_m q_{i_{m-1}} w_m \overset{n_1}{\vdash} q.$$

Observeu que $n_i \leqslant n$ per a $1 \leqslant i \leqslant m$. Per tant, podem aplicar la hipòtesi d'inducció a cadascuna de les transicions precedents. S'obtenen com a resultat les derivacions següents:

$$[p' B_1 q_{i_1}] \overset{*}{\Rightarrow} w_1,\ \ [q_{i_1} B_2 q_{i_2}] \overset{*}{\Rightarrow} w_2,\ \ \ldots,\ \ [q_{i_{m-1}} B_m q] \overset{*}{\Rightarrow} w_m.$$

Finalment, la transició inicial de l'autòmat: $Xpa \vdash B_m \cdots B_1 p'$, segur que donarà lloc en la gramàtica a la producció particular

$$[pXq] \to a[p' B_1 q_{i_1}][q_{i_1} B_2 q_{i_2}] \cdots [q_{i_{m-1}} B_m q].$$

Ajuntant cadascun dels fragments de derivació es completa aquesta primera part de la prova.

$$[pXq] \ \Rightarrow\ a[p' B_1 q_{i_1}][q_{i_1} B_2 q_{i_2}] \cdots [q_{i_{m-1}} B_m q] \ \overset{*}{\Rightarrow}\ a w_1 w_2 \cdots w_m = w\,.$$

b) $[pXq] \overset{*}{\Rightarrow} w \implies Xpw \overset{*}{\vdash} q$.

Procedirem per inducció sobre la longitud n de la derivació de la gramàtica.

- *Base*: $n = 1$. Si $[pXq] \Rightarrow w$, aleshores existeix en la gramàtica una producció $[pXq] \to w$ i $w = a \in \{\lambda\} \cup \Sigma$. Per tant, ha d'existir en l'autòmat una transició directa $Xqa \vdash q$. En conseqüència, $Xqw \overset{*}{\vdash} q$.
- *Pas inductiu*. En aquest cas, $[pXq] \overset{n+1}{\Longrightarrow} w$. Podem separar el primer pas d'aquesta derivació de tota la resta i es té, necessàriament:

$$[pXq] \ \Rightarrow\ a[p' B_1 q_{i_1}][q_{i_1} B_2 q_{i_2}] \cdots [q_{i_{m-1}} B_m q] \ \overset{n}{\Rightarrow}\ w\,,$$

ja que la forma genèrica de les produccions no terminals de la gramàtica és:

$$[pXq_{i_m}] \to a[qB_1 q_{i_1}][q_{i_1} B_2 q_{i_2}] \cdots [q_{i_{m-1}} B_m q_{i_m}].$$

Aquest primer pas de derivació ens permet dir que w és de la forma ax, amb $a \in \{\lambda\} \cup \Sigma$, i que en l'autòmat hi ha la transició directa: $Xpa \vdash B_m \cdots B_1 p'$. A més, podem assegurar que el mot x factoritza en $w_1 w_2 \cdots w_m$, amb $w_i \in \Sigma^*$, de manera que:

$$[p' B_1 q_{i_1}] \overset{n_1}{\Longrightarrow} w_1,\ [q_{i_1} B_2 q_{i_2}] \overset{n_2}{\Longrightarrow} w_2,\ \ldots,\ [q_{i_{m-1}} B_m q] \overset{n_m}{\Longrightarrow} w_m,$$

on $n_i \leqslant n$ per a $1 \leqslant i \leqslant m$.

Aplicant la hipòtesi d'inducció a cadascuna d'aquestes derivacions obtenim les transicions vàlides següents per a l'autòmat M:

$$B_1 p' w_1 \overset{*}{\vdash} q_{i_1},\ B_2 q_{i_1} w_2 \overset{*}{\vdash} q_{i_2},\ \ldots,\ B_m q_{i_{m-1}} w_m \overset{*}{\vdash} q.$$

Finalment, componem la primera transició directa amb les precedents per construir una transició vàlida sobre M, que consumeixi el mot w sencer:

$$\begin{aligned} Xpw = Xpaw_1 w_2 \cdots w_m &\vdash B_m \cdots B_2 B_1 p' w_1 w_2 \cdots w_m \\ &\overset{*}{\vdash} B_m \cdots B_2 q_{i_1} w_2 \cdots w_m \\ &\overset{*}{\vdash} B_m \cdots B_3 q_{i_2} w_3 \cdots w_m \\ &\cdots \\ &\overset{*}{\vdash} B_m q_{i_{m-1}} w_m \\ &\overset{*}{\vdash} q\,. \end{aligned}$$

En virtut de l'equivalència 1 que acabem de demostrar, es té:

$$\begin{aligned}\forall w \in \Sigma^* \qquad w \in \mathsf{L}(M) &\Longleftrightarrow \exists q \in Q \;\; Z_0 q_0 w \overset{*}{\vdash} q \\ &\Longleftrightarrow \exists q \in Q \;\; [q_0 Z_0 q] \overset{*}{\Rightarrow} w \\ &\Longleftrightarrow \exists q \in Q \;\; S \Rightarrow [q_0 Z_0 q] \overset{*}{\Rightarrow} w \\ &\Longleftrightarrow w \in \mathsf{L}(G_M)\,.\end{aligned}$$

□

Capítol 9 Autòmats bidireccionals

9.1 Introducció

Els dos models abstractes de computació estudiats fins ara, els autòmats finits i els autòmats amb pila, han estat definits amb la característica comuna de disposar d'un únic capçal de lectura unidireccional. El capçal va avançant d'esquerra a dreta a mesura que llegeix els símbols de l'entrada sense possibilitat de recular. El fet que les dues famílies de llenguatges estudiades, la dels llenguatges regulars i la dels incontextuals, puguin ser caracteritzades mitjançant autòmats unidireccionals, ens ha permès un tractament relativament senzill de les propietats d'ambdues famílies. Penseu, per exemple, en la impossibilitat de caure en bucles infinits en el processament de les entrades, conseqüència de la unidireccionalitat de la lectura. Però l'estudi d'aquests models bàsics quedaria incomplet si no analitzéssim la repercussió que pot tenir eventualment, respecte de la família dels llenguatges acceptats per cada model, el fet d'introduir-hi modificacions que no afectin la seva característica essencial, és a dir, la seva memòria. Tal és el cas de considerar que el capçal de lectura sigui bidireccional, és a dir, que pugui avançar i retrocedir, o que pugui haver-hi dos o més capçals de lectura actuant simultàniament. En aquest capítol estudiarem les repercussions de la primera d'aquestes modificacions.

En el cas dels autòmats finits, els models bidireccionals, tot i que reconeixen la mateixa família de llenguatges que els unidirecionals, permeten en molts casos de simplificar-ne notablement el disseny. En aquest capítol posarem de manifest l'existència de nombrosos llenguatges per als quals es fa difícil trobar directament autòmats finits unidireccionals que els reconeguin. En canvi, veurem que per a aquests mateixos llenguatges és molt fàcil de dissenyar autòmats bidireccionals. La construcció posterior d'un autòmat unidireccional equivalent quedarà aleshores com un procediment mecànic, encara que pugui ser complex.

En el cas dels autòmats amb pila, els models bidireccionals reconeixen una família de llenguatges més àmplia que els unidireccionals. La seva utilitat ve donada en bona part per l'existència d'un algorisme degut a Cook [Coo71] que permet simular el comportament de qualsevol autòmat amb pila determinista bidireccional en temps lineal sobre un computador.

9.2 Autòmats finits bidireccionals

Un *autòmat finit determinista bidireccional* (abreujadament un 2DFA, de l'anglès *2-way* *deterministic finite automaton*) és una estructura de la forma

$$M = \langle Q, \Sigma, \delta, q_0, F\rangle$$

en què Q, Σ, q_0 i F tenen el significat habitual de conjunt finit d'estats, alfabet d'entrada, estat inicial i conjunt d'estats acceptadors respectivament. També aquí δ designa una *funció de transició*, que ara és de la forma

$$\delta\colon Q \times \Sigma \longrightarrow Q \times \{\mathsf{e}, \mathsf{d}\}.$$

El funcionament dels autòmats bidireccionals és el següent. Inicialment l'autòmat està en l'estat q_0 i té el capçal situat davant del primer símbol de l'esquerra del mot d'entrada. Suposem que en un moment determinat l'autòmat es troba en un estat p i el símbol que hi ha al davant del capçal és a. Suposem que la funció de transició pren el valor $\delta(p, a) = (q, m)$. En aquestes condicions, l'autòmat passa de l'estat p a l'estat q, i mou el capçal una posició a l'esquerra o a la dreta segons que m valgui e o d, respectivament.[1]

Observem que un DFA dels estudiats fins ara, i que aquí anomenem *unidireccional*, pot ser interpretat com un cas particular d'un 2DFA en què tots els valors de la funció de transició són de la forma (q, d). En tal cas, sols el valor del nou estat q és rellevant, i sabem que només hi ha una extensió posible de la funció δ al domini $Q \times \Sigma^*$. En canvi, en el cas general dels 2DFA, no té sentit considerar una extensió d'aquest tipus. Designem per δ_1 la primera de les dues components de la funció δ, és a dir la que defineix el canvi d'estat. Suposem que estenguéssim aquesta funció a mots de manera que $\delta_1(p, w)$ fos l'estat en què es trobaria l'autòmat quan el capçal sortís per la dreta després de processar el mot w partint de l'estat p i del capçal situat a la primera posició de w. (Podríem considerar que els casos en què el capçal surt per l'esquerra de w o aquells en què l'autòmat entra en bucle a mig llegir w, donen lloc a una transició a un estat mort.) No podríem establir la igualtat entre $\delta_1(p, xy)$ i $\delta_1(\delta_1(p, x), y)$, ja que poden donar-se situacions del tipus

$$\begin{cases} \delta_1(p, x_1) = \delta_1(p, x_2) = q \\ \delta_1(p, x_1 y) \neq \delta_1(p, x_2 y)\,, \end{cases}$$

en què el comportament de $\delta_1(q, y)$ depèn del prefix que precedeix y i no sols de l'estat q.

Per tal de descriure la situació a què s'arriba després d'efectuar un nombre qualsevol de passos, introduirem el concepte de *configuració* o *descripció instantània*. Suposarem, sense pèrdua de generalitat, que Σ i Q no tenen símbols en comú. Definim el conjunt $\mathsf{ID}(M)$ de configuracions així:

$$\mathsf{ID}(M) = \Sigma^* Q \Sigma^*,$$

i interpretem que una seqüència xqy de $\mathsf{ID}(M)$ descriu la situació en què l'autòmat M, que està processant l'entrada $w = xy \in \Sigma^*$, es troba en l'estat $q \in Q$ i té el capçal situat davant del primer símbol del sufix y.

Per formalitzar el concepte de pas de procés (o de transició) de l'autòmat M, definirem en el conjunt $\mathsf{ID}(M)$ una *relació de transició directa*, que representarem pel signe $\vdash_{\overline{M}}$ (o simplement $\vdash$, quan M estigui sobreentès) de la manera següent. Cada valor de la forma

[1] És freqüent admetre una tercera possibilitat, a saber, que el capçal quedi immòbil. És fàcil de veure que aquesta modificació del model no afecta el conjunt dels llenguatges reconeguts.

$\delta(p, a) = (q, \mathsf{d})$ de la funció de transició indueix el conjunt de totes les transicions directes de la forma

$$xpay \vdash xaqy \qquad \forall x, y \in \Sigma^*,$$

mentre que cada valor de la forma $\delta(p, a) = (q, \mathsf{e})$ de la funció de transició indueix el conjunt de totes les transicions directes de la forma

$$xspay \vdash xqsay \qquad \forall s \in \Sigma \ \text{ i } \ \forall x, y \in \Sigma^*.$$

I no hi ha altres transicions directes que les definides per les regles anteriors.

Podem considerar ara el tancament reflexiu i transitiu d'aquesta relació de transició. Es tracta de considerar que dues configuracions estan relacionades quan la segona s'obté a partir de la primera mitjançant zero o més transicions directes a través de configuracions intermèdies. Representarem pel signe $\overset{*}{\vdash}$ aquesta relació de transició de zero o més passos. Formalment, la definició és la següent. Siguin $k, k' \in \mathsf{ID}(M)$ dues configuracions,

$$k \overset{*}{\vdash} k' \iff \exists n \geqslant 0 \ \exists k_0, \ldots, k_n \in \mathsf{ID}(M) \begin{cases} k = k_0 \\ \forall i < n \quad k_i \vdash k_{i+1} \\ k_n = k'. \end{cases}$$

La utilitat dels conceptes que acabem d'introduir es posa de manifest en la facilitat amb què podem ara descriure el llenguatge reconegut per un 2DFA. Es tracta d'acceptar els mots que fan passar l'autòmat des de la situació que abans hem definit com a inicial (en estat q_0 i amb el capçal davant del primer símbol) a una situació en què el capçal quedi a la dreta de l'últim símbol del mot i l'estat sigui un dels acceptadors.[2] És clar que això pot ser expressat formalment de la manera següent:

$$\mathsf{L}(M) = \{w \in \Sigma^* \mid \exists p \in F \ q_0 w \overset{*}{\vdash} wp\}.$$

Com a conseqüència de la definició de les transicions directes, no hi ha cap configuració que pugui venir a continuació (a través d'un o més passos) d'una de la forma wq, que correspon a tenir el capçal a la dreta de l'últim símbol de l'entrada. Tampoc no hi ha cap configuració que pugui seguir a una de la forma qay quan $\delta(q, a)$ és de la forma (p, e), que correspon a un moviment cap a l'esquerra quan el capçal és al davant del primer símbol de l'entrada. Observem que M accepta el mot buit si i sols si $q_0 \in F$, ja que en aquest cas la reflexivitat de la relació $\overset{*}{\vdash}$ permet que la condició d'acceptació s'escrigui de la forma $(q_0 \in F \ \wedge \ q_0 \overset{*}{\vdash} q_0)$.

Exemple 9.1 Considerem el llenguatge format per λ i pels nombres binaris que, seguits del seu revessat, són múltiples de cinc. Formalment,

$$L = \{w \in \{0,1\}^* \mid \mathsf{valor}_2(ww^{\mathsf{R}}) = \dot{5}\}.$$

A la figura 9.1 s'ha dibuixat el diagrama de transicions d'un 2DFA que reconeix el llenguatge **¢L\$**, és a dir, el format pels mots de L però escrits entre els símbols **¢** i **\$**. Observeu que l'única diferència que hi ha entre aquest graf i els dels DFAunidireccionals és que ara l'etiqueta de cada arc inclou, a més del símbol llegit, el moviment del capçal.

[2] És fàcil de veure que la classe de llenguatges reconeguts és independent de la convenció concreta que adoptem en aquest punt. Podríem convenir que el capçal quedés, per exemple, a l'esquerra del primer símbol.

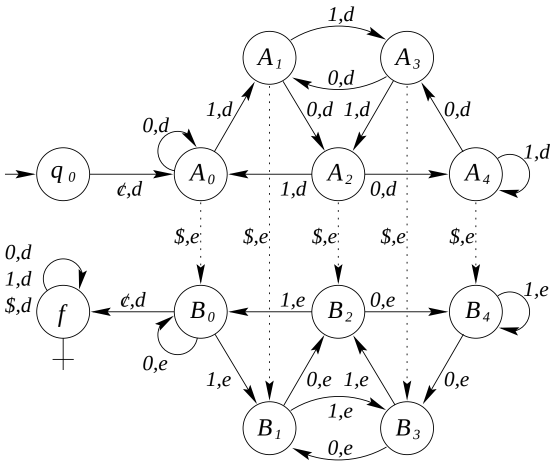

Fig. 9.1 *Diagrama de transicions d'un* 2DFA

Taula 9.1 *Funció de transició del* 2DFA *de la figura 9.1*

	¢	0	1	$
q_0	A_0, d	r, d	r, d	r, d
r	r, d	r, d	r, d	r, d
A_0	r, d	A_0, d	A_1, d	B_0, e
A_1	r, d	A_2, d	A_3, d	B_1, e
A_2	r, d	A_4, d	A_0, d	B_2, e
A_3	r, d	A_1, d	A_2, d	B_3, e
A_4	r, d	A_3, d	A_4, d	B_4, e
B_0	f, d	B_0, e	B_1, e	r, d
B_1	r, d	B_2, e	B_3, e	r, d
B_2	r, d	B_4, e	B_0, e	r, d
B_3	r, d	B_1, e	B_2, e	r, d
B_4	r, d	B_3, e	B_4, e	r, d
† f	f, d	f, d	f, d	f, d

Per tal de no perdre claredat, no han estat dibuixats els arcs que corresponen a transicions que duen a un únic estat mort, r, que tampoc no apareix al gràfic. L'especificació completa de la funció de transició ve donada a la taula 9.1.

L'autòmat s'ha construït a partir de dues còpies del DFA que reconeix els múltiples de cinc, que com és sabut té un estat per cada residu mòdul cinc. La còpia de la part superior del graf és recorreguda mentre el capçal llegeix l'entrada d'esquerra a dreta. Quan el capçal arriba a la marca de la dreta, hi ha una transició des de l'estat de la part superior a l'estat de la part inferior que correspon al mateix residu. A partir d'aquí s'inverteix el moviment del capçal, l'entrada torna a ser llegida de dreta a esquerra i és processada per la part inferior de l'autòmat. Si en arribar de nou a la marca de l'esquerra l'autòmat es troba en l'estat B_0, es produeix una transició a l'estat acceptador f que ja no és abandonat i que fa sortir el capçal per l'extrem de la dreta, per tal d'acabar correctament. En qualsevol altre cas, el procés acaba necessàriament a l'estat r i el mot és rebutjat.

És interessant observar que també podem construir un DFA unidireccional per reconèixer el llenguatge L. Es tracta de considerar simultàniament l'autòmat M_5 que reconeix els múltiples de cinc i l'autòmat M_5^{R} que reconeix el revessat d'aquest llenguatge. No és difícil adonar-se que l'autòmat que accepta L pot ser construït considerant estats amb dues components. La primera conté l'estat a què arriba M_5 en llegir un mot w d'esquerra a dreta. La segona conté l'estat en què caldria estar per tal que el mateix M_5, llegint w^{R}, arribés a l'estat acceptador, que és precisament l'estat a què arriba M_5^{R} en llegir w. Així doncs, l'autòmat cercat s'obté fent el producte cartesià de M_5 i M_5^{R} i prenent com a estats acceptadors els que tenen les dues components iguals entre si. Formalment, sigui $Q = \{q_0, q_1, q_2, q_3, q_4\}$ i sigui

$$M_5 = \langle Q, \{0\,,1\}, \delta_5, q_0, \{q_0\}\rangle,$$

on δ_5 està definida, quan $a \in \{0\,,1\}$, per la regla aritmètica

$$\delta_5(q_i, a) = q_j \iff 2 \cdot i + a \equiv j \pmod 5.$$

M_5^{R} té una estructura idèntica a la de M_5 amb l'única diferència que la seva funció de transició obeeix la regla

$$\delta_5^{\mathsf{R}}(q_i, a) = q_j \iff \delta_5(q_j, a) = q_i.$$

En aquestes condicions, l'autòmat M que reconeix L és

$$M = \langle Q \times Q, \{0\,,1\}, \delta, [q_0, q_0], F\rangle,$$

on

$$\delta([q_i, q_j], a) = [\delta_5(q_i, a), \delta_5^{\mathsf{R}}(q_j, a)]$$

i on

$$F = \{[q_i, q_i] \mid 0 \leqslant i \leqslant 4\}.$$

L'autòmat M construït així té 25 estats i és mínim.

A l'exemple anterior, en lloc de construir un 2DFA per al llenguatge L proposat, l'hem construït per al llenguatge $¢L\$$, on els símbols ¢ i \$ no formen part de l'alfabet de L. La raó de fer això ha estat, com resulta evident, facilitar el disseny de l'autòmat. El recurs a aquests símbols addicionals, que anomenem *marcadors d'extrems*, permet identificar el moment en què el capçal llegeix l'últim símbol del mot d'entrada sense provocar a continuació l'aturada de l'autòmat. Sense ells, aquesta identificació es fa impossible. Tot i això, en aquest capítol demostrarem que tot llenguatge acceptat per un 2DFA també és acceptat per algun DFA unidireccional. Per tant, donat un llenguatge L, si sabem construir un 2DFA que reconegui $¢L\$$, també podem construir un DFA unidireccional per a $¢L\$$. Aleshores, és immediat passar d'aquest DFA a un altre que accepti L. Així doncs, la introducció dels marcadors d'extrems no representa cap ampliació (ni tampoc cap restricció) dels llenguatges que poden ser acceptats per 2DFA. En conseqüència, en tots els exemples que donarem d'utilització de 2DFA farem ús dels marcadors d'extrems.

9.3 El problema de l'aturada en 2DFA

La bidireccionalitat del moviment del capçal introdueix, respecte de la unidireccionalitat, la possibilitat que els processos de càlcul no s'aturin. N'hi ha prou de considerar, per exemple, que la funció de transició inclogui un parell d'especificacions de la forma

$$\delta(p, a) = (q, \mathsf{d}) \quad \text{i} \quad \delta(q, b) = (p, \mathsf{e})$$

perquè l'autòmat entri en bucle així que llegeixi el submot ab, si ho fa partint de l'estat p. Aquesta possibilitat de no-aturada, que no existeix en els autòmats unidireccionals, està implícita, en canvi, en molts altres models de computació. En particular, quan es considera el conjunt de programes que poden ser escrits en qualsevol llenguatge usual de programació, és clar que aquest conjunt inclou programes que poden donar lloc a bucles sense fi. En general, el problema de determinar si un cert programa dóna lloc a un procés finit o infinit quan processa una entrada donada és un problema *indecidible*.[3]

En el cas dels autòmats finits bidireccionals, hi ha una manera senzilla de decidir quan un autòmat de s estats que llegeix una entrada de longitud n entra en bucle. Es tracta d'anar considerant les configuracions successives per les quals passa l'autòmat, partint de la configuració inicial i acabant, o bé quan el capçal creua un dels extrems de l'entrada, o bé quan es repeteix alguna de les configuracions considerades. La idea important és que, perquè l'autòmat entri en bucle, és condició necessària i suficient que hi hagi alguna configuració que es repeteixi. És clar que la condició és suficient. La necessitat deriva del fet que hi ha un nombre finit, $n \times s$, de configuracions possibles, cosa que, d'altra banda, garanteix la finitud del procediment proposat, que requereix un màxim de $n \times s$ passos.

9.4 Construcció d'un NFA unidirecional a partir d'un 2DFA

A l'exemple 9.1 hem exposat un 2DFA que reconeix un cert llenguatge. També hem posat de manifest que aquest mateix llenguatge pot ser reconegut per un DFA unidireccional. Aquesta possibilitat no és cap cas particular. Veurem tot seguit la manera de construir un NFA unidireccional a partir d'un 2DFA qualsevol. La demostració original d'aquest fet es troba a [RaS59] i a [She59], si bé nosaltres seguirem en aquest punt la referència [HoU79]. De totes maneres, cal advertir el lector que, per tal de facilitar la comprensió d'aquesta exposició, hem optat per un tractament informal d'algunes de les idees que introduirem en la demostració. Començarem presentant un exemple que ens servirà de referència al llarg de l'exposició que segueix.

Exemple 9.2 La taula 9.2 especifica la funció de transició d'un 2DFA de tres estats, dels quals q_0 és l'inicial i q_2 és l'únic estat acceptador.

Taula 9.2 *Funció de transició d'un* 2DFA

	a	b
q_0	q_1, d	q_0, e
q_1	q_2, e	q_2, d
† q_2	q_0, d	q_1, e

Per a cada 2DFA considerat i cada mot processat per l'autòmat podem dibuixar un

[3] Sobre el significat d'aquest terme, vegeu el comentari de la secció 2.4.

diagrama que reflecteixi aquest procés. Es tracta d'escriure els estats successius pels quals va passant l'autòmat, situant-los en la posició que ocupa el capçal just en el moment en què es disposa a llegir un nou símbol, és a dir, en la frontera entre el símbol que acaba de llegir i el nou símbol. Aquesta successió comença amb l'estat inicial escrit en la frontera esquerra del mot. Cada vegada que el capçal inverteix el sentit del moviment, el diagrama baixa una línia i continua escrivint els estats, de manera que la seqüència dels estats segueix, cap a la dreta i cap a l'esquerra, els moviments del capçal.

DEFINICIÓ. Anomenem *seqüències de creuaments* les subseqüències d'estats que apareixen en un diagrama de procés, i que corresponen als moments en què el capçal creua cada una de les fronteres entre dos símbols consecutius (més les dues fronteres dels extrems).

Exemple 9.3 La figura 9.2 il·lustra el diagrama del procés d'acceptació del mot *aabab* per l'autòmat definit a l'exemple 9.2.

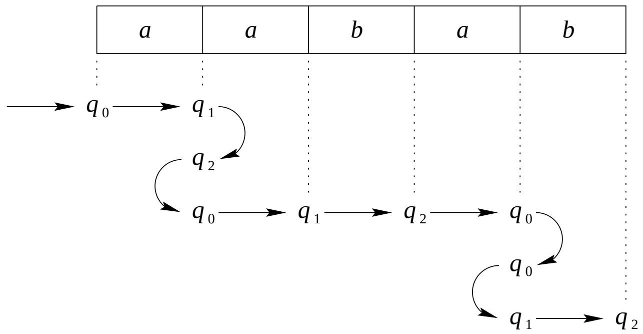

Fig. 9.2 *Diagrama d'un procés d'acceptació*

Les seqüències de creuaments corresponents a aquest procés són, d'esquerra a dreta, $[q_0]$, $[q_1, q_2, q_0]$, $[q_1]$, $[q_2]$, $[q_0, q_0, q_1]$ i $[q_2]$.

Observeu que els estats que ocupen posicions senars en una seqüència de creuaments corresponen a situacions en què el capçal s'està movent cap a la dreta. Per tant, el símbol que serà llegit en aquest estat és el que apareix a la dreta de la col·locació de la seqüència considerada en el diagrama. Anàlogament, els estats en posicions parelles corresponen a moviments del capçal cap a l'esquerra.

Observeu també que no totes les seqüències d'estats possibles són seqüències de creuaments. Per exemple, la seqüència

$$[q_3, q_5, q_2, q_5, q_1, q_4, q_4]$$

no pot ser cap seqüència de creuaments, perquè l'estat q_5, que apareix dues vegades en posició parella, hauria de donar origen, a causa del determinisme, a una repetició indefinida del procés. És a dir, una seqüència de creuaments que comença amb els quatre primers components $[q_3, q_5, q_2, q_5, \ldots]$, forçosament ha de ser la seqüència periòdica infinita

$$[q_3, q_5, q_2, q_5, q_2, q_5, q_2, \ldots].$$

Així doncs, una seqüència de creuaments és finita si i sols si no té cap estat repetit en dues posicions de la mateixa paritat.

Observeu, finalment, que les seqüències de creuaments que intervenen en processos de mots que són acceptats són totes elles de longitud senar. Altrament el capçal no hauria acabat el procés sortint per l'extrem de la dreta.

DEFINICIÓ. Diem que una seqüència de creuaments és *vàlida* quan és de longitud senar i no té cap parell d'estats repetits en posicions d'igual paritat.

Com a resultat del que acabem de dir i utilitzant aquesta definició, podem fer dues afirmacions. La primera és que en el procés d'acceptació d'un mot per un 2DFA només intervenen seqüències de creuaments vàlides. La segona és que, donat un 2DFA, el nombre de seqüències de creuaments vàlides d'aquest autòmat és, amb independència de l'entrada, finit. Aquest fet és rellevant perquè ens permetrà d'identificar les seqüències de creuaments vàlides amb els estats d'un autòmat unidireccional equivalent.

Considerem ara la relació que ha d'existir entre dues seqüències de creuaments vàlides que apareguin l'una immediatament a la dreta de l'altra en un procés que dugui a l'acceptació d'algun mot. Siguin s_1 i s_2 aquestes seqüències, on s_2 suposem que apareix immediatament a la dreta de s_1. Sigui a el símbol de l'entrada que hi ha entre s_1 i s_2. Si bé l'aparició d'aquestes dues seqüències en un punt determinat està condicionada per la globalitat del mot de l'entrada, ens restringirem únicament als condicionaments que comporta el fet que les dues seqüències estiguin separades pel símbol a. Observem que només un nombre reduït de seqüències pot aparèixer a la dreta de s_1 i del símbol a. Per descomptat, s_2 és una d'elles, però pot no ser l'única. Suposem, per exemple, que la funció de transició de l'autòmat conté una especificació de la forma $\delta(q_i, a) = (q_j, \mathsf{d})$ i que la seqüència s_2 no conté l'estat q_i en cap posició parella ni l'estat q_j en cap posició senar. Aleshores podem intercalar q_i seguit immediatament de q_j en qualsevol punt de s_2, amb la condició que q_i quedi en posició parella. La seqüència obtinguda continua essent vàlida i podria aparèixer a la dreta de la s_1 si només ens atinguéssim al condicionament local del símbol a que estem considerant.

Exemple 9.4 La figura 9.3 il·lustra el cas de dues seqüències, $s_1 = [q_1, q_2, q_3]$ i $s_2 = [q_4, q_5, q_6]$, que s'avenen a l'esquerra i la dreta, respectivament, d'un símbol a. Suposant que q_i no coincideix amb q_5 i que q_j no coincideix ni amb q_4 ni amb q_6, la seqüència $s_2' = [q_4, q_i, q_j, q_5, q_6]$ també podria aparèixer en el lloc de s_2 si només ens atenim al condicionament del símbol a considerat i prescindim de la resta del procés.

Estem ja en condicions de definir un NFA unidireccional que reconegui el llenguatge acceptat per un 2DFA donat. Sigui $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un 2DFA qualsevol. Considerem el NFA $M' = \langle Q', \Sigma, \delta', I', F'\rangle$, els components del qual es defineixen a continuació.

- Q' és el conjunt de seqüències de creuaments vàlides de M.
- δ' és la funció que a cada seqüència de creuaments vàlida i cada símbol de Σ fa correspondre el conjunt de seqüències de creuaments vàlides que poden aparèixer a la dreta de la seqüència i del símbol considerats.
- I' és un conjunt reduït a una sola seqüència: $\{[q_0]\}$.
- F' és el conjunt de seqüències de la forma $[q_i]$ amb $q_i \in F$.

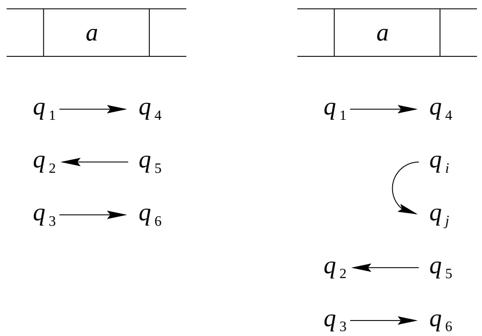

Fig. 9.3 *Dues seqüències dretes per a una mateixa seqüència esquerra*

Abans de comprovar que M' reconeix el mateix llenguatge que M, donarem un exemple d'aquesta construcció.

Exemple 9.5 Considerem de nou el 2DFA de l'exemple 9.2. Com que es tracta d'un autòmat de tres estats, el nombre de seqüències de creuaments vàlides és 57 (vegeu exercici 9.3). Partim de la seqüència inicial $[q_0]$. Com que $\delta(q_0, b) = (q_0, \mathsf{e})$, no hi pot haver cap seqüència vàlida a la dreta de $[q_0]$ quan el símbol considerat és una b. En canvi, la seqüència $[q_1]$ pot aparèixer a la dreta de $[q_0]$ quan el símbol és a. I també hi pot aparèixer la seqüència $[q_1, q_2, q_0]$, ja que $\delta(q_2, a) = (q_0, \mathsf{d})$. No hi ha cap més seqüència vàlida que pugui aparèixer a la dreta de $[q_0]$ amb entrada a. Així doncs, ja podem escriure la primera línia de la taula de δ'. Com que en aquesta línia s'han introduït les seqüències $[q_1]$ i $[q_1, q_2, q_0]$, continuem la taula analitzant quines són les seqüències vàlides que poden aparèixer a la dreta d'aquestes en els casos d'entrada a i b. Prosseguim d'aquesta manera fins a completar el conjunt de seqüències accessibles, i obtenim la funció de transició δ' que apareix a la taula 9.3.

Taula 9.3 *Funció de transició de l'autòmat indeterminista M'*

	a	b
$[q_0]$	$[q_1], [q_1, q_2, q_0]$	—
$[q_1]$	—	$[q_2]$
$[q_1, q_2, q_0]$	$[q_1], [q_1, q_2, q_0]$	—
† $[q_2]$	$[q_0], [q_0, q_0, q_1]$	—
$[q_0, q_0, q_1]$	—	$[q_2]$

Un cop determinitzat i minimitzat aquest autòmat, obtenim el DFA de quatre estats que reconeix el llenguatge $a(a + ba)^*b$.

PROPOSICIÓ. $\mathsf{L}(M') = \mathsf{L}(M)$.

DEMOSTRACIÓ. La inclusió en un dels sentits és molt senzilla. Per a cada mot acceptat per l'autòmat bidireccional M podem dibuixar el diagrama de procés que li correspon.

La successió de seqüències de creuaments que apareix en aquest diagrama constitueix un camí acceptador del mot considerat en l'autòmat indeterminista M'.

La inclusió en sentit contrari requereix una especificació més formal de la funció de transició δ' per tal de poder aplicar correctament una prova per inducció. El lector pot trobar-la a la referència [HoU79] citada anteriorment. □

Com a resultat de la proposició anterior tenim el teorema següent.

TEOREMA. *La família dels llenguatges reconeguts per autòmats finits deterministes bidireccionals és la família dels llenguatges regulars.*

La possibilitat de construir un autòmat determinista unidireccional a partir d'un de bidireccional és de molta utilitat en nombrosos problemes. Així, donat un llenguatge regular $L \subseteq \Sigma^*$ qualsevol, és fàcil generalitzar la construcció donada a l'exemple 9.1 per demostrar l'existència d'un 2DFA que reconeix el llenguatge

$$\{w \in \Sigma^* \mid ww^{\mathsf{R}} \in L\}.$$

9.5 Autòmats finits indeterministes bidireccionals

De manera anàloga als models deterministes, podem definir els *autòmats finits indeterministes bidireccionals* —abreviadament 2NFA— com una generalització dels deterministes unidireccionals, on ara la funció de transició adopta la forma

$$\delta: Q \times \Sigma \longrightarrow \mathcal{P}\left(Q \times \{\mathsf{e}, \mathsf{d}\}\right).$$

És a dir, a cada estat i símbol d'entrada es fa correspondre un conjunt de parells formats per un nou estat i un moviment del capçal.

Pot observar-se que el domini de definició de δ està restringit a $Q \times \Sigma$, en lloc de ser $\mathcal{P}(Q) \times \Sigma^*$ com en el cas unidireccional. Recordem que en el cas unidireccional, és indiferent definir la funció de transició d'una manera o d'una altra. Una funció definida sobre el domini restringit només pot ser estesa d'una única manera que sigui compatible amb els axiomes que caracteritzen les funcions de transició. En canvi, i per les mateixes raons adduïdes en el cas dels 2DFA, no té cap sentit aquest tipus d'extensió en els 2NFA.

La construcció d'un NFA unidireccional a partir d'un 2DFA, exposada a la secció precedent, s'aplica sense cap modificació al cas dels 2NFA. A cada mot acceptat per un 2NFA li pot correspondre una pluralitat de processos d'acceptació, per a cada un dels quals podrem considerar el diagrama de procés corresponent, definit de forma idèntica que en el cas dels 2DFA. Així doncs, el concepte de seqüència de creuaments vàlida es manté sense cap modificació. A l'hora de determinar el conjunt de seqüències de creuaments vàlides que poden aparèixer a la dreta d'una de donada, per a cada símbol de l'alfabet, se segueix el mateix criteri d'atenir-se únicament al símbol considerat, amb independència dels mots en què aquest símbol és processat. El fet que ara partim d'un autòmat indeterminista farà que, en general, calgui considerar un major nombre de possibilitats. Però la construcció és la mateixa. Podem, doncs, enunciar el teorema següent.

TEOREMA. *La família dels llenguatges reconeguts per autòmats finits indeterministes bidireccionals és la família dels llenguatges regulars.*

Tenim, com a conclusió, que els 2NFA representen el model més versàtil de tots els introduïts en aquest curs per caracteritzar els llenguatges regulars.

Exemple 9.6 És fàcil demostrar, via 2NFA, que el conjunt de les *primeres meitats* dels mots d'un llenguatge regular també és regular. És a dir que si $L_1 \subseteq \Sigma^*$ és un llenguatge regular, també ho és el llenguatge

$$L_2 = \{x \in \Sigma^* \mid \exists y \ \ |x| = |y| \ \wedge \ xy \in L_1\}.$$

Per demostrar-ho, partirem d'un DFA unidireccional que accepti L_1. Sigui $M_1 = \langle Q, \Sigma, \delta, q_0, F\rangle$ aquest autòmat. Considerem una "còpia" Q' dels estats de Q, de manera que per a cada estat $q \in Q$ representem per q' l'estat corresponent de Q'. Construirem un autòmat indeterminista bidireccional que reconegui ¢L_2\$. Aquest 2NFA tindrà la forma següent:

$$M_2 = \langle Q \cup Q' \cup \{q_0'', q_f''\}, \Sigma \cup \{¢, \$\}, \delta', \{q_0''\}, \{q_f''\}\rangle.$$

Els estats q_0'' i q_f'' són estats nous i són els únics inicial i acceptador, respectivament, de M_2. La funció δ' està constituïda per les transicicions següents:

1. Una transició inicial,
$$\delta'(q_0'', ¢) = \{(q_0, \mathsf{d})\}.$$
2. Totes les transicions de M_1,
$$\forall q \in Q \ \forall a \in \Sigma \quad \delta'(q, a) = \{(\delta(q, a), \mathsf{d})\}.$$
3. Transicions que fan passar dels estats de Q als corresponents de Q' quan s'arriba a l'extrem dret de l'entrada, i que inicien el recorregut en sentit contrari,
$$\forall q \in Q \quad \delta'(q, \$) = \{(q', \mathsf{e})\}.$$
4. Transicions entre estats de Q',
$$\forall p \in Q \ \forall a \in \Sigma \quad \delta'(p', a) = \{(q', \mathsf{e}) \mid \exists s \in \Sigma \ \delta(p, s) = q\}.$$
5. Transicions finals per al cas d'acceptació,
$$\forall q \in F \quad \delta'(q', ¢) = \{(q_f'', \mathsf{d})\},$$
$$\forall a \in \Sigma \cup \{\$\} \quad \delta'(q_f'', a) = \{(q_f'', \mathsf{d})\}.$$

Observem que entre dos estats donats, o bé existeixen les transicions del tipus 4 per a tots els símbols d'entrada, o bé no existeixen per a cap símbol. Per tant, donada una seqüència qualsevol de $n+1$ estats de Q, $(q_{i_0}, \ldots, q_{i_n})$, les dues afirmacions següents són equivalents:

1. *Algun* mot y de longitud n permet recórrer de manera determinista la seqüència $(q_{i_0}, \ldots, q_{i_n})$ sobre el graf de l'autòmat M_1.
2. *Tot* mot y de longitud n permet recórrer de manera indeterminista la seqüència $(q'_{i_0}, \ldots, q'_{i_n})$ sobre el graf de l'autòmat M_2, movent el capçal cap a l'esquerra.

Farem el raonament de les dues inclusions corresponents a la igualtat $\mathsf{L}(M_2) = L_2$.

1. $\mathsf{L}(M_2) \subseteq L_2$.

Si $x \in \mathsf{L}(M_2)$, els tipus de transicions definides a M_2 requereixen que, per acceptar un mot, calgui llegir-lo primer d'esquerra a dreta fins a arribar a la marca \$; a continuació, no es pot fer altra cosa que llegir l'entrada de dreta a esquerra fins a arribar a la marca

¢ i finalment cal llegir de nou l'entrada d'esquerra a dreta fins a superar la marca \$ de la dreta. Així doncs, ha d'haver-hi a M_2 un camí acceptador de la forma descrita a la figura 9.4.

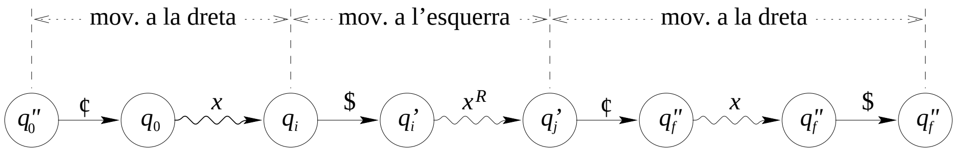

Fig. 9.4 *Esquema de camí acceptador*

Això implica dues coses:

1.1 $q_j \in F$.

1.2 Com que el mot x^{R} permet passar de q_i' a q_j' en M_2, hi ha d'haver algun mot y (en general, diferent de x^{R}) de la mateixa longitud que x^{R} que permeti passar de q_i a q_j en M_1.

Per tant, $\exists y \;\; |y| = |x| \wedge xy \in L_1$, i resulta $x \in L_2$.

2. $L_2 \subseteq \mathsf{L}(M_2)$.

Si $x \in L_2$, tenim que $\exists y \;\; |y| = |x| \wedge xy \in L_1$. Això vol dir que existeix un estat q_i i un estat $q_j \in F$ de l'autòmat M_1 tals que $q_0 x = q_i$ i que $q_i y = q_j$. Ara bé, l'existència d'aquesta segona transició comporta que, per a tot y' tal que $|y'| = |y|$, l'autòmat M_2 pot passar en un recorregut de dreta a esquerra de l'estat q_i' a l'estat q_j'. En particular, considerant $y' = x^{\mathsf{R}}$ resulta $x \in \mathsf{L}(M_2)$.

9.6 Autòmats amb pila bidireccionals

A diferència del que acabem de veure amb els autòmats finits, la bidireccionalitat del capçal dota els autòmats amb pila deterministes d'una potència de càlcul superior a la dels DPDA unidireccionals. La definició dels *autòmats amb pila deterministes bidireccionals*, per als quals usarem l'abreviatura 2DPDA, es fa a partir dels DPDAunidireccionals de manera anàloga a l'efectuada per als 2DFA a partir dels DFA unidireccionals. És a dir, únicament cal canviar la definició de la funció de transició afegint-hi una component addicional que especifiqui el tipus de moviment, a l'esquerra o a la dreta, que ha d'efectuar el capçal a cada transició. És clar que els DPDA unidireccionals es converteixen en casos particulars de 2DPDA. Així doncs, la família dels DCFL està inclosa en la dels llenguatges reconeguts per 2DPDA. Però els 2DPDA poden reconèixer llenguatges que no són DCFL, i ni tan sols són CFL. Vegem-ho amb un exemple.

Exemple 9.7 Al capítol 7 vam demostrar que el llenguatge

$$L = \{a^n b^n c^n \mid n \geqslant 0\}$$

no és CFL. En canvi, és ben senzill imaginar un 2DPDA que reconegui aquest llenguatge. Es tracta de llegir d'esquerra a dreta el prefix de a's carregant-les a la pila, i continuar la lectura de les b's, també d'esquerra a dreta, esborrant una a de la pila per cada b llegida. Si s'ha trobat el mateix nombre de a's que de b's, es fa recular el capçal fins a la primera

b de l'esquerra i es carreguen les b's a la pila. Finalment, llegint d'esquerra a dreta, es comprova que el nombre de c's de l'entrada sigui igual al nombre de b's a la pila.

Ara bé, que els 2DPDA puguin reconèixer llenguatges que no són CFL no significa que reconeguin tots els que sí són CFL. Aquest és, per ara, un problema obert. No es coneix cap CFL que no pugui ser reconegut per algun 2DPDA, però tampoc no s'ha pogut demostrar que qualsevol CFL és reconegut per algun 2DPDA. Per cert, l'algorisme de Cook citat a la introducció, que simula en temps lineal qualsevol 2DPDA, permetria d'establir la linealitat en temps de la complexitat dels CFL en el cas que se'n demostrés la possibilitat de ser reconeguts per 2DPDA. Ara com ara, els millors algorismes de què es disposa per reconèixer CFLfuncionen en temps $O(n^{2.8})$, essent n la longitud de l'entrada.

Finalment, la consideració dels *autòmats amb pila indeterministes bidireccionals* (per als quals utilitzarem l'acrònim 2NPDA) té un interès merament teòric. És obvi que representen una generalització tant dels 2DPDA com dels NPDAunidireccionals. El fet que els NPDA siguin casos particulars dels 2NPDA implica que els CFL formen una subfamília de la dels llenguatges reconeguts per 2NPDA. I com que tot 2DPDA és un cas particular d'un 2NPDA, de nou el llenguatge de l'exemple 9.7 posa de manifest que aquesta inclusió és estricta, és a dir, que els 2NPDA reconeixen llenguatges que no són CFL. Finalment, torna a ser un problema obert saber si també és estricta la inclusió de la família de llenguatges reconeguts per 2DPDA en la dels reconeguts per 2NPDA. És a dir, no es coneix cap llenguatge que sigui reconegut per un 2NPDA i que no ho sigui per algun 2DPDA. I tampoc no se sap demostrar que tot 2NPDA pugui ser simulat per algun 2DPDA.

Exercicis

9.1 Sigui L un llenguatge regular per al qual es disposa d'un DFA unidireccional que el reconeix. Doneu les idees bàsiques per construir, a partir d'aquest autòmat, un 2DFA per a cada un dels llenguatges següents:

1. $\{w \mid ww^{\mathrm{R}}ww^{\mathrm{R}} \in L\}$
2. $\{w \mid www^{\mathrm{R}}w^{\mathrm{R}} \in L\}$
3. $\{w \mid ww^{\mathrm{R}}w^{\mathrm{R}}w \in L\}$
4. $\{w \mid ww^{\mathrm{R}}ww \in L\}$

9.2 Sigui L un llenguatge regular. Demostreu, via consideració d'un 2DFA per a cada cas, que els llenguatges següents també són regulars:

1. $\{x \mid \exists y \ \ |x| = |y| \wedge xy \in L\}$
2. $\{x \mid \exists y \ \ x = y^{\mathrm{R}} \wedge xy \in L\}$
3. $\{x \mid \exists y \ \ x = y \wedge xy \in L\}$

9.3 Demostreu que el nombre de seqüències de creuaments vàlides d'un 2DFA de n estats és

$$\sum_{i=0}^{n-1} \left(\frac{n!}{(n-i)!} \right)^2 (n-i).$$

9.4 Trobeu 2DPDA que reconeguin cada un dels llenguatges següents. Es considera que Σ pot ser qualsevol alfabet de més d'un símbol i que el símbol c no pertany a Σ

1. $\{ww^{\mathrm{R}} \mid w \in \Sigma^*\}$

2. $\{ww \mid w \in \Sigma^*\}$
3. $\{xx^{\mathsf{R}}y \mid x, y \in \Sigma^+\}$
4. $\{xxy \mid x, y \in \Sigma^+\}$
5. $\{x_1c\ldots cx_n \mid x_1, \ldots, x_n \in \Sigma^* \ \wedge\ \exists i, j\ x_i = x_j^{\mathsf{R}}\}$
6. $\{x_1c\ldots cx_n \mid x_1, \ldots, x_n \in \Sigma^* \ \wedge\ \forall i \neq j\ x_i \neq x_j\}$
7. $\{xcy \mid x, y \in \Sigma^* \ \wedge\ y$ és un submot de $x\}$
8. $\{a_1a_2\ldots a_n\#b_n\ldots b_2b_1\#c_n\ldots c_2c_1 \ \mid\ n \geqslant 1 \ \wedge\ \forall i : 1 \leqslant i \leqslant n : (a_i = b_i \ \vee\ b_i = c_i)\}$
9. $\{xxy \mid x \in \mathsf{D}_1' \ \wedge\ y \in (a+b)^*\}$[4]
10. $\{a^{2^n} \mid n \geqslant 0\}$
11. $\{c^n x^k \mid n \geqslant 0 \ \wedge\ k \geqslant 2 \ \wedge\ x \in \Sigma^n\}$
12. $\{x^k \mid k \geqslant 2 \ \wedge\ x \in \Sigma^*\}$

[4] D_1' és el llenguatge de Dyck definit a l'exercici 2.15

Capítol 10 Sinopsi del curs

10.1 Introducció

Aquest capítol té com a objectiu principal donar una visió de conjunt de les idees exposades en aquest curs. Es tracta bàsicament de resumir la relació mútua entre les diferents famílies de llenguatges estudiades. Hem aprofitat aquesta visió sinòptica per presentar de manera molt resumida dues altres famílies de llenguatges que constitueixen, juntament amb les que ja coneixem, un catàleg conegut amb el nom de jerarquia de Chomsky.

Començarem, tanmateix, fent èmfasi en la relació entre gramàtiques i autòmats. D'entrada, convé precisar el diferent paper que tenen les gramàtiques i els autòmats en la definició dels llenguatges formals. Els autòmats són models de *reconeixedors* de llenguatges. Les gramàtiques són models de *generadors* de llenguatges. És clar que ambdós tipus de models tenen en comú el fet de servir de *definidors* de llenguatges. Però la manera com ho fan és essencialment diferent.

Els autòmats, com tots els reconeixedors, són algorismes que realitzen processaments efectius de les seves entrades. Naturalment, això només s'aplica als autòmats deterministes. En aquest sentit, el procés d'acceptar un mot és només un cas particularment senzill de càlcul algorísmic. Per tal de parlar en termes més generals convé tornar al plantejament introductori fet al capítol primer. Recordem que hem interpretat el concepte de *mots* com a codificacions d'uns elements que solen ser estructures de dades o, més generalment, estructures combinatòries. Quan ens restringim a problemes decisionals, el càlcul es redueix a l'acceptació o la no-acceptació del mot d'entrada, i el conjunt de mots acceptats constitueix un llenguatge. Ara bé, donada una classe d'estructures combinatòries, com per exemple la família dels grafs finits, a més de problemes decisionals, com pot ser el fet de saber si un graf donat conté algun cicle hamiltonià, poden considerar-se problemes no decisionals, com pot ser el fet de saber quina és la longitud màxima dels cicles continguts en un graf donat. L'estudi dels autòmats com a reconeixedors de llenguatges constitueix, doncs, un plantejament simplificat del problema general de l'estudi de la complexitat dels algorismes.

Per la seva banda, les gramàtiques, en tant que models generadors de llenguatges, també poden ser considerades com a casos particulars de sistemes més generals de generadors d'estructures combinatòries. Exemples típics d'operacions entre classes d'estructures combinatòries, que *generen* estructures complexes a partir d'estructures més simples, són la reunió, el producte cartesià, la diagonalització, la seqüenciació, el conjunt de parts finites, el multiconjunt de parts finites, el cicle, el marcatge o la substitució. En aquest context, és possible estendre el concepte de *recursivitat* de les gramàtiques a l'àmbit de *definicions recursives* de classes d'estructures combinatòries. El lector pot trobar un desenvolupament d'aquest tema a l'obra de Sedgewick i Flajolet [SeF96]. Veurem a continuació un exemple de generació diferent al de les gramàtiques incontextuals.

Exemple 10.1 A l'exemple 1.11 hem introduït una manera de codificar els arbres binaris i a l'exemple 2.22 hem donat una gramàtica que genera el conjunt de totes aquestes codificacions dels arbres binaris. Així doncs, aquest conjunt és un llenguatge incontextual. D'altra banda, a l'exemple 7.8 hem introduït els arbres binaris complets i hem demostrat que el conjunt de codificacions que corresponen a aquests arbres constitueix un llenguatge, inclòs en l'anterior, que no és regular. És fàcil constatar que la mateixa argumentació utilitzada a l'exemple 7.8 serveix per demostrar que el llenguatge considerat tampoc no és incontextual, ja que no hi ha cap diferència entre les dues versions dels lemes de bombament de Bar-Hillel quan s'apliquen a llenguatges uniliterals.

Tot i no tractar-se d'un llenguatge incontextual, hem vist que el llenguatge L dels arbres binaris complets admet la definició recursiva següent:

$$L = \{0\} \cup \{2xx \mid x \in L\}.$$

Hauríem pogut escriure aquesta definició recursiva de L utilitzant les operacions de reunió, producte cartesià i diagonalització de la manera següent:

$$L = \mathsf{reunió}\,(0\,,\mathsf{producte}\,(2\,,\mathsf{diagonal}(L))),$$

en què les definicions precises de les operacions considerades es poden trobar a la referència [SeF96] citada anteriorment. Es tracta, en aquest cas, tanmateix, d'operacions molt elementals el significat de les quals és força evident. L'avantatge d'aquesta segona definició és que posa clarament de manifest el seu caràcter generativorecursiu.

Un àmbit en el qual s'interrelacionen els algorismes (dels quals els autòmats en són models abstractes) i els sistemes generatius (entre els quals hi ha les gramàtiques) el constitueix el de l'*anàlisi d'algorismes* i, més en particular, el de l'estudi de la complexitat mitjana d'algorismes. El fet rellevant és que l'anàlisi de la complexitat d'un algorisme que opera sobre estructures combinatòries d'una classe determinada es veu enormement facilitat quan és possible generar recursivament aquesta classe d'estructures. De nou remetem el lector a l'obra de Sedgewick i Flajolet citada anteriorment.

10.2 Relació entre les famílies de llenguatges estudiades

A la figura 10.1 s'ha dibuixat un diagrama d'Euler que inclou les famílies de llenguatges estudiades al llarg d'aquest curs. A cada regió del diagrama figura un número que correspon a un exemple de llenguatge que pertany a la regió considerada. La definició d'aquests exemples es fa més endavant en aquesta mateixa secció. La zona ombrejada correspon a una regió que no se sap si és buida, com ja s'ha fet constar al capítol 9, a la secció dedicada als autòmats amb pila bidireccionals.

Convé remarcar algunes característiques d'aquest diagrama que corresponen a propietats de les famílies estudiades. (Cal assenyalar que el nom d'una família, precedit del

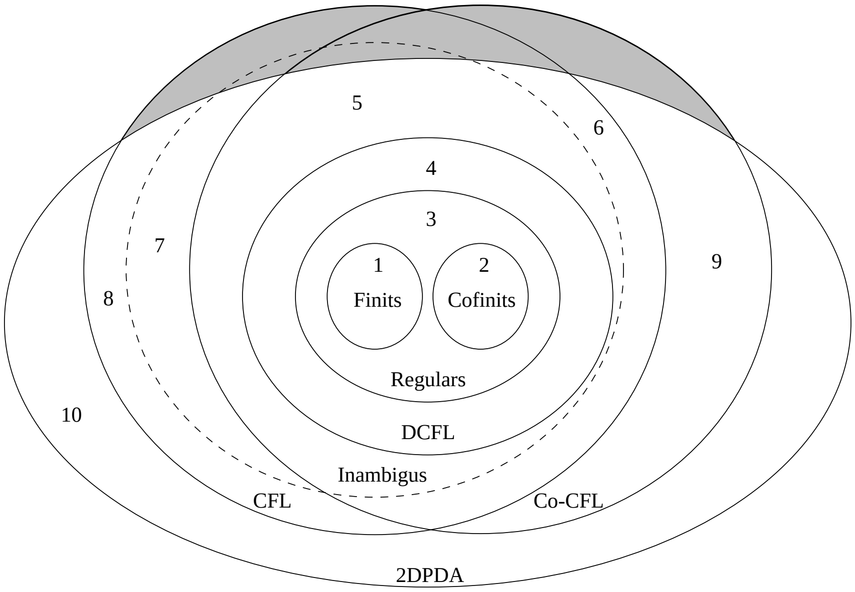

Fig. 10.1 *Diagrama d'Euler d'algunes famílies de llenguatges*

prefix '*co*', designa la família formada pels complementaris dels llenguatges de la família considerada.)

a) Cap llenguatge no pot ser alhora finit i cofinit.
b) Com que la família dels DCFL és tancada respecte de la complementació, està inclosa a la intersecció dels CFL i els co-CFL.
c) La regió que correspon als CFL inambigus ha estat dibuixada amb traç discontinu perquè, a diferència de les altres famílies considerades, no existeix cap model de descripció (ni autòmat ni gramàtica) específic d'aquesta família.
d) En el cas d'un CFL, no hi ha cap relació entre el fet de ser ambigu i el fet que el complementari també sigui CFL.

En els exemples que segueixen podem considerar, a l'efecte de l'operació de complementació, que l'alfabet de referència és $\Sigma = \{a, b, c, d, e\}$.

1. Exemple de llenguatge finit: $L_1 = \varnothing$.
2. Exemple de llenguatge cofinit: $L_2 = \Sigma^*$.
3. Exemple de llenguatge regular que no és ni finit ni cofinit: $L_3 = a^*b^*$.
4. Exemple de llenguatge incontextual determinista que no és regular:
$$L_4 = \{a^n b^n \mid n \geqslant 0\}.$$
5. Exemple de llenguatge incontextual, amb complementari també CFL, que és inambigu, però que no és determinista:
$$L_5 = \{w \in \Sigma^* \mid w = w^{\mathsf{R}}\}.$$
6. Exemple de llenguatge incontextual, amb complementari també CFL, però que és ambigu (tret de [HiU66]):
$$L_6 = \{a^p b^q c^r d^s e^t \mid (p = q \wedge r = s) \vee (q = r \wedge s = t)\}.$$
7. Exemple de llenguatge incontextual que és inambigu, però el complementari del qual no és CFL (tret de [HiU66]):

$$L_7 = \{a^p b^q c^r d^s \mid ((10p < q < 12p \ \vee \ 10q < p < 12q) \\ \wedge (10r < s < 12r \ \vee \ 10s < r < 12s)) \\ \vee (10q < r < 12q \ \wedge \ 6p < s < 8p)\}.$$

8. Exemple de llenguatge incontextual ambigu, el complementari del qual no és CFL:

$$L_8 = \{a^i b^j c^k \mid i = j \ \vee \ j = k\}.$$

9. Exemple de llenguatge que no és incontextual, però el complementari del qual sí que ho és:

$$L_9 = \{a^n b^n c^n \mid n \geqslant 0\}.$$

10. Exemple de llenguatge que no és incontextual, amb complementari que tampoc no ho és, però que és reconegut per un 2DPDA:

$$L_{10} = \{a^i b^j c^k \mid i = j \neq k \ \vee \ i \neq j = k\}.$$

10.3 La jerarquia de Chomsky

Les dues famílies principals de llenguatges estudiades en aquest curs, la dels regulars i la dels incontextuals, constitueixen les dues classes més simples d'una classificació de quatre nivells definida per N. Chomsky a [Cho56] i [Cho59]. Aquesta classificació ha perdut, amb el pas del temps, gran part del sentit ordenador que tenia inicialment. La raó d'això ha estat el desenvolupament que ha experimentat a partir dels anys setanta la teoria de la complexitat, que ha permès donar una tractament molt més ben fonamentat i d'un abast molt més general a l'estudi de la complexitat inherent dels problemes de càlcul. El lector pot trobar desenrotllat aquest plantejament a la referència [BDG95]. De tota manera, les referències a la jerarquia de Chomsky continuen essent freqüents en l'àmbit de la teoria de llenguatges formals. És per això que a continuació en fem una exposició breu. Si bé cada nivell de la jerarquia pot ser caracteritzat simultàniament en termes de gramàtiques i d'autòmats, aquí només donarem amb precisió les definicions dels nous models de gramàtiques, que són les més senzilles, i només farem una descripció informal dels nous tipus d'autòmats, que ja s'escapen de l'abast d'aquest curs.

1. Llenguatges de tipus 3. Es tracta dels llenguatges regulars. Els seus reconeixedors són els autòmats finits i els seus generadors són les gramàtiques regulars estudiades al capítol 6.

2. Llenguatges de tipus 2. Es tracta dels llenguatges incontextuals. Els seus reconeixedors són els autòmats amb pila indeterministes i els seus generadors són les gramàtiques incontextuals estudiades al capítol 2.

3. Llenguatges de tipus 1. Es tracta dels anomenats *llenguatges contextuals* (en anglès, *context-sensitive languages*, d'on deriva l'acrònim CSL). Els seus reconeixedors són els *autòmats amb espai fitat linealment* (*linear-bounded automata* en anglès, amb acrònim LBA). Es tracta de les màquines de Turing indeterministes, de què parlarem al punt següent, a les quals s'imposa la restricció de no poder utilitzar per al càlcul més espai de cinta que el que conté l'entrada. Quant als seus generadors, es tracta d'una extensió del concepte de gramàtica incontextual, que donem a continuació.

Una *gramàtica contextual* és una estructura de la forma

$$G = \langle V, \Sigma, P, S\rangle$$

en què V, Σ i S tenen la mateixa definició com a alfabet de variables, alfabet de terminals i variable inicial que en les CFGs. V i Σ són conjunts disjunts. P és un subconjunt finit de parells de

$$((V \cup \Sigma)^* - \Sigma^*) \times (V \cup \Sigma)^*$$

els elements del qual, anomenats també aquí produccions, s'escriuen de la forma $\alpha \to \beta$, amb $\alpha \in (V \cup \Sigma)^* - \Sigma^*$ i $\beta \in (V \cup \Sigma)^*$ i estan subjectes a les restriccions següents:

1. Si $\beta \neq \lambda$, aleshores $|\alpha| \leqslant |\beta|$.
2. Si $\beta = \lambda$, aleshores $\alpha = S$, i en aquest cas S no pot aparèixer al costat dret de cap producció.

La manera com una gramàtica contextual genera mots és similar a la manera com ho fa una gramàtica incontextual. Es diu que una cadena $\gamma\beta\delta$ deriva directament d'una cadena $\gamma\alpha\delta$ quan $(\alpha \to \beta) \in P$. Anàlogament, es defineix la derivació $\alpha \overset{*}{\Rightarrow} \beta$ com el tancament reflexiu i transitiu de la derivació directa. En aquestes condicions, el llenguatge generat per una gramàtica G continua essent

$$\mathsf{L}(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}.$$

Exemple 10.2 Considerem de nou el llenguatge $L = \{a^n b^n c^n \mid n \geqslant 0\}$, que a l'exemple 7.0 hem demostrat que no és incontextual. Es tracta, en canvi, d'un llenguatge contextual. No és difícil comprovar que és generat per la gramàtica

$$G = \langle \{S, A, B, C\}, \{a, b, c\}, P, S\rangle$$

que té el conjunt P de produccions següent:

$$\begin{aligned} S &\to A \mid \lambda \\ A &\to aAB \mid C \\ CB &\to bCc \mid bc \\ cB &\to Bc\,. \end{aligned}$$

Així, el mot *aabbcc* és derivat de la manera següent

$$S \Rightarrow A \Rightarrow aAB \Rightarrow aaABB \Rightarrow aaCBB \Rightarrow aabCcB \Rightarrow aabCBc \Rightarrow aabbcc.$$

4. Llenguatges de tipus 0. Es tracta dels llenguatges anomenats *enumerables recursivament*. Els seus reconeixedors són les *màquines de Turing*. Es tracta, succintament, d'autòmats de nombre finit d'estats, amb una cinta d'entrada infinita que inicialment està tota en blanc a excepció de la zona que conté el mot d'entrada, dotats d'un capçal bidireccional, i que, a més de la capacitat de llegir els símbols de la cinta, tenen també la capacitat d'escriure a la mateixa cinta.[1]

També per a aquests llenguatges existeix un tipus de gramàtiques que els generen. Es tracta de les gramàtiques de tipus 0 o *gramàtiques sense restriccions*. Una gramàtica de tipus 0 és una estructura de la forma $G = \langle V, \Sigma, P, S\rangle$ en què, com sempre, V i Σ són

[1] El lector pot trobar-ne una descripció formal al capítol 3 de [SACL01].

dos alfabets disjunts anomenats de variables i terminals respectivament, S és la variable inicial i P és un subconjunt finit de parells de

$$((V \cup \Sigma)^* - \Sigma^*) \times (V \cup \Sigma)^*.$$

Es tracta, doncs, de parells del mateix tipus que els de les gramàtiques contextuals, que continuem anomenant produccions i escrivint de la forma $\alpha \to \beta$, amb $\alpha \in (V \cup \Sigma)^* - \Sigma^*$ i $\beta \in (V \cup \Sigma)^*$, però que a diferència de les produccions contextuals no estan subjectes a cap mena de restricció addicional, és a dir, que α i β poden ser cadenes qualssevol, sempre que α contingui almenys una variable.

Exemple 10.3 Per donar exemples de llenguatges enumerables recursivament i que no siguin contextuals cal apujar considerablement el llistó de la dificultat de la computació. Una manera senzilla de fer-ho seria prendre en consideració llenguatges *no recursius*, és a dir, llenguatges per als quals no hi ha cap algorisme reconeixedor que sigui d'aturada segura.[2] Hi ha, tanmateix, exemples de llenguatges "naturals" (és a dir, que no provenen de la utilitzacio de definicions artificioses, com les basades en procediments de diagonalització) que són recursius però no són contextuals. Un dels exemples més clàssics és degut a Hunt [Hun73] i es troba recollit a [AHU74]. Es tracta d'estendre la definició d'expressions regulars, de manera que incloguin l'operador intersecció. El llenguatge considerat és el conjunt d'expressions d'aquest tipus tals que cada una d'elles descriu el conjunt de tots els mots sobre el seu alfabet.

Cal remarcar el fet que les definicions de les gramàtiques de tipus 3, 2 i 1 són casos particulars de les de tipus 2, 1 i 0, respectivament. Els llenguatges corresponents a aquestes gramàtiques formen, per tant, una *jerarquia* en el sentit que cada una de les famílies considerades està inclosa en la següent. Hem donat, per a cada cas, exemples que posen de manifest que es tracta d'inclusions estrictes en tots els casos.

[2] Una definició precisa del concepte de llenguatge recursiu pot trobar-se al mateix capítol de la referència que acabem de citar.

Referències

[AHU74] ALFRED V. AHO, JOHN E. HOCPCROFT i JEFFREY D. ULLMAN, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, Mass., 1974.

[ASU86] ALFRED V. AHO, RAVI SETHI i JEFFREY D. ULLMAN, *Compilers: Principles, Techniques and Tools*, Addison-Wesley, Reading, Mass., 1986.

[Ard60] D. N. ARDEN, *Delayed logic and finite state machines*, Theory of Computing Machine Design, Univ. of Michigan Press, Ann Arbor, Mich., 1960, pàg. 1–35.

[BDG95] JOSÉ L. BALCÁZAR, JOSEP DÍAZ i JOAQUIM GABARRÓ, *Structural Complexity I*, Springer-Verlag, Berlin Heidelberg, (2a ed.) 1995.

[BPS61] Y. BAR-HILLEL, M. PERLES i E. SHAMIR, *On formal properties of simple phrase-structure grammars*, Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung **14** (1961), pàg. 143–172.

[BeM97] JOHAN VAN BENTHEM i ALICE TER MEULEN, *Handbook of logic and language*, Elsevier Science B.V., Amsterdam, 1997.

[Ber79] JEAN BERSTEL, *Transductions and Context-Free Languages*, B. G. Teubner, Stuttgart, 1979.

[Cho56] NOAM CHOMSKY, *Three models for the description of language*, IRE Trans. on Information Theory **2** núm. 3 (1956), pàg. 113–124.

[Cho59] NOAM CHOMSKY, *On certain formal properties of grammars*, Information and Control **2** núm. 2 (1959), pàg. 137–167.

[CFS94] F. COMELLAS, J. FÀBREGA, A. SÀNCHEZ i O. SERRA, *Matemàtica discreta*, Edicions UPC, Barcelona, 1994.

[Coo71] STEPHEN A. COOK, *Linear time simulation of deterministic two-way pushdown automata*, Proc. IFIP Congress, vol. 71, North-Holland, Amsterdam, 1971, pàg. 172–179.

[Fra93] XAVIER FRANCH GUTIÉRREZ, *Estructura de dades. Especificació, disseny i implementació*, Edicions UPC, Barcelona, 1993.

[HiU66] THOMAS N. HIBBARD i JOSEPH ULLIAN, *The independence of inherent ambiguity from complementedness among context-free languages*, Journal of the ACM **13** núm. 4 (1966), pàg. 588–593.

[HMU01] JOHN E. HOPCROFT, RAJEEV MOTWANI i JEFFREY D. ULLMAN, *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading, Mass., 2001.

[HoU79] JOHN E. HOPCROFT i JEFFREY D. ULLMAN, *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading, Mass., 1979.

[Hun73] H. B. HUNT, *The equivalence problem for regular expressions with intersection is not polynomial in tape*, Tech. Report TR 73-156, Dept. of Computer Science, Cornell University, Ithaca, N.Y. (1973).

[Kle56] STEPHEN C. KLEENE, *Representation of events in nerve nets and finite automata*, Automata Studies (C. E. Shanon i J. McCarthy, eds.), Princeton Univ. Press, Princeton, N.J., 1956, pàg. 3–42.

[Knu68] DONALD E. KNUTH, *The art of computer programming. Volume 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass., 1968.

[MNY60] R. MCNAUGHTON i H.YAMADA, *Regular expressions and state graphs for automata*, IEEE Trans. on Electronic Computers **8** (1960), pàg. 39–47, reeditat per E. F. Moore [Moo64, pàg. 157–174].

[Moo64] EDWARD F. MOORE, *Sequential Machines: Selected Papers*, Addison-Wesley, Reading, Mass., 1964.

[Ogd68] W. F. OGDEN, *A helpful result for proving inherent ambiguity*, Mathematical Systems Theory **2** núm. 3 (1968), pàg. 191–194.

[RaS59] MICHAEL O. RABIN i DANA SCOTT, *Finite automata and their decision problems*, IBM Journal of Research and Development **3** núm. 2 (1959), pàg. 114–125, reeditat per E. F. Moore [Moo64, pàg. 63–91].

[Sal69] ARTO SALOMAA, *Theory of Automata*, Pergamon Press, Oxford, 1969.

[SeF96] ROBERT SEDGEWICK i PHILIPPE FLAJOLET, *Analysis of Algorithms*, Addison-Wesley, Reading, Mass., 1996.

[Sen97] GÉRAUD SÉNIZERGUES, *The equivalence problem for Deterministic Pushdown Automata is decidable*, Proc. 24th Intern. Coll. of Automata, Languages and Programming (ICALP'97) (P. Degano, R. Gorrieri i A. Marchetti-Spaccamela, eds.), vol. LNCS 1256, Springer-Verlag, Berlin Heidelberg, 1997, pàg. 671–681.

[SACL01] M. SERNA, C. ÀLVAREZ, R. CASES i A. LOZANO, *Els límits de la computació. Indecidibilitat i NP-completesa*, Edicions UPC, Barcelona, 2001.

[She59] J. C. SHEPHERDSON, *The reduction of two-way automata to one-way automata*, IBM Journal of Research and Development **3** núm. 2 (1959), pàg. 198–200, reeditat per E. F. Moore [Moo64, pàg. 92–97].

[Wir88] NIKLAUS WIRTH, *Programming in Modula-2*, Springer-Verlag, Berlin Heidelberg, 1988.