Exemples de construcció d'autòmats finits

Material de suport del curs:

Llenguatges, gramàtiques i autòmats

Lluís Màrquez
Enrique Romero

Març de 2001

Índex

1 Introducció

Aquest document pretén servir de suport a l'estudiant de l'assignatura Gramàtiques, Llenguatges i Autòmats (LGA) que s'imparteix a la Facultat d'Informàtica de Barcelona com assignatura obligatòria de tercer quadrimestre. Conté un conjunt d'exemples resolts sobre un tema que considerem d'importància cabdal dintre del temari de l'assignatura: la construcció d'autòmats finits.

Hem organitzat el material en diferents seccions segons la temàtica dels continguts que pretenem transmetre. Cada secció consta d'una part introductòria i de motivació i de la resolució comentada de diversos exemples il·lustratius de la matèria explicada. Volem remarcar que aquests apunts posen l'èmfasi en la construcció raonada dels autòmats i no pas en la demostració formal que els autòmats construïts reconeixen efectivament els llenguatges proposats. Així mateix tampoc pretén ser un "manual" en el sentit clàssic de donar "receptes" per a resoldre problemes tipus sino que té més vocació de mostrari de problemes variats i significatius. Malgrat això, sí que posem èmfasi en l'aspecte metodològic d'atacar la construcció d'autòmats finits per a llenguatges regulars complicats a partir d'una formalització adequada del llenguatge objectiu, que permeti la descomposició del problema en subproblemes més senzills i, a partir d'aquí, aplicar els algoritmes d'operació amb autòmats per a construir la solució buscada.

Lluís Màrquez i Enrique Romero

Professors del departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona Salgado, 1–3. 08034 Barcelona.
E–mail: {lluism,eromero}@lsi.upc.es

2 Llenguatges regulars senzills

En aquesta secció veurem alguns exemples senzills de construcció de DFA's. Entenem per "senzills" que admetin una construcció directa amb certa facilitat. Deixem per a les seccions següents els exemples més difícils. Tots els exemples inclouen un primer pas de formalització del llenguatge regular. Encara que pugui semblar que la formalització presentada no té res a veure amb la posterior estratègia de construcció de l'autòmat creiem que es important adjuntar-la per dos motius: 1) per practicar la formalització de llenguatges; 2) per poder encarar la construcció també via descomposició en subproblemes (vegeu la secció següent).

Exercici 2.1 *Trobeu el* DFA *mínim corresponent al llenguatge dels mots, sobre l'alfabet* $\{a\,,b\}$*, en què tot parell de* a*'s adjacents va seguit immediatament d'un parell de* b*'s adjacents.*

El llenguatge proposat es pot descriure formalment de la manera següent:

$$L = \{w \in \{a\,,b\,,c\}^* \mid \forall x, y \in \{a\,,b\}^* \;\; w = xaay \implies (\exists z \in \{a\,,b\}^* \; y = bbz)\}\,.$$

L'autòmat que el reconegui es basarà tan sols a saber reconèixer les seqüències de símbols "aa" i "bb" en l'ordre adequat. Fixeu-vos que només haurà de comprovar que hi ha una seqüència de dues b's consecutives cada vegada que detecti dues a's consecutives. Noteu que λ, a, b i tots els mots que no continguin dues a's consecutives pertanyen trivialment al llenguatge L. Seguint aquesta idea, es pot construir directament el DFA de la figura 2.1.

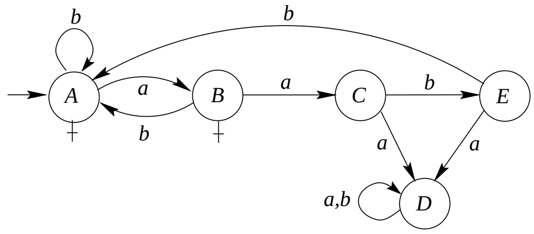


Fig. 2.1 DFA *mínim que reconeix el llenguatge L de l'exemple 2.1*

El funcionament de l'autòmat és prou senzill. Els estats finals A i B són els encarregats de reconèixer el patró aa. Així, mentre no es trobin dues a's seguides el càlcul es manté en els estats A i B i, per tant, el mot d'entrada és acceptat. Quan es detecta el submot aa el DFA passa a l'estat C, no final, i espera llegir dues b's consecutives per tornar a l'estat final A –passant per l'estat intermedi E que és l'encarregat de comptar la primera b—. Si el mot no conté aquestes dues b's consecutives a partir de l'estat C

aleshores serà rebutjat. Això pot passar en el cas que aparegui una a intermèdia o si el mot s'acaba en aquest moment. En el primer dels casos el DFA evoluciona cap a l'estat pou D, ja sigui des de C o des de E, i rebutja el mot. En el segon cas, el càlcul acabarà a l'estat C o E, que són no acceptadors.

Comprovem, ara, que el DFA sigui mínim. Efectivament,

$$\begin{aligned} \mathsf{E}_M^0 &: \ \{A,B\},\{C,D,E\} \\ \mathsf{E}_M^1 &: \ \{A\},\{B\},\{C,D\},\{E\} \\ \mathsf{E}_M^2 &: \ \{A\},\{B\},\{C\},\{D\},\{E\} \\ \mathsf{E}_M^3 &= \ \mathsf{E}_M^2. \end{aligned}$$

Exercici 2.2 *Trobeu el* DFA *mínim corresponent al llenguatge dels mots sobre l'alfabet $\{a\,,b\,,c\}$ en què tota a va seguida —no necessàriament immediatament— d'almenys dues b's —no necessàriament consecutives—.*

Una possible formalització directa del llenguatge proposat en aquest apartat es la següent:

$$L = \{w \in \{a\,,b\,,c\}^* \mid \forall x, y \in \{a\,,b\,,c\}^* \ \ w = xay \implies |y|_b \geqslant 2\}\,.$$

Adoneu-vos que, tanmateix, el llenguatge es pot enunciar d'una manera més senzilla, ja que el parell de b's que hi ha d'haver darrera de l'última a "val" per a totes les a's anteriors. Així, el llenguatge L està format pels mots que, o bé no tenen a's, o bé després de la darrera a contenen, com a mínim, dues b's. Formalment,

$$L = \{b\,,c\}^* \cup \{xay \mid x \in \{a\,,b\,,c\}^* \ \wedge \ y \in \{b\,,c\}^* \ \wedge \ |y|_b \geqslant 2\}\,,$$

o, alternativament,

$$L = \{b\,,c\}^* \cup \{a\,,b\,,c\}^* a \{b\,,c\}^* b \{b\,,c\}^* b \{b\,,c\}^*\,,$$

que és equivalent a

$$L = \{b\,,c\}^* \cup \{a\,,b\,,c\}^* ac^*bc^*b\{b\,,c\}^*$$

i també a

$$L = \{b\,,c\}^* \cup \{a\,,b\,,c\}^* a\{b\,,c\}^* bc^*bc^*\,.$$

Seguint aquesta idea, podem construir directament un DFA molt senzill que reconegui L, a base d'identificar quina és l'última a del mot i assegurar-se que és seguida d'almenys dues b's. Aquest autòmat està descrit a la figura 2.2.

El funcionament de l'autòmat és el següent. Té un sol estat acceptador, A, que és inicial i del qual no se surt mentre els símbols llegits siguin b's o c's. Així, aquest estat

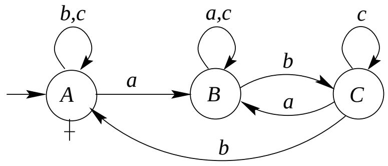


Fig. 2.2 DFA *mínim que reconeix el llenguatge L de l'exemple 2.2*

accepta trivialment els mots de $\{b\,,c\}^*$. L'estat B, que és no acceptador, recull tots els mots que contenen alguna a i tals que la darrera a no és seguida per cap b, és a dir, el llenguatge $\{a\,,b\,,c\}^*ac^*$. Si després d'aquesta darrera a el mot té com a mínim dues b's més, l'autòmat evolucionarà una altra vegada cap a l'estat acceptador A, passant primer per l'estat C que compta la primera de les b's. Altrament, acabarà el seu càlcul a l'estat B o C i serà rebutjat. Així, l'autòmat també accepta els mots del llenguatge

$$\{xay \mid x\in\{a\,,b\,,c\}^* \ \wedge\ y\in\{b\,,c\}^* \ \wedge\ |y|_b \geqslant 2\}$$

Com que no hi ha cap altra manera d'accedir a l'estat final A, ja tenim el que volíem: $L = \mathcal{L}(M)$.

Acabem comprovant que el DFA sigui mínim. Efectivament,

$$\begin{aligned}\mathsf{E}^0_M &: \ \{A\},\{B,C\}\\ \mathsf{E}^1_M &: \ \{A\},\{B\},\{C\}\\ \mathsf{E}^2_M &= \ \mathsf{E}^1_M.\end{aligned}$$

3 Construcció de DFAs per composició d'autòmats simples

En aquest apartat veurem com la descomposició del problema original en subproblemes més senzills és, sovint, una bona estratègia que ens ajuda a simplificar la construcció de l'autòmat per al llenguatge proposat. Per fer-ho, només necessitarem conèixer els algorismes de combinació d'autòmats associats a les operacions bàsiques amb llenguatges, es a dir: concatenació, reunió, intersecció, complementació, tancament de Kleene, etc.

Exercici 3.1 *Construïu el* DFA *mínim corresponent al llenguatge dels mots, sobre l'alfabet $\{a\,,b\}$, en què tot prefix de longitud superior o igual a 3 té un nombre parell de a's o un nombre parell de b's.*

Partim d'una primera formalització del llenguatge proposat.

$$L = \{w\in\{a\,,b\}^* \mid \forall x,y\in\{a\,,b\}^* \ \ w{=}xy \ \wedge\ |x| \geqslant 3 \implies (|x|_a{=}\dot{2} \ \vee\ |x|_b{=}\dot{2})\}$$

Si ens guiem per la formaització presentada, sembla difícil construir directament un DFA, que reconegui el llenguatge. D'una banda el quantificador universal obliga a comprovar, per a *cada* prefix x de longitud igual o superior a tres, que es compleix la condició $(|x|_a = \dot{2} \lor |x|_b = \dot{2})$. D'altra banda, aquesta condició sobre el comptatge d'a's i b's, tot i que és clarament regular, no és tampoc de resolució directa.

Com a primera observació, adoneu-vos que si passem al llenguatge complementari, el quantificador universal es converteix en un existencial i la connectiva condicional es converteix en conjunció:

$$\bar{L} = \{w \in \{a\,,b\}^* \mid \exists x, y \in \{a\,,b\}^* \;\; w = xy \land |x| \geqslant 3 \land |x|_a = \dot{2}+1 \land |x|_b = \dot{2}+1\}.$$

Aparentment, el problema de comprovar que cada prefix (de longitud igual o superior a tres) compleix una determinada condició regular sembla més difícil que el problema de cercar-ne *només un* que compleixi la condició contrària. Encara que en realitat es tracta d'un problema equivalent, la presentació amb el quantificador existencial resulta més senzilla i, addicionalment, ens permetrà expressar el llenguatge $\bar{L}$ en termes de la concatenació de dos llenguatges més senzills. Així, del fet que els mots de $\bar{L}$ s'han de poder partir en dos trossos x i y, on el primer compleix $(|x| \geqslant 3 \land |x|_a = \dot{2}+1 \land |x|_b = \dot{2}+1)$ i el segon és simplement qualsevol mot de $\{a\,,b\}^*$, se'n dedueix:

$$\bar{L} = \{x \in \{a\,,b\}^* \mid |x| \geqslant 3 \land |x|_a = \dot{2}+1 \land |x|_b = \dot{2}+1\} \cdot \{a\,,b\}^* = L_1 \cdot \{a\,,b\}^*$$

Amb vista a la construcció de l'autòmat, simplificarem encara més la formalització del llenguatge L_1 considerant-lo com a intersecció de tres llenguatges més senzills. Siguin els llenguatges següents:

$$L_2 = \{w \in \{a\,,b\}^* \mid |w| \geqslant 3\}\,,$$
$$L_3 = \{w \in \{a\,,b\}^* \mid |x|_a = \dot{2}+1\} \text{ i}$$
$$L_4 = \{w \in \{a\,,b\}^* \mid |x|_b = \dot{2}+1\}\,.$$

Amb aquestes definicions, és clar que $L_1 = L_2 \cap L_3 \cap L_4$ (recordeu que les conjuncions es tradueixen directament a intersecció de llenguatges) i, per tant, $\bar{L}$ es pot expressar de la manera següent:

$$\bar{L} = (L_2 \cap L_3 \cap L_4) \cdot \{a\,,b\}^*\,.$$

Tant L_2, com L_3 i L_4 són llenguatges regulars molt senzills que permeten la construcció directa de DFAs que els reconeixen. Els tres DFA's, M_2, M_3 i M_4, de la figura 3.1 reconeixen, respectivament, cadascun d'aquests llenguatges.

Comencem fent la intersecció d'aquest tres DFA's via producte cartesià d'estats. Aquesta operació està descrita, seguint la notació habitual, a la taula de transicions de la taula 3.1.

Amb vista a una major claredat, rebategem els estats de la taula anterior i els numerem des del 1 fins al 9. Aquest simple canvi de noms està reflectit a les tres

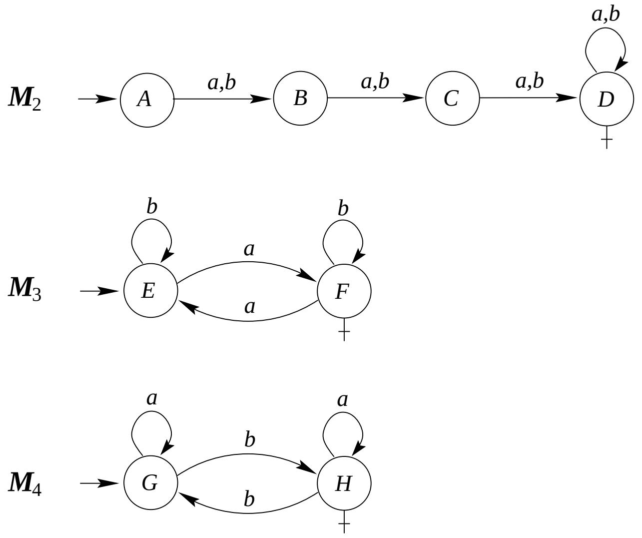


Fig. 3.1 *M_2, M_3, M_4: DFAs mínims que reconeixen els llenguatges L_2, L_3 i L_4*

primeres columnes de la taula 3.2. Les columnes 4 i 5 restants defineixen el DFA que resulta de concatenar $\{a\,,b\}^*$ a l'autòmat precedent, és a dir, defineixen un DFA per a $\bar{L}$. Adoneu-vos que l'operació de "*concatenar* Σ^*" a un NFA qualsevol és tan senzilla com substituir a tots els estats finals les transicions que ja tenien, per transicions cap a ells mateixos amb tots els símbols de l'alfabet.

En aquest cas tot queda igual a excepció de l'estat 9 —únic estat final— al qual hem substituït les seves transicions orignals per bucles cap a ell mateix amb els símbols a i b.

Ara minimitzem el DFA de la taula 3.2 i ens adonem que els estats 4 i 8 són equivalents.

$$
\begin{aligned}
\mathsf{E}^0_M &: \ \{1,2,3,4,5,6,7,8\},\{9\} \\
\mathsf{E}^1_M &: \ \{1,2,3,4,5,8\},\{6\},\{7\},\{9\} \\
\mathsf{E}^2_M &: \ \{1,2,3\},\{4,8\},\{5\},\{6\},\{7\},\{9\} \\
\mathsf{E}^3_M &: \ \{1\},\{2\},\{3\},\{4,8\},\{5\},\{6\},\{7\},\{9\} \\
\mathsf{E}^4_M &= \mathsf{E}^3_M.
\end{aligned}
$$

Fem la complementació a l'autòmat mínim per a $\bar{L}$ obtingut en el pas anterior i ja tenim un DFA mínim per a L. Recordem que passar al complementari en DFA's consisteix simplement a canviar els estats finals per no–finals i viceversa. El DFA resultant està expressat a la taula de transicions de la taula 3.3. I, equivalentment, a la figura 3.2.

Exercici. Doneu una interpretació a cadascun dels estats de l'autòmat de la figura 3.2, en termes del llenguatge dels mots que porten des de l'estat inicial a l'estat en questió.

Taula 3.1 *Intersecció dels tres* DFAs *de la figura 3.1*

	a	b
AEG	BFG	BEH
BFG	CEG	CFH
BEH	CFH	CEG
CEG	DFG	DEH
CFH	DEH	DFG
DFG	DEG	DFH
DEH	DFH	DEG
DEG	DFG	DEH
† DFH	DEH	DFG

Taula 3.2 *Intersecció dels tres* DFAs *de la figura 3.1 i concatenació amb* $\{a, b\}^*$

	a	b	a	b
1	2	3	2	3
2	4	5	4	5
3	5	4	5	4
4	6	7	6	7
5	7	6	7	6
6	8	9	8	9
7	9	8	9	8
8	6	7	6	7
† 9	7	6	9	9

Exercici. Repetiu aquest esquema de construcció d'autòmats finits per composició d'autòmats senzills per als llenguatges dels exercicis 2.1 i 2.2 (podeu partir de les formalitzacions que es donen en aquella secció). Comproveu que el resultat obtingut coincideix amb els DFA's mínims donats als mateixos exercicis.

Exercici 3.2 *Construïu el* DFA *mínim corresponent al llenguatge dels mots, sobre l'alfabet* $\Sigma = \{a, b, c\}$*, tals que si comencen per* a*, aleshores contenen el submot* aba *i si comencen per* b*, aleshores tenen un parell de* b*'s separades per un nombre parell de símbols.*

Considereu els quatre llenguatges següents:

$L_1 = a\Sigma^*$

Taula 3.3 DFA *mínim que reconeix el llenguatge L de l'exemple 3.1*

	a	b
†1	2	3
†2	4,8	5
†3	5	4,8
†4,8	6	7
†5	7	6
†6	4,8	9
†7	9	4,8
9	9	9

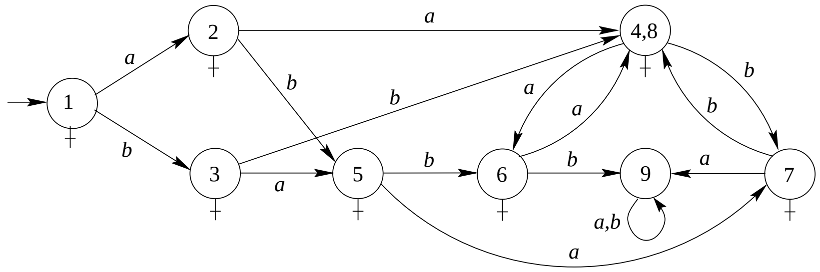


Fig. 3.2 DFA *mínim que reconeix el llenguatge L de l'exemple 3.1*

$$L_2 = \Sigma^* aba \Sigma^*$$
$$L_3 = \Sigma^* b$$
$$L_4 = \{w \in \Sigma^* \mid \exists x, y, z \in \Sigma^* \quad w = xbybz \ \wedge \ |y| = \dot{2}\}$$

que són, respectivament, els mots que comencen per a, els mots que contenen la cadena aba, els mots que acaben per b i els mots que contenen un parell de b's separades per un nombre parell de símbols. Amb aquestes definicions, el llenguatge L es pot escriure com:

$$L = \{w \in \Sigma^* \mid (w \in L_1 \implies w \in L_2) \ \wedge \ (w \in L_3 \implies w \in L_4)\}$$

Aquesta formulació és equivalent a la següent:

$$L = \{w \in \Sigma^* \mid (w \notin L_1 \ \vee \ w \in L_2) \ \wedge \ (w \notin L_3 \ \vee \ w \in L_4)\},$$

que, al seu torn, és equivalent a:

$$L = (\bar{L_1} \cup L_2) \cap (\bar{L_3} \cup L_4).$$

Aquesta darrera formalització ens dóna ja una manera clara per a construir el DFA buscat a partir dels autòmats (senzills) que reconeixen els llenguatges L_1, L_2, L_3 i L_4

i les operacions de pas al complementari, reunió i intersecció. Feu-ho.

Exercici 3.3 *Sigui L un llenguatge regular qualsevol sobre l'alfabet $\Sigma = \{a, b\}$. Doneu un mètode per construir un* DFA *corresponent al llenguatge dels mots sobre $\{a, b\}$ tals que tot sufix diferent de λ pertany al llenguatge L*

El llenguatge objectiu es pot formalizar de la manera següent:

$$M = \{w \in \Sigma^* \mid \forall x, y \in \Sigma^* \quad w = xy \ \wedge \ y \neq \lambda \implies y \in L\}.$$

Considerem ara el llenguatge complementari $\bar{M}$:

$$\bar{M} = \{w \in \Sigma^* \mid \exists x, y \in \Sigma^* \quad w = xy \ \wedge \ y \neq \lambda \ \wedge \ y \notin L\}.$$

Tal com hem fet a l'exercici 3.1, el llenguatge $\bar{M}$ es pot expressar com a concatenació de Σ^* i el llenguage dels mots y que compleixen $(y \neq \lambda \ \wedge \ y \notin L)$, és a dir:

$$\bar{M} = \Sigma^* \cdot \{y \in \Sigma^* \mid y \neq \lambda \ \wedge \ y \notin L\} = \Sigma^* \cdot (\Sigma^+ \cap \bar{L}).$$

Per tant, es té que :

$$M = \overline{\Sigma^* \cdot (\Sigma^+ \cap \bar{L})}.$$

Previ coneixement d'un DFA per al llenguatge regular L, aquesta darrera formalització ens dóna ja una manera directa de calcular el DFA per a M.

Exercici. Feu aquest exercici fins al final prenent com a definició de L el llenguatge regular definit a l'exercici 2.2.

4 Quan la descomposició en subproblemes falla

En aquesta secció veurem com, a vegades, l'estratègia de descomposar el llenguatge objectiu en d'altres subllenguatges no és adequada per arribar a la solució buscada. Això pot ser degut a que no hi hagi una descomposició natural del llenguatge en termes de les operacions bàsiques de combinació d'autòmats (exercici 4.1), o bé quan aquesta descomposició existeix però porta al cul de sac d'haver de considerar algun llenguatge no regular en el camí de la solució (exercici 4.2).

Exercici 4.1 *Construïu el* DFAs *mínim corresponent al llenguatge dels mots, sobre lálfabet $\{a, b\}$, 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.)*

El fet que el patró aba de símbols a comptar tingui un prefix no nul igual a un sufix (comença igual que acaba i, per tant, es pot encavalcar amb si mateix) fa que el llenguatge $\{w \in \{a\,,b\}^* \mid |w|_{aba} = \dot{2}\}$ no es pugui descomposar de manera natural en forma de concatenacions.

Observació. Comproveu que si, en canvi, el patró fos abb, el llenguatge resultant $L = \{w \in \{a\,,b\}^* \mid |w|_{abb} = \dot{2}\}$ es podria expressar com $\bar{A} \cdot (abb \cdot \bar{A} \cdot abb)^* \cdot \bar{A}$, on $A = \Sigma^* abb \Sigma^*$. Demostreu-ho i caracteritzeu quan això és possible.

Per resoldre el problema plantejat haurem de seguir un altre camí. Considerem el DFA M, que reconeix els mots que contenen el submot aba i que està representat a la figura figura 4.1:

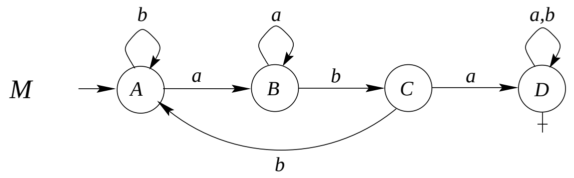


Fig. 4.1 DFA *mínim que reconeix el llenguatge* $\{w \in \{a\,,b\}^* \mid \exists x, y \in \{a\,,b\}^* \;\; w = xabay\}$

Per decidir si el submot aba apareix un nombre parell de vegades el que farem és utilitzar el DFA precedent com a comptador mòdul 2. Observeu que el comptador mòdul 2 es pot implementar amb un DFA de dos estats. Intuïtivament, necessitarem una *còpia*, M', de l'autòmat M que concatenarem a l'original seguint l'esquema de la figura 4.2.

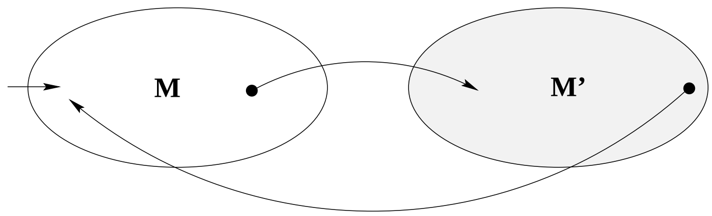


Fig. 4.2 *Esquema de la combinació dels autòmats M i M'*

Així, quan es processi un mot d'entrada, cada *volta* sencera sobre l'autòmat representarà un parell de submots aba. Caldrà posar com a acceptadors els estats de M (nombre parell de submots aba) mentre que els corresponents a M' seran no acceptadors. Resta per resoldre la qüestió dels possibles encavalcaments. La solució dependrà en cada cas del patró de símbols que estiguem considerant. En aquest cas, el patró aba és molt senzill i només pot tenir encavalcament d'un sol símbol: la darrera a del patró actual pot ser la primera del següent (en un cas general caldria mirar quin és el sufix més llarg que és alhora prefix del patró i això ens donaria el nombre de símbols d'encavalcament). Així, a l'hora de concatenar M amb M', de l'estat C amb un símbol a no cal anar a l'estat A' de M' sinó al B' que correspon a haver llegit ja el primer símbol a del següent submot a reconèixer. Això mateix passa a l'hora de tornar a lligar M' amb M: cal connectar C' amb B i no pas amb A. L'autòmat sencer està representat a la figura 4.3. Noteu que, per raons de simplicitat en el gràfic resultant, els estats corresponents a la part de M' no estan dibuixats en el mateix ordre que els de M.

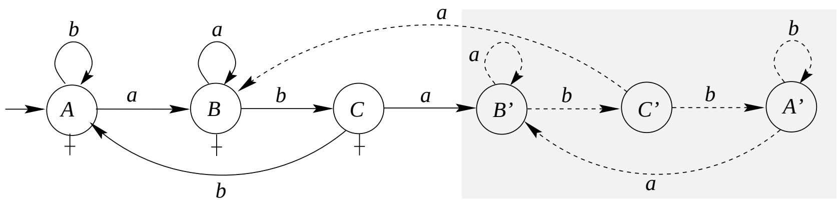


Fig. 4.3 DFA *que reconeix el llenguatge de l'exercici 4.1*

Comprovem que aquest DFA sigui mínim i ja tenim la solució buscada.

$$
\begin{aligned}
\mathsf{E}^0_M &: \ \{A,B,C\},\{B',C',A'\}\\
\mathsf{E}^1_M &: \ \{A,B\},\{C\},\{B',A'\},\{C'\}\\
\mathsf{E}^2_M &: \ \{A\},\{B\},\{C\},\{B'\},\{A'\},\{C'\}\\
\mathsf{E}^3_M &= \ \mathsf{E}^2_M .
\end{aligned}
$$

Exercici. Doneu una interpretació a cadascun dels estats de l'autòmat de la figura 4.3, en termes del llenguatge dels mots que porten des de l'estat inicial a l'estat en qüestió.

Exercici 4.2 *Construïu el* DFA*s mínim corresponent al llenguatge dels mots, sobre l'alfabet* $\{a\,,b\}$*, en què tot prefix de longitud parella té el mateix nombre de* a*'s que de* b*'s.*

El llenguatge proposat en aquest apartat té una primera definició formal molt semblant, des del punt de vista lògic, al de l'exercici 3.1. Així:

$$L = \{w \in \{a\,,b\}^* \mid \forall x,y \in \{a\,,b\}^* \ \ w = xy \ \wedge \ |x| = \dot{2} \implies |x|_a = |x|_b\}\,.$$

És a dir que cal comprovar que cada prefix de longitud parella (de longitud superior o igual a tres, abans) té el mateix nombre de a's que de b's (nombre parell de a's o b's, abans). Adoneu-vos que, a diferència del llenguatge de l'apartat 3.1, la condició que cal comprovar ara per a cada prefix de longitud parella és una condició típicament no regular (comptar el mateix nombre de a's que de b's i, en general, qualsevol proporció de a's respecte b's dóna lloc a llenguatges no regulars).

Per tant, si plantegem la mateixa descomposició en subllenguatges a partir del complementari que hem fet a l'apartat 3.1:

$$\bar{L} = \{w \in \{a\,,b\}^* \mid \exists x, y \in \{a\,,b\}^* \;\; w = xy \;\wedge\; |x| = \dot{2} \;\wedge\; |x|_a \neq |x|_b\} = (L_1 \cap L_2) \cdot \Sigma^*,$$

on $L_1 = \{w \in \{a\,,b\}^* \mid |w| = \dot{2}\}$ i $L_2 = \{w \in \{a\,,b\}^* \mid |w|_a \neq |w|_b\}$, ens trobarem amb el problema irresoluble d'haver de construir un DFA per a un llenguatge L_2 que no és regular.

Malgrat aquesta dificultat l'enunciat ens fa pensar que L sí que és regular. En conseqüència, haurem de trobar una manera alternativa d'expressar-lo que ens permeti veure que el llenguatge és més senzill del que sembla a priori. En aquest cas la senzillesa ve donada per la restricció que suposa exigir que **cada** prefix de longitud parella tingui tantes a's com b's.

Si mirem quins són els mots que pertanyen a L ens adonarem de quina és la seva estructura i veurem la manera de construir un DFA que els reconegui.

λ és el mot més petit de L. A continuació, tant a com b són mots de L ja que no tenen cap altre prefix de longitud parella que λ. Dels mots de longitud dos, només ab i ba són de L, ja que el segon símbol ve determinat pel primer —si el prefix de longitud 2 ha de tenir tantes a's com b's aleshores si el mot comença per a el segon símbol ha de ser una b i viceversa—. Els mots de longitud 3 de L es formen sobre els mots de longitud 2 i són: aba, abb, baa i bab. El tercer símbol és lliure ja que els mots de longitud 3 tenen exactament els mateixos prefixos de longitud parella que els mots de longitud 2. En canvi, el quart símbol torna venir determinat pel tercer i així els mots de longitud 4 són també quatre: $abab$, $abba$, $baab$ i $baba$.

La llei general és que els símbols en posicions senars els podem triar, mentre que els símbols en posició parella són determinats (just el contrari) pel símbol en la posició anterior. L'arbre de prefixos de la figura 4.4 reflecteix tots els mots de L fins a una longitud de 8 símbols.

És prou evident, ara, que els mots de L són els que es formen concatenant un nombre arbitrari de vegades els mots ab i ba i acabant-los, eventualment, amb un nou símbol a o b per tenir els mots de longitud senar. Expressat de manera senzilla:

$$L = \{ab, ba\}^* \cdot \{\lambda, a, b\}\,.$$

L és doncs un llenguatge regular i podem construir directament un senzill DFA de quatre estats que el reconegui. Aquest DFA està expressat a la figura 4.5.

Comprovem, finalment, que sigui mínim. Efectivament,

$$\begin{aligned} \mathsf{E}^0_M &: \; \{A, B, C\}, \{D\} \\ \mathsf{E}^1_M &: \; \{A\}, \{B\}, \{C\}, \{D\} \\ \mathsf{E}^2_M &: \; = \equiv_1 \mathsf{E}^1_M. \end{aligned}$$

$$
\lambda \begin{cases}
a-b \begin{cases}
a-b \begin{cases}
a-b \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases} \\
b-a \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases}
\end{cases} \\
b-a \begin{cases}
a-b \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases} \\
b-a \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases}
\end{cases}
\end{cases} \\
b-a \begin{cases}
a-b \begin{cases}
a-b \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases} \\
b-a \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases}
\end{cases} \\
b-a \begin{cases}
a-b \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases} \\
b-a \begin{cases} a-b \ \ldots \\ b-a \ \ldots \end{cases}
\end{cases}
\end{cases}
\end{cases}
$$

Fig. 4.4 *Estructura dels mots del llenguatge associat a l'exercici 4.2*

Exercici. Doneu una interpretació a cadascun dels estats de l'autòmat anterior, en termes del llenguatge dels mots que porten des de l'estat inicial a l'estat en questió.

Exercici. Considerem el mateix llenguatge, però sobre l'alfabet {a ,b ,c}. Quin és ara el seu DFA mínim?

Observació. Fixeu-vos que si canviem el quantificador universal per un quantificador existencial en la definició del llenguatge L d'aquest exercici (és a dir, canviem l'expressió "tot prefix" per "té un prefix"), el llenguatge resultant és no regular.

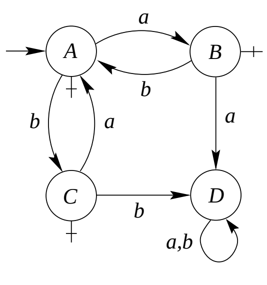


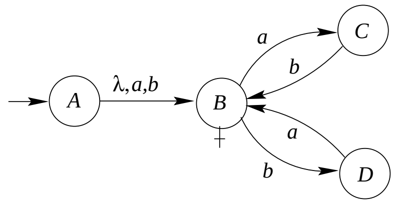
Fig. 4.5 DFA *que reconeix el llenguatge* $L = \{ab, ba\}^* \cdot \{\lambda, a, b\}$

Exercici 4.3 *Construïu el* DFAs *mínim corresponent al llenguatge dels mots, sobre l'alfabet* $\{a\,,b\}$*, en què tot sufix de longitud parella té el mateix nombre de* a*'s que de* b*'s.*

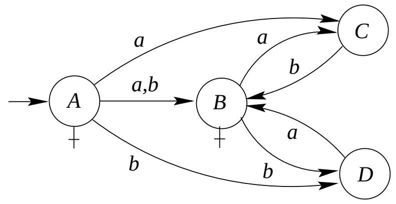
Observeu que el llenguatge proposat fa referència a la mateixa propietat (mateix nombre d'a's i b's) que el llenguatge definit en l'exercici anterior, amb l'única diferència que ara s'ha de donar per a cada *sufix* i no per a cada prefix com abans. Els mateixos raonaments que en l'exercici anterior, intercanviant prefixos per sufixos, ens portarien a una expressió senzilla per al llenguatge proposat:

$$L = \{\lambda, a, b\} \cdot \{ab, ba\}^*$$

Vist així, podem atacar directament la construcció d'un autòmat que reconegui L. Femho en dos passos. És immediat comprovar que l'autòmat següent reconeix el llenguatge $\{\lambda, a, b\} \cdot \{ab, ba\}^*$.



Un cop eliminada la λ–transició obtenim el NFA següent:



Determinitzem i minimitzem per obtenir el DFA mínim resultat. La taula següent correspon a la determinització.

	a	b
$\dagger\, A$	BC	BD
$\dagger\, BC$	C	BD
$\dagger\, BD$	BC	D
C	$\emptyset$	B
D	B	$\emptyset$
$\dagger\, B$	C	D
$\emptyset$	$\emptyset$	$\emptyset$

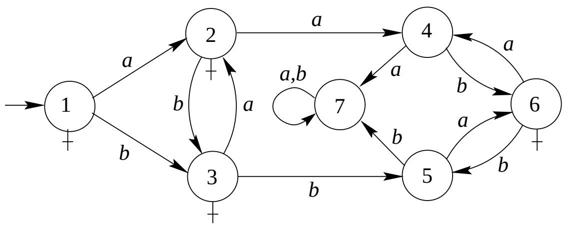
I la següent, a un rebateig dels noms dels estats per facilitar-ne la lectura.

	a	b
$\dagger\,1$	2	3
$\dagger\,2$	4	3
$\dagger\,3$	2	5
4	7	6
5	6	7
$\dagger\,6$	4	5
7	7	7

Comprovem ara que aquest autòmat és mínim.

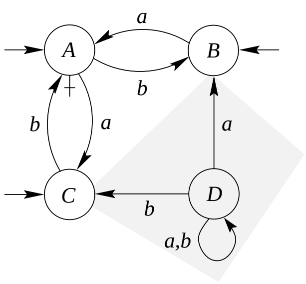
$$\mathsf{E}_M^0 : \ \{1,2,3,6\},\{4,5,7\}$$
$$\mathsf{E}_M^1 : \ \{1\},\{2\},\{3\},\{6\},\{4\},\{5\},\{7\}$$
$$\mathsf{E}_M^2 \ = \ \mathsf{E}_M^1.$$

I ja tenim l'autòmat solució que buscàvem, expressat en el diagrama de transicions següent:



Exercici. Doneu una interpretació a cadascun dels estats de l'autòmat anterior, en termes del llenguatge dels mots que porten des de l'estat inicial a l'estat en questió.

Observació. Com que el llenguatge proposat en aquest exercici és el revessat del de l'exercici anterior, també hauríem pogut plantejar la construcció de l'autòmat a partir del DFA de l'exercici anterior. Recordem que l'operació de *revessar* un autòmat consisteix senzillament a intercanviar estats inicials i acceptadors i a *invertir* el sentit de les transicions. Així l'operació de revessament sobre el DFA resultant de l'exercici anterior dóna com a resultat el NFA següent.



Observeu que la part ombrejada correspon a un estat que ha quedat inaccessible i, per tant, pot ser eliminat.

La taula següent correspon a la minimització del NFA precedent.

	a	b
$\dagger\, ABC$	AB	AC
$\dagger\, AB$	B	AC
$\dagger\, AC$	AB	C
B	$\emptyset$	A
C	A	$\emptyset$
$\dagger\, A$	B	C
$\emptyset$	$\emptyset$	$\emptyset$

Si ara fem la identificació d'estats següent:

$$ABC=1\,,\ AB=2\,,\ AC=3\,,\ B=4\,,\ C=5\,,\ A=6\ \text{ i }\ \emptyset=7$$

podrem comprovar que el resultat coincideix exactament amb el que havíem obtingut construïnt l'autòmat directament.

Observació. Tant a l'exercici 4.2 com al 4.3 hem obtingut expressions regulars pels llenguatges proposats. La construcció de DFAs a partir d'expressions regulars és l'objectiu de la secció següent.

Exercici. Trobeu el DFA mínim que reconeix el llenguatge dels mots sobre $\{a\,,b\}$ tals que contenen almenys un parell de símbols a, un parell de símbols b i tals que tenen el mateix nombre d'ocurrències del mot ab que del mot ba.

5 Construcció de DFAs a partir d'expressions regulars

Suposem que coneixem l'expressió regular d'un llenguatge i volem construir el seu DFA mínim associat. Per definició, les expressions regulars estan formades per concatenacions, reunions i estrelles de Kleene d'altres expressions regulars més simples. Com que el conjunt dels llenguatges regulars és tancat respecte d'aquestes tres operacions, la construcció de DFAs a partir d'expressions regulars és molt fàcil. Només cal construir els autòmats per a les expressions atòmiques (λ, $\emptyset$ i els símbols de l'alfabet) i després fer les operacions amb aquests autòmats segons la prioritat de les operacions que defineixen l'expressió regular. Finalment, només cal determinitzar i minimitzar.

Aquesta idea ens permet amb tota seguretat construir el DFA mínim. Només hi ha un problema: el camí pot ser molt llarg, feixuc i farragós. Si l'expressió es llarga, el nombre d'operacions serà molt gran. A més, s'ha de tenir en compte que sol ser convenient determinitzar i minimitzar els autòmats dels passos intermedis, per no trobar-se després amb un autòmat indeterminista massa gran. La pregunta, doncs, sembla òbvia. Existeix alguna manera més “hàbil” de construir un autòmat “senzill” a partir de l'expressió regular? La resposta, com ja podeu imaginar, és afirmativa.

Per veure com es podria tractar aquest problema, recordem que el llenguatge reconegut per un autòmat (tant determinista com indeterminista) es pot interpretar en funció dels camins existents a l'autòmat (vist com un graf dirigit i etiquetat) entre els estats inicials i els estats finals. Un mot pertany al llenguatge reconegut per un autòmat si i només si existeix un camí (que anomenarem camí acceptador) que surt d'algun estat inicial i arriba a algun estat final, seguint les direccions de les arestes, i consumint tots els símbols del mot segons les etiquetes del graf. Podem, doncs, construir l'autòmat de manera que tots els mots del llenguatge (i només aquests) tinguin un camí des d'un estat inicial a un estat final. Aquesta idea, que no és nova, és especialment útil quan el llenguatge el tenim expressat en forma d'expressió regular, ja que la pròpia sintaxi ens donarà una manera de construir l'autòmat. Com ja hem comentat abans, tota expressió regular, o bé és una expressió atòmica (λ, $\emptyset$ o els símbols de l'alfabet), o bé està formada per concatenacions, reunions, i estrelles de Kleene d'altres expressions regulars més simples. Veiem com es poden tractar cadascuna d'aquestes tres operacions d'una manera senzilla i, en cert sentit, òptima.

Estrella de Kleene d'una expressió regular

La idea d'aquesta construcció, que està representada a la figura 5.1, és senzilla: es tracta de fer un bucle sobre l'estat inicial amb l'expressió regular, i fer que l'estat inicial també sigui final. D'aquesta manera, es poden donar tantes voltes com es vulgui (o cap) sobre l'estat inicial, i en cadascuna d'elles es consumeix un mot de l'expressió regular.

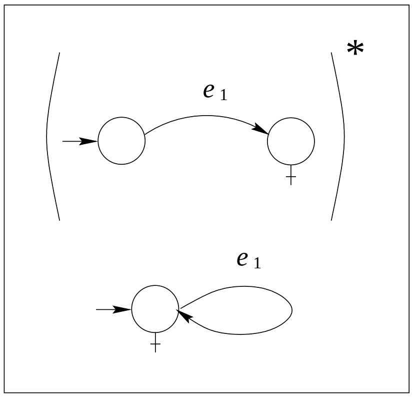


Fig. 5.1 *Esquema de l'estrella de Kleene d'una expressió regular*

Per exemple, un autòmat per a l'expressió $(bba)^*$ el podem trobar a la figura 5.2.

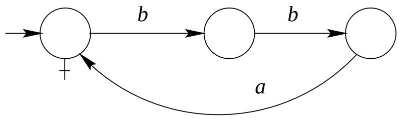


Fig. 5.2 DFA *que reconeix* $(bba)^*$

De fet, podem observar que tota estrella de Kleene està associada a algun bucle dins l'autòmat. Clarament, a l'inrevés també és veritat.

S'ha d'anar, peró, amb molta cura, doncs aquesta associació no sempre és tan senzilla, com es pot veure als dos exemples següents.

La figura 5.3 representa un autòmat per a l'expressió $(bba^*)^*$,

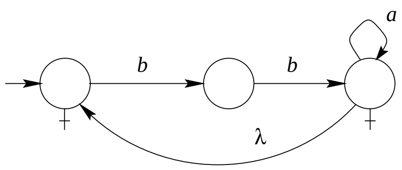


Fig. 5.3 NFA *que reconeix* (bba^*)

mentre que la figura 5.4 representen dos λ-NFAs per a $(b^*ba)^*$.

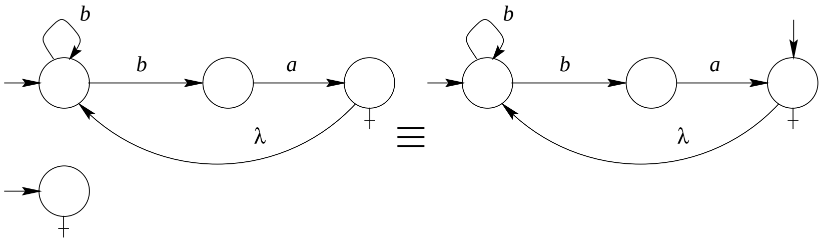


Fig. 5.4 *Dos* λ-NFAs *que reconeixen* $(b^*ba)^*$

Intentar fer un bucle sense utilitzar λ-transicions per aquests dos últims exemples no sembla immediat. Sembla clar, doncs, que podrem tenir algun problema quan o bé l'estat inicial o bé algun dels estats finals ja formi part d'un altre bucle.

Concatenació de dues expressions regulars

En aquest cas, la idea és aconseguir formar un sol camí a partir dels dos camins originals, tal com mostra esquemàticament la figura 5.5.

Per aconseguir-ho, s'ha d'intentar identificar els estats finals del primer autòmat amb els incials del segon com a l'exemple següent, on volem un autòmat per a l'expressió $(ab) \cdot (a^*ba)$:

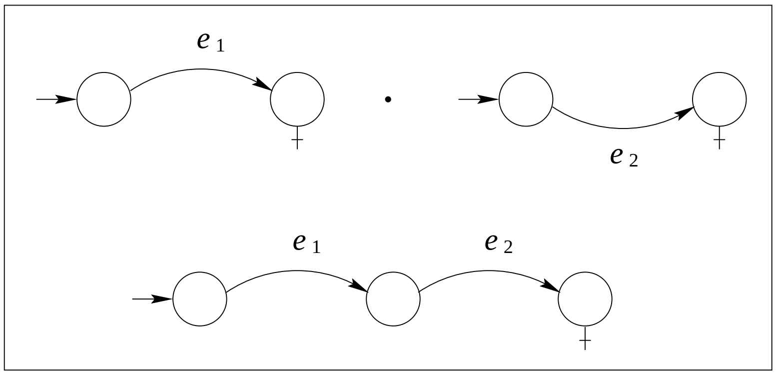
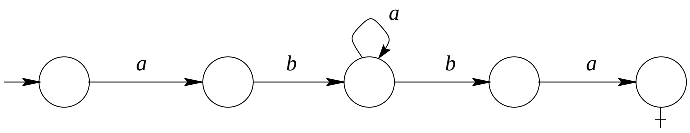
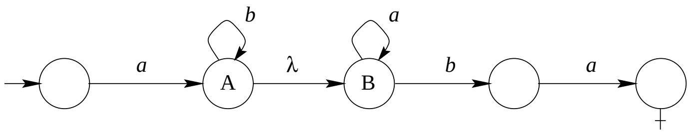


Fig. 5.5 *Esquema de concatenació de dues expressions regulars*



S'ha de tenir en compte, però, que no sempre podrem identificar els estats de manera que l'autòmat resultant sigui determinista. Això es pot veure a l'exemple següent, on volem un autòmat per a l'expressió $(ab^*) \cdot (a^*ba)$:



Intentar identificar A i B ens portaria a un autòmat incorrecte, ja que acceptaria $a\{a,b\}^*ba$.

En general, es pot veure que aquests problemes només els podem tenir quan els estats finals del primer autòmat i els estats inicials del segon autòmat formin part d'algun bucle (que com ja hem vist, està associat amb alguna estrella de Kleene).

Reunió de dues expressions regulars

Tal com il·lustra la figura 5.6, la idea és aconseguir tenir els dos camins junts al mateix autòmat, de manera que existeixin les dues alternatives (i cap més) per anar dels estats inicials als estats finals. El que no sembla massa convenient és duplicar els autòmats, connectant els estats inicials i finals, doncs l'autòmat resultant seria massa gran.

Una bona estratègia és la d'intentar reaprofitar els estats ja existents, com a l'exemple següent, on volem un autòmat per a l'expressió $aba^*ba + aaaa$:

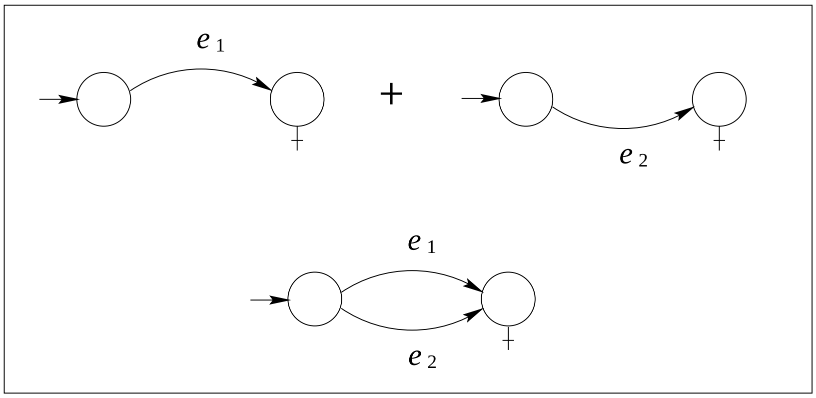
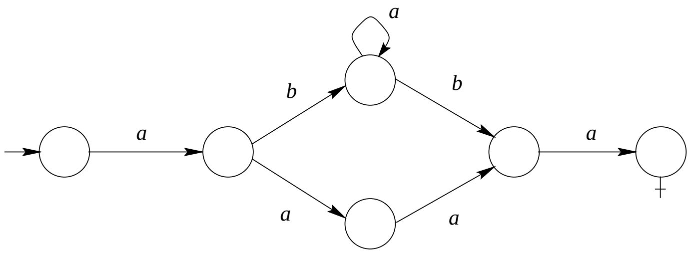
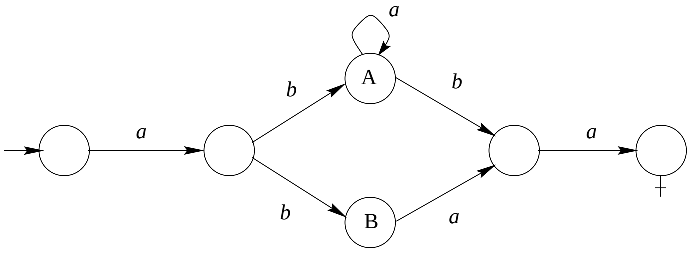


Fig. 5.6 *Esquema per a la reunió de dues expressions regulars*



S'ha de tenir en compte, però, que no sempre podrem reaprofitar els estats de manera que l'autòmat resultant sigui determinista. Això es pot veure a l'exemple següent, on volem un autòmat per a l'expressió $aba^*ba + abaa$:



Intentar reaprofitar els estats (identificant A i B) ens portaria a acceptar mots incorrectes, com qualssevol de aba^+aa.

En general, és molt fàcil veure que aquests problemes només els podem tenir quan hi ha bucles (és a dir, estrelles de Kleene) involucrats.

Estrella de Kleene de la reunió d'expressions regulars

És la generalització natural de la idea de l'estrella de Kleene. S'han de fer bucles sobre l'estat inicial (que també és final) de manera que els únics camins possibles siguin els associats a les subexpresiones regulars de l'expressió total. D'aquesta manera, es poden donar tantes voltes com es vulgui (o cap), i a cada volta consumir mots associats a les subexpressiones regulars. La idea de reaprofitar els estats (com a la reunió) segueix essent una idea molt interessant, quan sigui possible fer-ho. Aquesta idea está representada a la figura 5.7.

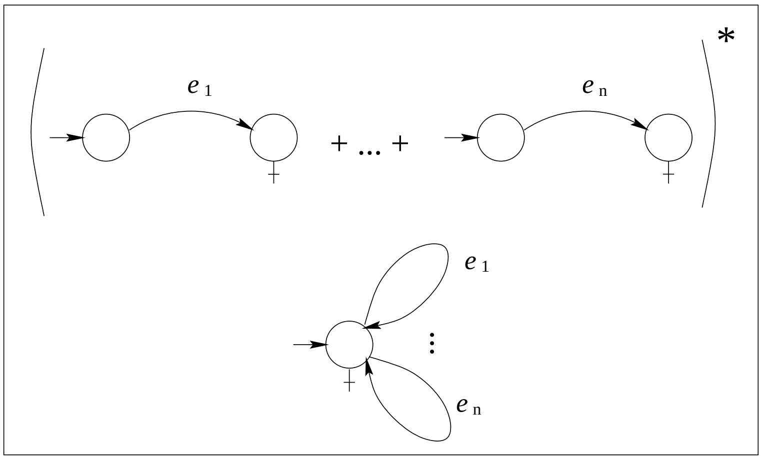


Fig. 5.7 *Esquema per a l'estrella de Kleene de la reunió d'expressions regulars*

Per exemple, un autòmat per a l'expressió $(aba^*ba + aaaa + abaa)^*$ podria ser el de la figura 5.8.

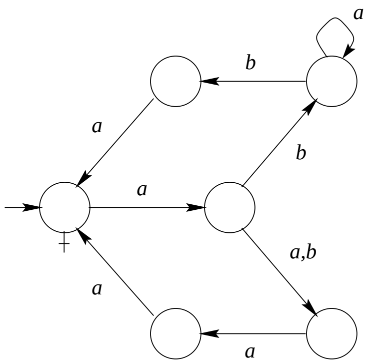
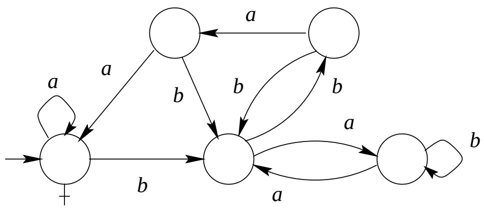


Fig. 5.8 NFA *que reconeix el llenguatge* $(aba^*ba + aaaa + abaa)^*$

Exercici 5.1 *Construïu el* DFA *mínim corresponent a l'expressió regular següent sobre l'alfabet* $\{a\,, b\}$*:* $(a + b(bb + bab + ab^*a)^*baa)^*$

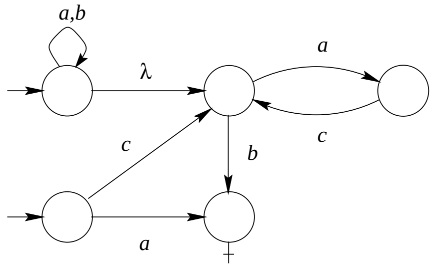
De manera immediata es pot construir el següent autòmat:



que ja és el DFA mínim (comproveu-ho).

Exercici 5.2 *Construïu el* DFA *mínim corresponent a l'expressió regular següent sobre l'alfabet* $\{a\,,b\,,c\}$*:* $((a+b)^*+c)\cdot(ac)^*b+a$

De manera bastant immediata es pot construir el següent autòmat:



La taula següent és la que correspon a l'autòmat mínim associat al λ-NFA precedent. Comproveu-ho.

	a	b	c
1	2	3	4
†2	5	3	6
†3	5	3	∅
4	6	7	∅
5	5	3	6
6	∅	∅	4
†7	∅	∅	∅
∅	∅	∅	∅

6 Altres exercicis sobre DFAs

El que s'ha vist a les seccions anteriors no pot (tampoc no ho pretén) abastar tots els possibles problemes de construcció d'autòmats que es puguin plantejar. Hi ha molts problemes que, en principi, no es poden resoldre d'una manera més o menys sistemàtica. Això no vol dir que no siguin problemes interessants. Vegem-ne alguns exemples.

Exercici 6.1 *Demostreu que si L és un llenguatge regular, aleshores també ho són els llenguatges:* $\mathsf{prefixos}(L)$, $\mathsf{sufixos}(L)$ *i* $\mathsf{factors}(L)$.

Farem una demostració constructiva. Com que L és un llenguatge regular sabem que existeix un autòmat que reconeix L. Per simplicitat, suposem que $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ és un DFA que el reconeix (podría ser el mínim o no). A partir de M construïrem autòmats que reconeguin la resta de llenguatges.

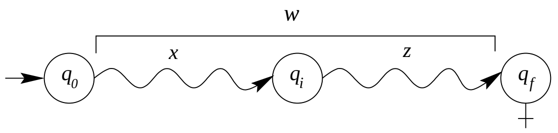
Concentrem-nos en primer lloc en el llenguatge dels prefixos de L. Com ja sabeu, les tres definicions següents són equivalents per a definir $\mathsf{prefixos}(L)$:

1) Els llenguatge format pels mots que són prefix d'algun mot de L.

2) $\{x \in \Sigma^* \mid \exists z \in \Sigma^* \quad x{\cdot}z \in L\} = \{x \in \Sigma^* \mid \exists z \in \Sigma^* \ \ \exists w \in L \quad w = x{\cdot}z\}$.

3) La reunió de tots els prefixos de tots els mots de L.

Evidentment, això inclou els propis mots de L, doncs és clar que $L \subseteq \mathsf{prefixos}(L)$. Sigui w un mot de L. Aleshores és acceptat per l'autòmat M. L'esquema següent representa el camí acceptador de w sobre l'autòmat M. L'estat q_i representa qualsevol estat intermedi en el camí de w sobre M. Això ens permet considerar el mot w com a concatenació de dos submots x i z, on x és un prefix i z és un sufixs de w.



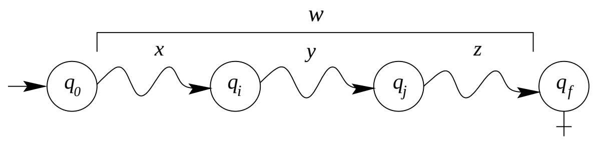
Si convertim l'estat q_i en estat final, el nou autòmat acceptarà el prefix x. Si repetim aquesta operació per a tots els estats que estiguin en el camí acceptador de w sobre M, acceptarem tots els prefixos de w. Si fem això per a tots els estats que estiguin en algun camí acceptador sobre l'autòmat, el resultat serà l'acceptació de tots els prefixos x de tots els mots de L, és a dir, $\mathsf{prefixos}(L)$. Una manera d'incloure tots aquests estats és considerar el conjunt d'estats *útils* de l'autòmat M, que passarà a ser el nou conjunt d'estats finals[1]. Un estat q és *útil* (o *coaccessible*) quan es pot arribar des de q fins a algun estat final.

Per altra banda, si el que fem és considerar com estats inicials a tots els estats q_i que es troben en les condicions del dibuix anterior, el resultat serà un autòmat que accepta

[1]De fet podríem considerar només els estats *accessibles*, però no és necessari per la demostració i d'aquesta manera resulta més senzilla.

tots els sufixos z de tots els mots de L, és a dir $\mathsf{sufixos}(L) = \{z \in \Sigma^* \mid \exists x \in \Sigma^* \;\; xz \in L\}$. Una manera d'incloure tots aquests estats és considerar el conjunt d'estats *accessibles* de l'autòmat M, que passarà a ser el nou conjunt d'estats inicials[2]. Un estat q és *accessible* quan es por arribar des d'algun estat inicial fins a q.

Finalment, el cas del submots inclou els dos darrers. Fixem-nos, en la figura següent, en el camí acceptador del mot $w \in L$ sobre l'autòmat M, partint el mot w en tres fragments qualssevols, x, y i z.



Amb vista a acceptar el submot genèric y, de w, convertirem tots els estats q_i i q_j de la figura precedent en estats inicials i finals, respectivament. Prendrem el conjunt d'estats *accessibles* en el primer cas i el conjunt d'estats *útils* en el segon[3]. D'aquesta manera l'autòmat resultant accepta $\mathsf{factors}(L)$, que, com ja sabem, es pot definir de la manera següent: $\{y \in \Sigma^* \mid \exists x, z \in \Sigma^* \;\; x \cdot y \cdot z \in L\}$.

La demostració formal de la correctesa de les construccions proposades és la següent.

Demostració. Sigui $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, un autòmat finit determinista que accepta el llenguatge L. Construïrem, a partir d'aquest, tres autòmats: M_1, M_2 i M_3, que acceptaran: $\mathsf{prefixos}(L)$, $\mathsf{sufixos}(L)$ i $\mathsf{factors}(L)$, respectivament. Aquesta demostració es pot modificar adequadament pel cas que l'autòmat de partida sigui indeterminista.

1) $M_1 = \langle Q, \Sigma, \delta, q_0, F_1 \rangle$, on $F_1 = \{q \in Q \mid \exists z \in \Sigma^* \;\; q \cdot z \in F\} = \mathsf{útils}(L)$, és un DFA tal que $\mathcal{L}(M_1) = \mathsf{prefixos}(L)$, ja que per a tot $x \in \Sigma^*$ es compleix:

$$\begin{aligned} x \in \mathcal{L}(M_1) &\iff \exists q_i \in F_1 \;\; q_0 \cdot x = q_i \\ &\iff \exists z \in \Sigma^* \;\; \exists q_i \in F_1 \;\; \exists q_f \in F \;\; q_0 \cdot x = q_i \;\wedge\; q_i \cdot z = q_f \\ &\iff \exists z \in \Sigma^* \;\; \exists q_f \in F \;\; q_0 \cdot (x \cdot z) = q_f \\ &\iff \exists z \in \Sigma^* \;\; x \cdot z \in \mathcal{L}(M) = L \\ &\iff x \in \mathsf{prefixos}(L) \end{aligned}$$

2) $M_2 = \langle Q, \Sigma, \delta, I_2, F \rangle$, on $I_2 = \{q \in Q \mid \exists x \in \Sigma^* \;\; q_0 \cdot x = q\} = \mathsf{accessibles}(L)$, és un NFA tal que $\mathcal{L}(M_2) = \mathsf{sufixos}(L)$, ja que per a tot $z \in \Sigma^*$ es compleix:

$$\begin{aligned} z \in \mathcal{L}(M_2) &\iff \exists q_i \in I_2 \;\; \exists q_f \in F \;\; q_i \cdot z = q_f \\ &\iff \exists x \in \Sigma^* \;\; \exists q_i \in I_2 \;\; \exists q_f \in F \;\; q_0 \cdot x = q_i \;\wedge\; q_i \cdot z = q_f \\ &\iff \exists x \in \Sigma^* \;\; \exists q_f \in F \;\; q_0 \cdot (x \cdot z) = q_f \\ &\iff \exists x \in \Sigma^* \;\; x \cdot z \in \mathcal{L}(M) = L \\ &\iff z \in \mathsf{sufixos}(L) \end{aligned}$$

[2]De fet podríem considerar només els estats *útils*, però no és necessari per la demostració i d'aquesta manera resulta més senzilla.

[3]Les mateixes consideracions que en les anotacions precedents són vàlides en aquest cas.

3) $M_3 = \langle Q, \Sigma, \delta, I_3, F_3 \rangle$, on $I_3 = \mathsf{accessibles}(L)$ i $F_3 = \mathsf{útils}(L)$, és un NFA tal que $\mathcal{L}(M_3) = \mathsf{factors}(L)$, ja que per a tot $y \in \Sigma^*$ es compleix:

$$\begin{aligned}
y \in \mathcal{L}(M_3) \quad &\Longleftrightarrow \quad \exists q_i \in I_3 \;\; \exists q_j \in F_3 \;\; q_i \cdot y = q_j \\
&\Longleftrightarrow \quad \exists x \in \Sigma^* \;\; \exists q_i \in I_3 \;\; \exists q_j \in F_3 \;\; q_0 \cdot x = q_i \;\wedge\; q_i \cdot y = q_j \\
&\Longleftrightarrow \quad \exists x \in \Sigma^* \;\; \exists q_j \in F_3 \;\; q_0 \cdot (x \cdot y) = q_j \\
&\Longleftrightarrow \quad \exists x, z \in \Sigma^* \;\; \exists q_f \in F \;\; \exists q_j \in F_3 \;\; q_0 \cdot (x \cdot y) = q_j \;\wedge\; q_j \cdot z = q_f \\
&\Longleftrightarrow \quad \exists x, z \in \Sigma^* \;\; \exists q_f \in F \;\; q_0 \cdot (x \cdot y \cdot z) = q_f \\
&\Longleftrightarrow \quad \exists x, z \in \Sigma^* \;\; x \cdot y \cdot z \in \mathcal{L}(M) = L \\
&\Longleftrightarrow \quad y \in \mathsf{factors}(L)
\end{aligned}$$

Exemple. Sigui $L \subseteq \{a, b\}^*$ el llenguatge format pels mots que no contenen la subcadena aba i no acaben en a. La figura 6.1 mostra el seu DFA mínim i autòmats que accepten $\mathsf{prefixos}(L)$, $\mathsf{sufixos}(L)$ i $\mathsf{factors}(L)$, respectivament.

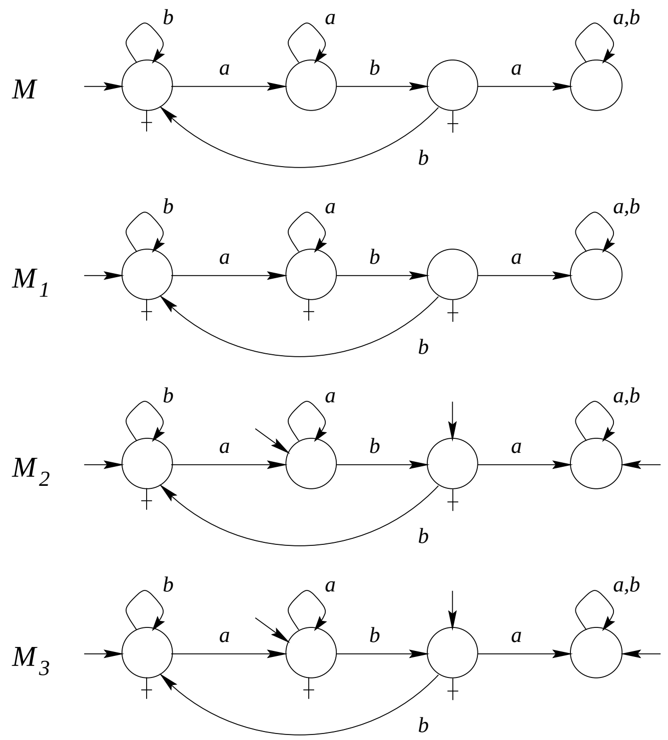


Fig. 6.1 *Autòmats associats a l'exercici 6.1*

Exercici. Trobeu els DFAs mínims per a M_1, M_2 i M_3, i una expressió dels llenguatges associats.

Exercici. Doneu algorismes per calcular els conjunts d'estats *accessibles* i *útils* d'un

NFA, de manera que sigui possible construir, **de manera efectiva**, autòmats que reconeguin els llenguatges $\mathsf{prefixos}(L)$, $\mathsf{sufixos}(L)$ i $\mathsf{factors}(L)$.

Exercici. Demostreu que si un DFA és mínim, aleshores tots els seus estats són *accessibles*, i l'únic estat que no és *útil* és el "pou no final" (en el cas que hi hagi).

Exercici. Demostreu que si L és un llenguatge regular tal que el seu DFA mínim no conté cap estat que sigui "pou no final", aleshores es compleix que $\mathsf{prefixos}(L) = \Sigma^*$. Podeu trobar alguna condició que s'hagi de complir per poder dir el mateix del llenguatge $\mathsf{sufixos}(L)$?

Observació. Com ja us heu adonat, hi ha una gran relació entre els conceptes *accessible* i *útil* que hem utilitzat aquí i els conceptes *accessible* i *fecund* definits a l'entorn de les gramàtiques incontextuals. Creieu que és una casualitat?

Exercici 6.2 *Demostreu que si L és un llenguatge regular, aleshores també ho és el llenguatge format pels mots de L que no són prefixos de cap altre mot de L.*

Una possible formalització d'aquest llenguatge és la següent:

$$L' = \{w \in L \mid \forall y \in \Sigma^+ \;\; w{\cdot}y \notin L\}.$$

Així doncs, es té que $L' = L \cap L_1$, on $L_1 = \{w \in \Sigma^* \mid \forall y \in \Sigma^+ \;\; w{\cdot}y \notin L\}$. Si existís un autòmat per a L_1, ja hauríem resolt el problema, ja que hauríem expressat L' com a intersecció de dos llenguatges regulars (i per les propietats de tancament dels llenguatges regulars, seria regular). El seu complementari $\overline{L}_1 = \{w \mid \exists y \in \Sigma^+ \;\; w{\cdot}y \in L\}$, és la reunió de tots els prefixos propis de tots els mots de L, i λ (en el cas que $L \neq \emptyset$). Tot i que no és immediat construir un autòmat per $\overline{L}_1$, ens podem inspirar en la construcció de $\mathsf{prefixos}(L)$ per fer-ho (veure enunciat anterior): partint d'un DFA M que reconeix L, els estats que s'han de convertir en estats finals han de complir dues propietats:

1) Ser *útils*, per acceptar prefixos.

2) Poder arribar a algun altre estat final (o ell mateix) amb un mot de Σ^+ (representat per y a la formalització de $\overline{L}_1$), de manera que sigui prefix *propi* d'algun mot de L (representat per $w{\cdot}y$).

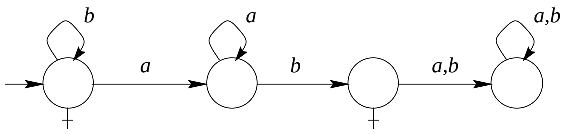
Si $L \neq \emptyset$, λ ja estarà inclòs, ja que l'estat inicial compleix les dues propietats anteriors. Una vegada construït un autòmat per $\overline{L}_1$, només cal fer el complementari, i fer la intersecció amb L.

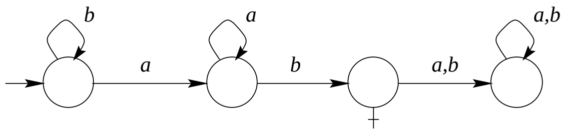
Però aquest problema es pot resoldre d'una altra manera, encara que molt similar en la seva idea principal. Clarament, $L' \subseteq L$. Com L és regular, els mots de L són acceptats per un cert DFA $M = \langle Q, \Sigma, \delta, q_0, F\rangle$. Per tant, tots els mots de L' tenen un camí acceptador a M. Però els mots de L', a més, no poden ser prefixos d'altres paraules de L. Això vol dir que el seu camí acceptador ha d'acabar en un estat (final) de M de manera que no es pugui arribar altre cop a estat final amb un mot diferent de λ. Formalment, només cal considerar el DFA $M' = \langle Q, \Sigma, \delta, q_0, F_1 \rangle$, on $F' = \{q \in F \mid \forall y \in \Sigma^+ \;\; q \cdot y \notin F\}$.

Demostració.

$$\begin{aligned}
w \in \mathcal{L}(M') &\iff \exists q_f \in F' \;\; q_0 \cdot w = q_f \\
&\iff \exists q_f \in F' \;\; (q_0 \cdot w = q_f \;\wedge\; \forall y \in \Sigma^+ \;\; q_f \cdot y \notin F) \\
&\iff \exists q_f \in F \;\; (q_0 \cdot w = q_f \;\wedge\; \forall y \in \Sigma^+ \;\; q_f \cdot y \notin F) \\
&\iff (\exists q_f \in F \;\; q_0 \cdot w = q_f) \;\wedge\; (\forall y \in \Sigma^+ \;\; q_0 \cdot w \cdot y \notin F) \\
&\iff w \in L \;\wedge\; \forall y \in \Sigma^+ \;\; w \cdot y \notin L \\
&\iff w \in L \cap L_1 = L'
\end{aligned}$$

Exemple. Sigui $L \subseteq \{a\,, b\}^*$ el llenguatge format pels mots associats a l'expressió regular $b^*(\lambda + a^+ b)$. La següent figura mostra els DFAs mínims respectius per a L i L'.





Exercici. Doneu un algorisme per calcular el conjunt d'estats F'.

Exercici. Demostreu que si L és un llenguatge regular tal que el seu DFA mínim no conté cap estat que sigui "pou no final", aleshores es compleix que $L' = \emptyset$.

Exercici 6.3 *Demostreu que si L és un llenguatge regular, aleshores també ho és el llenguatge format pels mots de L que no contenen com a prefix cap altre mot de L.*

Una possible formalització d'aquest llenguatge és la següent:

$$L' = \{w \in L \mid \forall x \in \Sigma^* \; \forall y \in \Sigma^+ \quad w = x \cdot y \implies x \notin L\}$$

Així doncs, es té que $L' = L \cap L_1$, on $L_1 = \{w \in \Sigma^* \mid \forall x \in \Sigma^* \; \forall y \in \Sigma^+ \;\; w = x \cdot y \implies x \notin L\}$. Tot i que pot semblar molt similar a l'enunciat anterior, és força diferent quant a la seva resolució, ja que $\overline{L_1}$ es pot expressar com

$$\overline{L}_1 = \{w \in \Sigma^* \mid \exists x \in \Sigma^* \; \exists y \in \Sigma^+ \;\; w = x \cdot y \;\wedge\; x \in L\} = L \cdot \Sigma^+.$$

Així doncs, $L' = L \cap L_1 = L \cap \overline{L \cdot \Sigma^+}$. Per les propietats de tancament del llenguatges regulars, és immediat demostrar que L' és regular. A més, aquesta demostració ens dóna un algorisme per a trobar L' a partir dels DFAs de L i Σ^+.