Lógica en la Informática

Josefina Sierra Santibáñez

7 de marzo de 2020

Formas normales y cláusulas

- $F_1 \wedge \ldots \wedge F_n$ es lógicamente equivalente al conjunto de fórmulas $\{F_1, \ldots, F_n\}$ asociatividad, conmutatividad e idempotencia $\wedge$.
- Literal: símbolo proposicional p (literal positivo) o símbolo proposicional negado $\neg p$ (literal negativo).
- Cláusula: disyunción de literales $l_1 \vee \ldots \vee l_n$, donde cada l_i es un literal, o equivalentemente $p_1 \vee \ldots \vee p_m \vee \neg q_1 \vee \ldots \vee \neg q_n$ donde p_i y q_j son símbolos de predicado.
- CNF (forma normal conjuntiva): conjunción de cláusulas (disyunciones de literales).

$$(l_{1,1} \vee \ldots \vee l_{1,k_1}) \wedge \ldots \wedge (l_{n,1} \vee \ldots \vee l_{n,k_n})$$

- Conjunto de cláusulas: una fórmula en CNF, ya que una conjunción de cláusulas es lógicamente equivalente a un conjunto de cláusulas.

Formas normales y cláusulas

- DNF (forma normal disyuntiva): disyunción de conjunciones de literales $(l_{1,1} \wedge \ldots \wedge l_{1,k_1}) \vee \ldots \vee (l_{n,1} \wedge \ldots \wedge l_{n,k_n})$
- Cláusula de Horn: es una cláusula que tiene como máximo un literal positivo, de la forma $p_1 \vee \neg q_1 \vee \ldots \vee \neg q_n$ o de la forma $\neg q_1 \vee \ldots \vee \neg q_n$ donde p_i y q_j son símbolos de predicado.
- Cláusula vacía: disyunción de cero literales, se denota $\square$.

Extendemos la sintáxis y semántica de la Lógica Proposicional

- Sintáxis: Si $F_1, \ldots, F_n$ son fórmulas, $\bigwedge_{i=1}^{n} F_i$ y $\bigvee_{i=1}^{n} F_i$ son fórmulas
- Semántica: la conjunción de cero fórmulas es una tautología, la disyunción de cero fórmulas es insatsifactible.

  Dada una interpretación I

  $$eval_I(\bigwedge_{i=1}^{n} F_i) = min\{1, eval_I(F_1), \ldots, eval_I(F_n)\}$$
  $$eval_I(\bigvee_{i=1}^{n} F_i) = max\{0, eval_I(F_1), \ldots, eval_I(F_n)\}$$

Formas normales y cláusulas (lema de sustitución)

- Para toda fórmula F hay al menos una fórmula lógicamente equivalente que está en CNF (**ej 1** del capítulo 3 de los apuntes).
- Para toda fórmula F hay al menos una fórmula lógicamente equivalente que está en DNF (**ej 1**).
- Dada una fórmula F se puede calcular una fórmula F^c lógicamente equivalente a F que está en CNF aplicando exhaustivamente el lema de sustitución y
  1. doble negación y leyes De Morgan
  2. distributividad

  Descripción de este algoritmo y demostración de su terminación: ejercicio 2 del capítulo 3 de los apuntes de la asignatura. Pero el tamaño de F^c puede ser exponencialmente mayor que el de F. Por ejemplo, si aplicamos este algoritmo a la fórmula
  $(p_1 \wedge q_1) \vee \ldots \vee (p_n \wedge q_n)$
  el resultado es una fórmula con 2^n cláusulas
  $(p_1 \vee p_2 \vee \ldots \vee p_n) \wedge (q_1 \vee p_2 \vee \ldots \vee p_n) \wedge \ldots \wedge (q_1 \vee q_2 \vee \ldots \vee q_n)$
  cada cláusula contiene o bien p_i o bien q_i para $i = 1, \ldots, n$.

Formas normales y cláusulas (transformación Tseitin)

- Transformación de Tseitin en CNF evita este crecimiento exponencial preservando la satisfactibilidad de la fórmula, en lugar de la equivalencia lógica.

Propiedades de la transformación Tseitin

1. El tamaño de la fórmula resultante F^c es lineal con respecto al tamaño de F.
2. El coste del algoritmo es lineal en el tamaño de F.
3. F^c es satisfactible si y sólo si F es satisfactible.
4. F^c es un conjunto de cláusulas con tres literales como máximo.

Ejemplos de aplicación de la transformación de Tseitin

- Transparencias 35 a 41 del capítulo Propositional Logic del material docente del curso Combinatorial Problem Solving del Master in Innovation and Research in Informatics. Autores: Albert Oliveras y Enric Rodríguez-Carbonell.
- Ejercicios de primer examen parcial: ej. 2 de Primavera 2018 (figura), ej. 2 de Primavera 2017, ej. 3 de Otoño 2017, ej. 2 Primavera 2015, ej. 2 de Primavera 2012 y ej. 2 Otoño 2011.

Formas normales y cláusulas (transformación Tseitin)

Transformación de Tseitin transforma una fórmula F de la lógica proposicional en una fórmula F^c en CNF equisatisfactible.

- Nueva variable z_i por cada subfórmula G_i que no sea un literal.
- Si $G_i \equiv G_k \wedge G_h$, se añaden $\{\neg z_k \vee \neg z_h \vee z_i, \neg z_i \vee z_k, \neg z_i \vee z_h\}$ a F^c, donde z_r representa la subfórmula G_r, si G_r es un literal; en otro caso, z_r es la nueva variable introducida para G_r.
- Si $G_i \equiv G_k \vee G_h$, se añaden $\{z_k \vee z_h \vee \neg z_i, z_i \vee \neg z_k, z_i \vee \neg z_h\}$ a F^c, donde z_r representa la subfórmula G_r, si G_r es un literal; en otro caso, z_r es la nueva variable introducida para G_r.
- Si $G_i \equiv \neg G_k$, y G_k no es un símbolo proposicional, se añaden $\{\neg z_i \vee \neg z_k, z_i \vee z_k\}$ a F^c, donde z_k representa la nueva variable introducida para G_k.
- Si z_0 es la variable introducida para la subfórmula G_0 igual a F, se añade una cláusula de la forma $\{z_o\}$ a F^c.

Los conjuntos de cláusulas F^c y F son equisatisfactibles.

Ejercicios

- Demuestra que una cláusula es tautología si y sólo si contiene a la vez p y $\neg p$ para algún símbolo proposicional p (**ej 5**).
- Sea S un conjunto de cláusulas que no contiene la cláusula vacía. Demuestra que S es satisfactible construyendo un modelo para S en las siguientes situaciones (**ej 6**):
  - Toda cláusula de S contiene al menos un literal positivo.
  - Toda cláusula de S contiene al menos un literal negativo.
  - Para todo símbolo proposicional p se cumple que: o bien p aparece sólo en literales positivos en S, o bien p aparece sólo en literales negativos en S.
- ¿Cuántas cláusulas distintas (como conjuntos de literales) hay?
- ¿Cuántas cláusulas insatisfactibles hay?
- ¿Cuántas cláusulas distintas que no sean tautologías hay?
- ¿Cuántas cláusulas distintas que contienen exactamente un literal por cada símbolo proposicional hay? (**ej 7**)

Ejercicios

- Sea S un conjunto de cláusulas de Horn que no contiene la cláusula vacía. Demuestra que S es satisfactible (construyendo un modelo para S) si en S no hay ninguna cláusula que conste de un único literal positivo (**ej 9**).
- Demuestra que el enunciado previo no es cierto cuando S no es de Horn (**ej 10**).
- Un literal es puro en una fórmula en CNF si aparece o bien siempre negado o bien siempre sin negar. Una cláusula es redundante si contiene al menos un literal puro. Si C' es una cláusula redundante, entonces $\{C_1, \ldots, C_n, C'\}$ es satisfactible si y sólo si $\{C_1, \ldots, C_n\}$ es satisfactible (**ej 11**).
- Describe el algoritmo más eficiente que se te ocurra para comprobar si una fórmula en CNF es tautología. ¿Qué coste tiene? (Sugerencia: ver **ej 12**)

Satisfactibilidad y Tautología: CNF vs DNF

- Comprobar si una fórmula en DNF es tautología es NP-completo.
- Comprobar si una fórmula en DNF es satisfactible es lineal (ver solución del ejercicio 12 del capítulo 3).
- Transformar una fórmula en DNF preservando la satisfactibilidad es NP-completo.

- Comprobar si una fórmula en CNF es tautología es lineal.
- El problema K-SAT consiste en encontrar una interpretación que satisfaga una fórmula expresada en K-CNF, i.e. cuyas cláusulas contengan exactamente K literales.
- Una fórmula en CNF puede convertirse en una fórmula equisatisfactible de 3-CNF en tiempo polinómico (ver solución del ejercicio 8 del capítulo 3).
- El problema K-SAT con $K \geq 3$ es NP-completo, mientras que 2-SAT es polinómico.