Tema 4:
Algoritmo DPLL y Teorema de Herbrand

Dpto. Ciencias de la Computación Inteligencia Artificial
Universidad de Sevilla

Lógicas Informática
(Tecnologías Informáticas)
Curso 2017–18

Contenido

Otro Algoritmo Proposicional

- Presentaremos un algoritmo más para estudiar la satisfactibilidad de un conjunto de fórmulas proposicionales:
  - El **algoritmo DPLL** (Davis–Putnam–Logemann-Loveland)
- Este algoritmo hace una búsqueda sistemática de modelos.
- Y al igual que los Tableros Semánticos, suele representarse gráficamente mediante un árbol.

El algoritmo DPLL

- Es un algoritmo para determinar la **satisfactibilidad** de un conjunto de cláusulas.
- El algoritmo DPLL es un refinamiento (presentado en 1962 por Davis, Logemann y Loveland) de un algoritmo propuesto por Davis y Putnam (en 1960).
- DPLL es la base de muchos "SAT solvers": programas para determinar la satisfactibilidad de un conjunto de fórmulas proposicionales (habitualmente, cláusulas).
- Puede utilizarse como algoritmo de decisión, pero si la respuesta es positiva también permite obtener una valoración que es modelo del conjunto de cláusulas de entrada.

Tableros vs DPLL

Tienen características propias que distinguen claramente ambos métodos:

- **Algoritmo DPLL**:
  - No trabaja con fórmulas arbitrarias sino sobre conjuntos de **cláusulas**. Es preciso "preprocesar" el conjunto de fórmulas, pasándolo a forma clausal.
  - Está entre los algoritmos más eficientes para la lógica proposicional, pero no se extiende fácilmente a otras lógicas.
- **Tableros semánticos**:
  - Trabaja directamente sobre el conjunto de fórmulas proposicionales.
  - No es tan eficiente como DPLL, pero es muy flexible y puede adaptarse a otras lógicas (como la lógica de primer orden, lógicas descriptivas, modales, etc.).
  - Resulta útil en el estudio teórico de diversas lógicas, para probar propiedades formales como la completitud.

Estructura del algoritmo

- Podemos distinguir dos partes en el algoritmo:
    1. **Propagación de unidades**. Esta parte está dedicada a la **simplificación** del conjunto sobre el que se trabaja.
    2. **División**. Esta parte organiza la búsqueda de una valoración que muestre que el conjunto es satisfactible.
- El algoritmo puede describirse de manera recursiva:
    - Dado un conjunto de clásulas S, en una primera fase simplificamos S mediante la propagación de unidades.
        - Si tras este proceso el conjunto S queda vacío, el conjunto es satisfactible.
        - Si durante el proceso de simplificación aparece la cláusula vacía, el conjunto es insatisfactible.
    - Una vez simplificado S, se obtienen mediante división dos conjuntos S' y S'' a los que volvemos a aplicar el procedimiento de simplificación y división.

Propagación de unidades

Esta fase utiliza dos reglas para simplificar el conjunto de cláusulas, S, sobre el que se trabaja.

- Se elige una cláusula unitaria $L \in S$ y se aplican consecutivamente las dos reglas siguientes:
  1. **Subsunción unitaria**. Se eliminan de S todas las cláusulas subsumidas por L, es decir, que contengan el literal L (inluida la propia cláusula L).
  2. **Resolución unidad**. Se elimina el literal complementario L^c de todas las cláusulas de S.
- El proceso vuelve a repetirse hasta que no queden cláusulas unitarias en S.

División

- Tras el proceso de simplificación si el algoritmo no ha parado, entonces S no contiene cláusulas unitarias.
- Se elige un literal L que aparezca en una cláusula de S y se construyen los conjuntos: $S \cup \{L\}$ y $S \cup \{L^c\}$. A continuación:
  1. Se aplica recursivamente el procedimiento a $S \cup \{L\}$; es decir, aplicamos de nuevo propagación de unidades (eligiendo necesariamente la cláusula unitaria L) y después división, etc.
  2. Si $S \cup \{L\}$ resulta ser insatisfactible, aplicamos el procedimiento a $S \cup \{L^c\}$.

Ejemplo

$$
\begin{aligned}
S \quad = \quad & \{\{a,b\},\{\neg a,b\},\{a,\neg b\},\{a,\neg d\},\\
& \{\neg a,\neg b,\neg c\},\{b,\neg c,\},\{c,\neg f\},\{f\}\}
\end{aligned}
$$

- Propagación de unidades:

  f:

  $$\{\{a,b\},\{\neg a,b\},\{a,\neg b\},\{a,\neg d\},\{\neg a,\neg b,\neg c\},\{b,\neg c\},\{c\}\}$$

  c:

  $$\{\{a,b\},\{\neg a,b\},\{a,\neg b\},\{a,\neg d\},\{\neg a,\neg b\},\{b\}\}$$

  b:

  $$\{\{a\},\{a,\neg d\},\{\neg a\}\}$$

  $\neg a$:

  $$\{\Box,\{d\}\}$$

- Insatisfactible.

Ejemplo (II)

$$\{\{p, q, r\}, \{\neg p, q, r\}, \{\neg q,\ r\}, \{\neg q,\ \neg r\}, \{q,\ \neg r\}\}$$

$$\frac{p}{\{\{q, r\}, \{\neg q, r\}, \{q, \neg r\}, \{\neg q, \neg r\}\}} \qquad \frac{\neg p}{\{\{q, r\}, \{\neg q, r\}, \{q, \neg r\}, \{\neg q, \neg r\}\}}$$

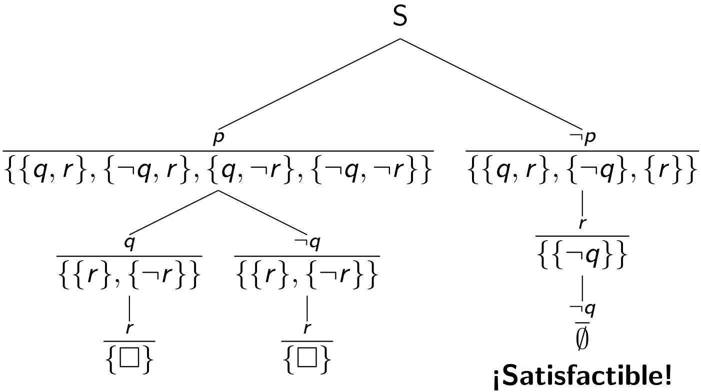
$$\frac{q}{\{\{r\}, \{\neg r\}\}} \quad \frac{\neg q}{\{\{r\}, \{\neg r\}\}} \qquad \frac{q}{\{\{r\}, \{\neg r\}\}} \quad \frac{\neg q}{\{\{r\}, \{\neg r\}\}}$$

$$\frac{r}{\{\square\}} \quad \frac{r}{\{\square\}} \qquad \frac{r}{\{\square\}} \quad \frac{r}{\{\square\}}$$

- Por tanto,
  $S = \{\{p, q, r\}, \{\neg p, q, r\}, \{\neg q,\ r\}, \{\neg q,\ \neg r\}, \{q,\ \neg r\}\}$
  es **insatisfactible**.

Ejemplo (III)

$$S = \{\{p, q, r\}, \{\neg p, q, r\}, \{p, \neg q\}, \{p, r\}, \{\neg p, \neg q, r\}, \{\neg p, q, \neg r\}, \{\neg p, \neg q, \neg r\}\}$$



- Un modelo de S está dado por $v(p) = 0$, $v(r) = 1$, $v(q) = 0$.

Bajando de LPO a Proposicional

A continuación vamos a ver cómo podemos reducir el caso de LPO (Primer Orden) al caso más sencillo y completo de LP (Proposicional):

- Comenzaremos manipulando las fórmulas de LPO para poder obtener fórmulas equivalentes (basándonos en la Forma Prenex de las mismas): la llamada Forma de Skolem.
- Posteriormente, daremos un resultado que permite reducir la consistencia de las fórmulas de Primer Orden a un conjunto de fórmulas proposicionales.

Formas de Skolem

- **Objetivo:** Restringir la complejidad sintáctica de las fórmulas de primer orden, sin perder expresividad.
- Trabajaremos con fórmulas **cerradas**.
- Primer paso: Trasladar los cuantificadores a la izquierda obteniendo un forma normal prenexa.
- El siguiente paso es eliminar los cuantificadores **existenciales** hasta obtener una **fórmula universal** (es decir, una fórmula en forma prenex que sólo contiene cuantificadores universales).
- La fórmula universal obtenida al final de estas transformaciones es una **forma de Skolem** de la fórmula inicial.
- El proceso de eliminación aumenta el lenguaje con nuevos símbolos de función y nuevas constantes.
- En general, NO existe equivalencia entre la fórmula obtenida y la fórmula inicial, aunque se conserva la consistencia de la fórmula original.

Funciones y constantes de Skolem

Para eliminar los cuantificadores existenciales de una fórmula en forma prenexa aplicamos las siguientes reglas:

- Por cada bloque

$$\forall x_1 \cdots \forall x_n \exists y F(x_1, \ldots, x_n, y)$$

  introducir un nuevo símbolo de función g (aridad n), y reescribir como

$$\forall x_1 \ldots \forall x_n F(x_1, \ldots, x_n, g(x_1, \ldots, x_n))$$

  (decimos que $\forall x_1 \ldots \forall x_n \, F(x_1, \ldots, x_n, g(x_1, \ldots, x_n))$ se obtiene a partir de $\forall x_1 \cdots \forall x_n \exists y F(x_1, \ldots, x_n, y)$ introduciendo una función de Skolem).
- Por cada bloque del tipo $\exists x F(x)$ añadir una nueva constante c y reescribir como $F(c)$.
  (Se dice que $F(c)$ se obtiene a partir de $\exists x \, F(x)$ introduciendo una constante de Skolem).

Funciones y constantes de Skolem: Propiedades

- g se denomina una *función de Skolem* y c una *constante de Skolem*.
- La introducción de funciones y constantes de Skolem conserva la consistencia. Es decir, dada una fórmula F, si F' se obtiene a partir de F introduciendo una función o una constante de Skolem, entonces,

$$F \text{ tiene un modelo} \quad \Leftrightarrow \quad F' \text{ tiene un modelo.}$$

- Además, la introducción de funciones y constantes de Skolem, no aumenta el conocimiento deducible en el lenguaje original. Es decir,
  - Si F es una fórmula cerrada de un lenguaje de primer orden L y F' se obtiene a partir de F introduciendo una función o una constante de Skolem, entonces para toda fórmula H del lenguaje L se tiene:

$$F' \models H \quad \Leftrightarrow \quad F \models H$$

Formas de Skolem

Sea L un LPO y F una fórmula cerrada de L.

- Una **forma de Skolem** de F es una fórmula universal G que se obtiene a partir de una forma prenexa de F mediante introducciones sucesivas de funciones o constantes de Skolem.
- Dado un conjunto $\Sigma = \{F_1, \ldots, F_n\}$ de fórmulas cerradas de L, sea Σ' el conjunto formado por las fórmulas de Skolem de lo elementos de Σ. Entonces,
  - Σ tiene un modelo si y sólo si Σ' tiene un modelo.
  - Para toda fórmula H del lenguaje L,

$$\Sigma \models H \quad \Leftrightarrow \quad \Sigma' \models H$$

Ejemplo

- $\forall x \exists y \exists z (P(y, x) \wedge P(z, y))$
- Dependencia de y con respecto a x: Elegimos mediante una función f_1:

$$\forall x \exists z (P(f_1(x), x) \wedge P(z, f_1(x)))$$

- Dependencia de z con respecto a x: Elegimos mediante una función f_2:

$$\forall x (P(f_1(x), x) \wedge P(f_2(x), f_1(x)))$$

- La fórmula universal

$$\forall x \, (P(f_1(x), x) \wedge P(f_2(x), f_1(x)))$$

es una **Forma de Skolem** de la fórmula inicial.

Otros ejemplos

$\exists x_1 \forall y_1 \exists z_2 \forall x_2 \exists y_2 ((x_1 + b = y_1) \rightarrow (x_2 \cdot y_2 = 0 + z_2))$

1. $\forall y_1 \exists z_2 \forall x_2 \exists y_2 ((c_1 + b = y_1) \rightarrow (x_2 \cdot y_2 = 0 + z_2))$
   [c nueva constante]
2. $\forall y_1 \forall x_2 \exists y_2 ((c_1 + b = y_1) \rightarrow (x_2 \cdot y_2 = 0 + f(y_1)))$
   [f nuevo símbolo de función de aridad 1]
3. $\forall y_1 \forall x_2 ((c_1 + b = y_1) \rightarrow (x_2 \cdot g(y_1, x_2) = 0 + f(y_1)))$
   [g nuevo símbolo de función de aridad 2]
4. **Forma de Skolem**:
   $\forall y_1 \forall x_2 ((c_1 + b = y_1) \rightarrow (x_2 \cdot g(y_1, x_2) = 0 + f(y_1))))$

En el lenguaje $LO = \{<, =\}$

- Sea $F \equiv \exists x \forall y (x < y \vee x = y)$
  Forma de Skolem: $\forall y (c < y \vee c = y)$
- $F \equiv \forall x \forall z (x < z \rightarrow \exists y (x < y \wedge y < z)$.
  Forma de Skolem:
  $\forall x \forall z (x < z \rightarrow x < f(x, z) \wedge f(x, z) < z)$.

Cláusulas (recordatorio)

Sea L un LPO.

- Una fórmula F es un literal si es una fórmula atómica o la negación de una fórmula atómica.
- Una cláusula es una disyunción de literales. Por tanto, es una fórmula abierta.
- Como en el caso de la lógica proposicional, identificaremos una cláusula con el conjunto de los literales que aparecen en ella.
- Denotaremos por $\square$ a la cláusula vacía.
- Dada una fórmula F de L una forma clausal de F es un conjunto de cláusulas S (no necesariamente del lenguaje L) tal que

$$F \text{ tiene un modelo} \iff S \text{ tiene un modelo}$$

Forma clausal

Para obtener una forma clausal de una fórmula $F(x_1, \ldots, x_n)$ seguimos los siguientes pasos:

1. Obtener el cierre universal de $F(x_1, \ldots, x_n)$. Es decir, la fórmula G dada por $\forall x_1 \ldots \forall x_n \, F(x_1, \ldots, x_n)$.
2. Obtener una forma de Skolem de G. Dicha forma de Skolem es una fórmula universal G_S.
3. Eliminar los cuantificadores universales de G_S. Así obtenemos una fórmula abierta H.
4. Obtener una forma normal conjuntiva de H (siguiendo el mismo procedimiento que en el caso proposicional). Dicha forma será
$$\bigwedge_{j=1}^{n} C_j$$
siendo cada C_j una cláusula.
5. La forma clausal de F es el conjunto de cláusulas $S = \{C_1, \ldots, C_n\}$.

Ejemplos

- $\forall x[\forall y\, H(x,y) \rightarrow \exists z \forall u(u \neq z \rightarrow P(z,u))]$
  - $\forall x \exists y\, [H(x,y) \rightarrow \exists z \forall u(u \neq z \rightarrow P(z,u))]$
  - $\forall x \exists y \exists z \forall u[H(x,y) \rightarrow (u \neq z \rightarrow P(z,u))]$
  - $\forall x \exists z \forall u\, [H(x, f_1(x)) \rightarrow (u \neq z \rightarrow P(z,u))]$
  - $\forall x \forall u[H(x, f_1(x)) \rightarrow (u \neq f_2(x) \rightarrow P(f_2(x), u))]$

  Forma clausal:

  $$\{\{\neg H(x, f_1(x)), u = f_2(x), P(f_2(x), u)\}\}$$

- $\forall x \forall z(x < z \rightarrow \exists y(x < y \wedge y < z))$
  - Forma clausal

    $$((\neg(x < z) \vee x < f(x,z)) \wedge (\neg(x < z) \vee f(x,z) < z))$$

  - O también

    $$\{\{\neg(x < z), x < f(x,z)\}, \{\neg(x < z), f(x,z) < z\}\}$$

Forma clausal de un conjunto de fórmulas

- Una forma clausal de un conjunto Γ es un conjunto de cláusulas S tal que

$$\Gamma \text{ tiene un modelo} \quad \Leftrightarrow \quad S \text{ tiene un modelo}$$

- Si $\Gamma = \{F_1, \ldots, F_n\}$ es un conjunto de fórmulas podemos obtener una forma clausal de Γ calculando, por el método anterior, una forma clausal S_j para cada elemento F_j de Γ. La forma clausal de Γ es entonces el conjunto de cláusulas:

$$S = S_1 \cup \cdots \cup S_n$$

Reducción a la lógica proposicional

Sea Γ un conjunto de fórmulas de un lenguaje L.

- El cálculo de formas de Skolem reduce la consistencia de Γ a la consistencia de un conjunto Σ de fórmulas abiertas (de hecho, cláusulas) en un nuevo lenguaje L' (que extiende a L con nuevos símbolos).
- Es posible dar un paso más y reducir la consistencia de Γ a la consistencia de un conjunto de fórmulas abiertas y cerradas.
  - Una fórmula abierta y cerrada puede identificarse con su esqueleto proposicional y, por tanto, considerarse una fórmula proposicional.

Definición. Sea Σ un conjunto de fórmulas abiertas. La **extensión de Herbrand** de Σ, es el conjunto, $EH(\Sigma)$, formado por todas las fórmulas cerradas que pueden obtenerse sustituyendo, de todas las formas posibles, las variables de las fórmulas de Σ por términos cerrados.

Teorema de Herbrand

Teorema. Sea Σ un conjunto de fórmulas abiertas.

Son equivalentes:

1. Σ tiene un modelo.
2. $EH(\Sigma)$ tiene un modelo.
3. $EH(\Sigma)$ es satisfactible <u>proposicionalmente</u>.
   - Es decir, si identificamos cada fórmula de $EH(\Sigma)$ con su esqueleto proposicional, entonces el conjunto de fórmulas proposicionales así obtenido es satisfactible.

Observaciones.

- Este resultado reduce la consistencia de Σ a la de un conjunto de fórmulas proposicionales: $EH(\Sigma)$.
- $EH(\Sigma)$ puede ser un conjunto **infinito**. Sin embargo,
- (Compacidad) Si $EH(\Sigma)$ es inconsistente, entonces existe un subconjunto finito Σ_0 de $EH(\Sigma)$ que es inconsistente.

Ejemplo

- El conjunto $\Sigma = \{p(x) \to p(f(f(x))), \neg p(f(f(f(x)))), p(f(a))\}$ es inconsistente.
- Su extensión de Herbrand $EH(\Sigma)$, contiene entre otras las fórmulas
  1. $p(f(a))$
  2. $p(a) \to p(f(f(a)))\}$ $[\{x/a\}]$
  3. $\neg p(f(f(f(a))))$ $[\{x/a\}]$
  4. $p(f(a)) \to p(f(f(f(a))))$ $[\{x/f(a)\}]$
  5. $\neg p(f(f(f(f(a)))))$ $[\{x/f(a)\}]$
  6. $p(f(f(a))) \to p(f(f(f(f(a)))))\}$ $[\{x/f(f(a))\}]$
  7. $\neg p(f(f(f(f(f(a))))))$ $[\{x/f(f(a))\}]$

     $\vdots$
- El subconjunto de $EH(\Sigma)$ formado por las fórmulas 1, 3 y 4 es inconsistente.