Satisfabilidad Lógica

Carlos Linares López

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

10 de enero de 2014

Definiciones

- Una fórmula proposicional está en *Forma Normal Conjuntiva* (CNF) si es de la forma:

$$\bigwedge_{i=1}^{N} \bigvee_{j=1}^{k_i} l_{ij}$$

- Un literal l consiste en una variable x afirmada (x) o negada ($\overline{x}$)
- Un literal l es puro en F si $\bar{l}$ no ocurre en ninguna cláusula de F
- Una *tautología* es una fórmula proposicional siempre cierta para cualquier asignación de sus expresiones atómicas.

Problema de satisfabilidad

Definición 1

El problema de satisfabilidad consiste en encontrar un modelo M que satisfaga F, $M \models F$

Método de resolución

- Si F es un conjunto de cláusulas y l es pura en F entonces $F - l$ es satisfacible si y sólo si F es satisfacible

Definición 2

$$Res(F, x) = \begin{cases} F - x & \textit{si } x \textit{ es pura en } F \\ C_1 \vee C_2 & (x \vee C_1) \in F, (\overline{x} \vee C_2) \in F \\ & \textit{y } (C_1 \vee C_2) \textit{ no es tautologia} \\ C \in F & x \notin F \end{cases}$$

Propiedades

- El modelo de resolución asegura que el número de variables decrementa:

$$\frac{(p \vee r), (\overline{p} \vee s)}{(r \vee s)} \qquad \frac{p, \overline{p}}{\{\varnothing\}}$$

  pero no necesariamente el número de cláusulas
- Si resulta la cláusula vacía, $\{\varnothing\}$, F no es satisfacible
- Si $F = \varnothing$, F es satisfacible
- La resolución no genera modelos, pero preserva la satisfabilidad del problema resultante
- La aplicación directa del método de resolución da lugar al algoritmo de Davis-Putnam

Algoritmo de Davis-Putnam

1. Elegir un literal $l \in F$
2. Aplicar $\text{Res}(F, l)$ y anotar la variable usada y las cláusulas involucradas
3. Si resulta la cláusula vacía, detenerse. PROBLEMA NO SATISFACIBLE
4. Si resulta $F = \varnothing$ ir a 5. En otro caso, volver a 1
5. PROBLEMA SATISFACIBLE. Considerar la lista de variables y cláusulas en orden inverso, calculando un valor $\top$ ó $\perp$ para todas las variables involucradas[1]

[1] $\top$ y $\perp$ significan cierto y falso respectivamente

Ejemplo I

Considerar el problema de satisfabilidad lógica con las siguientes cláusulas:

$$
\begin{aligned}
C_1 &: (p \vee q) \\
C_2 &: (\overline{p} \vee r) \\
C_3 &: (q \vee \overline{r}) \\
C_4 &: \quad \overline{q}
\end{aligned}
$$

y encontrar un modelo M que la satisfaga con el algoritmo de Davis-Putnam (DP)

Ejemplo II

- **Paso 0**

$$G_0 = \{C_i\}_{i=1}^4$$

Inicialmente se consideran todas las clausulas

$$\text{varSelect}[0] = p$$

varSelect almacena la selección de variables por paso, y layerSeq las clausulas

$$\text{layerSeq}[0] = G_0 \backslash \text{Res}(G_0, p) = G_0 \backslash \{C_3, C_4, C_5 : (q \vee r)\} = \{C_1, C_2\}$$

Ejemplo III

- **Paso 1**

$$G_1 = \{C_i\}_{i=3}^{5}$$
$$\text{varSelect}[1] = r$$
$$\text{layerSeq}[1] = G_1 \backslash \text{Res}(G_1, r) = G_1 \backslash \{C_4, C_6 : q\} = \{C_3, C_5\}$$

Cláusulas pendientes

se escoge r

nótese que la resolución de C_3 con C_5 genera la clausula C_6

Ejemplo IV

- **Paso 2**

  $G_2 = \{C_4, C_6\}$ *Cláusulas pendientes*
  $\text{varSelect}[2] = q$ *Se escoge la última variable q*
  $\text{layerSeq}[2] = G_2 \backslash \text{Res}(G_2, q)$ *Problema no Satisfacible*

- La no factibilidad se detecta al intentar resolver $C_4 : (\overline{q})$ con la variable q que da la cláusula vacía $\{\varnothing\}$

Ejemplo I

Considerar el problema de satisfabilidad lógica con las siguientes cláusulas:

$$\begin{array}{ll} C_1{:}(p \vee q) & C_5{:}\quad (\overline{r} \vee \overline{s}) \\ C_2{:}(\overline{p} \vee \overline{r}) & C_6{:}\quad (u \vee w) \\ C_3{:}(q \vee \overline{r}) & C_7{:}(s \vee \overline{u} \vee \overline{w} \vee x) \\ C_4{:}\quad \overline{q} & \end{array}$$

y encontrar un modelo M que la satisfaga con el algoritmo de Davis-Putnam (DP)

Ejemplo II

- **Paso 0**

$G_0 = \{C_i\}_{i=1}^{7}$ — *Inicialmente se consideran todas las clausulas*

$\text{varSelect}[0] = p$ — *almacena la selección de p*

$\text{layerSeq}[0] = G_0 \backslash \text{Res}(G_0, p) = G_0 \backslash \{C_3, C_4, C_5, C_6, C_7\} = \{C_1, C_2\}$ — *y las clausulas implicadas*

- La resolución de las cláusulas C_1 y C_2 genera exactamente la cláusula $C_3 : (q \vee \overline{r})$

Ejemplo III

- **Paso 1**

$G_1 = \{C_i\}_{i=3}^{7}$ *Cláusulas pendientes*

$\text{varSelect}[1] = u$ *se escoge u*

$\text{layerSeq}[1] = G_1 \backslash \text{Res}(G_1, u) = G_1 \backslash \{C_3, C_4, C_5\} = \{C_6, C_7\}$ *y se almacenan las clausulas usadas*

- Nótese que la resolución de las cláusulas C_6 y C_7 genera la cláusula $(s \vee w \vee \overline{w} \vee x)$ que es una tautología y, por lo tanto, no se añade

Ejemplo IV

- **Paso 2**

$$G_2 = \{C_3, C_4, C_5\}$$
$$\text{varSelect}[2] = q$$
$$\text{layerSeq}[2] = G_2 \backslash \text{Res}(G_2, q) = G_1 \backslash \{C_5, C_8 : (\overline{r})\} = \{C_3, C_4\}$$

Cláusulas pendientes

Se escoge q

nótese que la resolución de C_3 y C_4 genera C_8

y se almacenan las clausulas usadas

Ejemplo V

- **Paso 3**

$$
\begin{aligned}
&G_3 = \{C_5, C_8\} && \textit{Cláusulas pendientes}\\
&\text{varSelect}[3] = r && \textit{Se escoge } r\\
&\text{layerSeq}[3] = G_3 \backslash \text{Res}(G_3, r) = && \bar{r} \textit{ es pura en } G_3\\
&\qquad G_3 \backslash \varnothing = &&\\
&\qquad \{C_5, C_8\} && \textit{y se almacenan las clausulas usadas}
\end{aligned}
$$

- Puesto que r es pura, todas las clausulas que la contienen se eliminan.

Ejemplo VI

- **Paso 4**: Se consideran los vectores de variables y cláusulas en orden inverso:

$$\text{layerSeq}^{-1} = (\{C_9\}, \{C_5, C_8\}, \{C_3, C_4\}, \{C_6, C_7\}, \{C_1, C_2\})$$
$$\text{varSelect}^{-1} = \langle s, r, q, u, p \rangle$$

Ejemplo VII

y se calculan por pasos asignaciones (función h) de verdadero o falso a cada variable que satisfagan sólo las clausulas implicadas en cada paso como se muestra en la siguiente tabla:

Paso	Variable	Cláusulas	Asignación
4	$\langle s \rangle$	$\{C_9\}$	$h(s) = \bot$
3	$\langle r \rangle$	$\{C_5, C_8\}$	$h(r) = \bot$
2	$\langle q \rangle$	$\{C_3, C_4\}$	$h(q) = \bot$
1	$\langle u \rangle$	$\{C_6, C_7\}$	$h(u) = \top, h(w) = \bot, h(x) = \bot$
0	$\langle p \rangle$	$\{C_1, C_2\}$	$h(p) = \top$

Cuadro : Generación del modelo

Algoritmo de Davis-Putnam-Logemann-Loveland

Definición 3

Resolución Unitaria: *forma de resolución en la que uno de los padres es una cláusula unitaria*

- Sea F un conjunto de fórmulas y $p \in \mathrm{At}$ donde At es el conjunto de literales de F. F es satisfacible si y sólo si $F \cup \{p\}$ es satisfacible o $F \cup \{\overline{p}\}$ es satisfacible

Algoritmo de Davis-Putnam-Logemann-Loveland

Definición 4 *Se denomina* **reducción** *de una fórmula F por un modelo parcial v a la fórmula resultante $F_v = Red(F, v)$ en la que se han propagado las asignaciones de v*

- Si el modelo es completo y resulta el conjunto vacío, $\varnothing$, entonces la fórmula F es satisfacible y el modelo v lo valida

Pseudocódigo

DPLL (F, v)

Entrada: F es la expresión lógica en CNF a satisfacer
v es el modelo que se construye

Salida: Un modelo v tal que $v \models F$ o $\varnothing$ si no existe ninguno

```
if (∅ ∈ F)
    return ∅
G = Red(F, v)
if (G = ∅)
    return v
else
    l = SeleccionarLiteral(G)
    return DPLL (F, v ∪ {l}) ∨ (F, v ∪ {l̄})
```

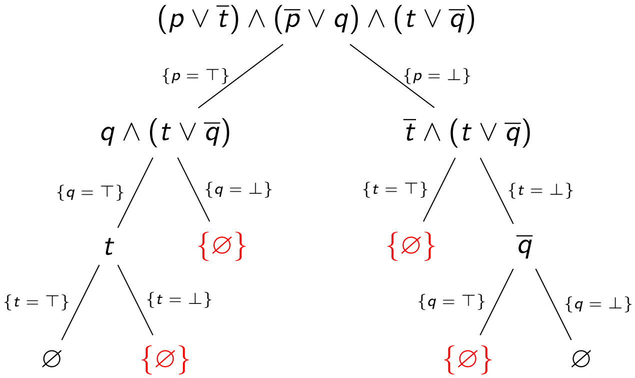
Ejemplo I

Considerar el problema de satisfabilidad lógica con las siguientes cláusulas:

$$C_1{:}(p \vee \overline{t})$$
$$C_2{:}(\overline{p} \vee q)$$
$$C_3{:}(t \vee \overline{q})$$

y encontrar un modelo M que la satisfaga con el algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

Ejemplo II


$(p \vee \overline{t}) \wedge (\overline{p} \vee q) \wedge (t \vee \overline{q})$
$\{p = \top\}$
$\{p = \bot\}$
$q \wedge (t \vee \overline{q})$
$\overline{t} \wedge (t \vee \overline{q})$
$\{q = \top\}$
$\{q = \bot\}$
$\{t = \top\}$
$\{t = \bot\}$
t
$\{\emptyset\}$
$\{\emptyset\}$
$\overline{q}$
$\{t = \top\}$
$\{t = \bot\}$
$\{q = \top\}$
$\{q = \bot\}$
$\emptyset$
$\{\emptyset\}$
$\{\emptyset\}$
$\emptyset$

Ejemplo III

- DPLL se podría haber detenido con el primer modelo:

$$v_1 = \{p = \top, q = \top, t = \top\}$$

pero si se le permite continuar encontraría todos los modelos adicionales:

$$v_2 = \{p = \bot, q = \bot, t = \bot\}$$

Propiedades

- DPLL es un algoritmo de el primero en profundidad por lo que tiene:
  - Consumo de memoria lineal
  - Consumo de tiempo exponencial

  en el número de variables
- Mejoras al algoritmo DPLL:
  - Algoritmo de selección de la próxima variable, SeleccionarLiteral(G)
  - Aprendizaje de nuevas cláusulas satisfacibles
  - *Backjumping* (o *backtracking* no cronológico)
  - Cálculo incremental de la función de reducción

Bibliografía

Marek, Victor W.
Introduction to Mathematics of Satisfiability
CRC Press, 2009