

DIM0436

23. DPLL e teorias

20141021

Sumário

1 Provedores SAT

2 Teorias da primeira ordem: aritmética

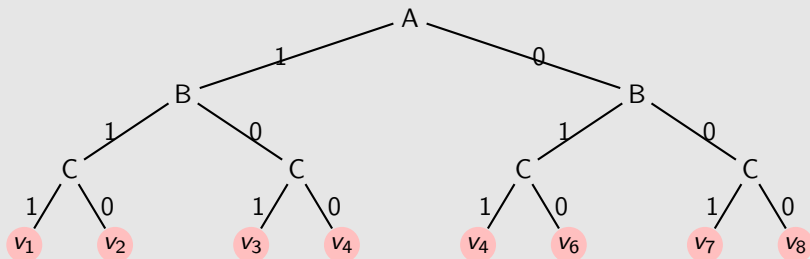
1 Provedores SAT

2 Teorias da primeira ordem: aritmética

Exemplo de árvore binária de decisão

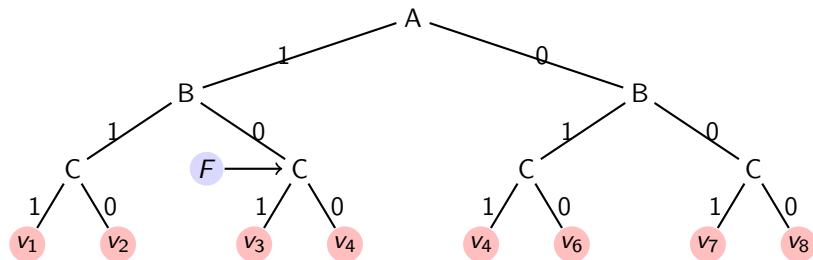
$$\Phi = (\neg A \vee B) \wedge (\neg B \vee C)$$

Árvore de decisão



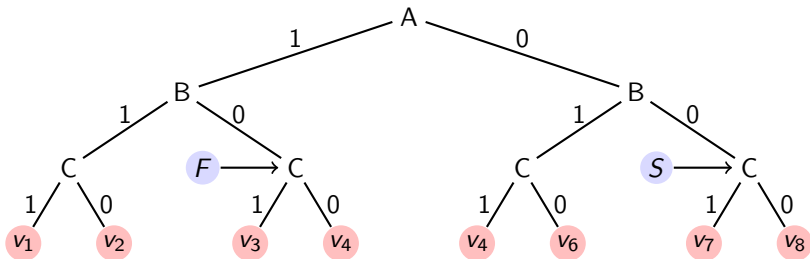
FNC

- $\Delta = \text{CNF}(\Phi) = \{\{\neg A, B\}, \{\neg B, C\}\}$
- $\Delta \mid A, \neg B = \{\{\perp, \perp\}, \{\top, C\}\} = \square$



Melhorada possível

- $\Delta = CNF(\Phi) = \{\{\neg A, B\}, \{\neg B, C\}\}$
- $\Delta \mid \neg A, \neg B = \{\{\top, \perp\}, \{\top, C\}\} = \square$
- Todas as cláusulas são subsumidas uma vez que $\{A \mapsto 0, B \mapsto 0\}$



DPLL(Δ , d)

```
if  $\Delta = \{\}$  then
  return  $\{\}$ 
else if  $\{\} \in \Delta$  then
  return UNSAT
else if  $L = \text{DPLL}(\Delta | P_{d+1}, d + 1) \neq \text{UNSAT}$  then
  return  $L \cup P_{d+1}$ 
else if  $L = \text{DPLL}(\Delta | \neg P_{d+1}, d + 1) \neq \text{UNSAT}$  then
  return  $L \cup \neg P_{d+1}$ 
else
  return UNSAT
end if
```

Resolução unitária

Problema

- Seja $\Delta = \{\{\neg A, B\}, \{\neg B, \neg C\}, \{C, \neg D\}\}$
- $\Delta|A = \{\{B\}, \{\neg B, \neg C\}, \{C, \neg D\}\}$
- Nesse ponto $\Delta \neq \{\} \wedge \{\} \notin \Delta$
- Mas poderíamos já declarar **sucesso**

Propagação unitária

- Antes do teste de sucesso/falha, propagar cláusulas unitárias
- A fase de resolução unitária produz dois resultados
 - ▶ I : literais presentes como cláusulas unitárias ou derivados por resolução unitária
 - ▶ Γ uma nova FNC.

Exemplos

$$\textcircled{1} \Delta = \{\{\neg A, \neg B\}, \{B, C\}, \{\neg C, D\}, \{A\}\}$$

$$\triangleright I = \{A, \neg B, C, D\}$$

$$\triangleright \Gamma = \{\}$$

$$\textcircled{2} \Delta = \{\{\neg A, \neg B\}, \{B, C\}, \{\neg C, D\}, \{C\}\}$$

$$\triangleright I = \{C, D\}$$

$$\triangleright \Gamma = \{\{\neg A, \neg B\}\}$$

Algoritmo $DPLL_{Unit}(\Delta)$

$(I, \Gamma) \leftarrow UNIT - RESOLUTION(\Delta)$

if $\Gamma = \{\}$ **then return** I

else

if $\{\} \in \Gamma$ **then return** UNSATISFIABLE

else

choose a literal L in Γ

if $L = DPLL(\Gamma|L) \neq UNSATISFIABLE$ **then return** $L \cup I \cup \{L\}$

else if $L = DPLL(\Gamma|\neg L) \neq UNSATISFIABLE$ **then return** $L \cup I \cup \{\neg L\}$

elsereturn UNSATISFIABLE

end if

end if

end if

Backtracking cronológico

- Se os dois valores duma variável ao nível n geram uma contradição, o algoritmo $DPLL_{Unit}$ volta ao nível $n - 1$
- Se o backtracking voltar até o nível 0 a FNC é incoerente.

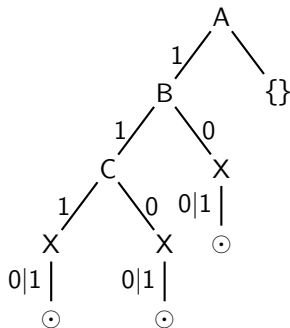
Diferenças

- Mudar do nível corrente a um nível inferior é **backtracking**
- Se o backtracking ao nível n for feito só após tentar os 2 valores ao nível $n + 1$, é **backtracking cronológico**

Exemplo

$\Delta =$

- ① $\{A, B\}$
- ② $\{B, C\}$
- ③ $\{\neg A, \neg X, Y\}$
- ④ $\{\neg A, X, Z\}$
- ⑤ $\{\neg A, \neg Y, Z\}$
- ⑥ $\{\neg A, X, \neg Z\}$
- ⑦ $\{\neg A, \neg Y, \neg Z\}$



Backtracking não cronológico

Conjunto de conflito

- Backtracking não cronológico pode ser usado após identificação de toda valoração que contribui à derivação da cláusula vazia.
- Esse conjunto é o **conjunto de conflito**
- Em vez de um backtracking até a última variável mudada, o algoritmo volta à variável de decisão **mais recente** do conjunto de conflito.

Exemplo

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

- 1 $\{A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1\}$
- 2 Por resolução unitária, $Y \mapsto 1, Z \mapsto 1\}$
- 3 Cláusula ?? vira vazia
- 4 O conjunto de conflito é $\{A \mapsto 1, X \mapsto 1, Y \mapsto 1, Z \mapsto 1\}$
- 5 Vamos dar um outro valor a X
- 6 Outro conflito : $\{A \mapsto 1, X \mapsto 1, Z \mapsto 1\}$
- 7 Como não tem mas valores para X , voltamos para $A \mapsto 0$

Pista de melhorada

$\Delta =$

① $\{A, B\}$

② $\{B, C\}$

③ $\{\neg A, \neg X, Y\}$

④ $\{\neg A, X, Z\}$

⑤ $\{\neg A, \neg Y, Z\}$

⑥ $\{\neg A, X, \neg Z\}$

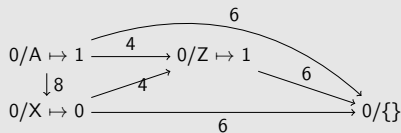
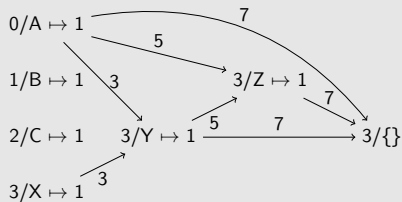
⑦ $\{\neg A, \neg Y, \neg Z\}$

- Cada vez que a resolução unitária descobre uma contradição, tem a possibilidade de identificar uma cláusula implicada pela FNC
- Essa cláusula permitiria a realização de novas implicações pela resolução unitária (assim $\{\neg A, \neg X\}$ é uma consequência da FNC acima)
- É a cláusula de conflito

Grafo de implicação

Como ler o grafo

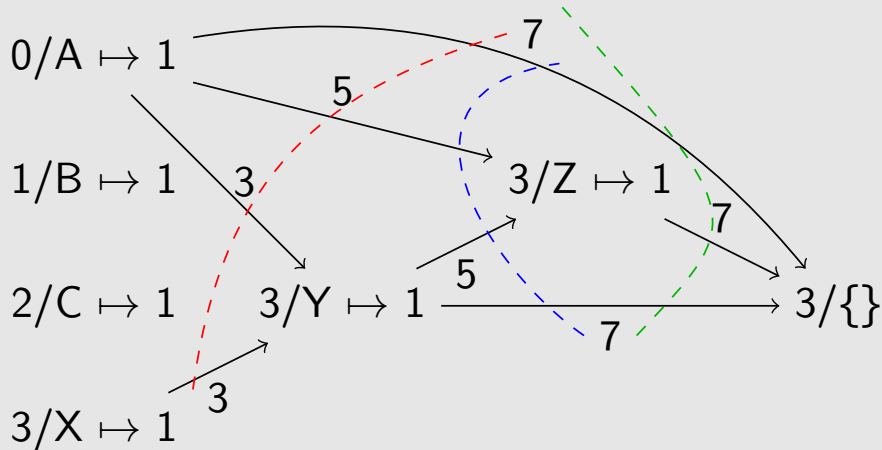
- $3/Y = 1$: a variável Y é valorada a verdadeira ao nível 3, usando cláusula 3, $A \mapsto 1, X \mapsto 1$



Calcular uma cláusula conflito

- Todo corte do grafo de implicação define um conjunto de conflito enquanto o corte separa as variáveis de decisão da contradição.
- Todo nó com uma aresta saindo cruzando o corte faz parte do conjunto de conflito.
- Após a identificação de um conjunto de conflito, a cláusula de conflito é calculada a partir da negação da valoração do conjunto de conflito
Assim se ele for $\{A \mapsto 1, B \mapsto 0, C \mapsto 0\}$, a cláusula de conflito é $\{\neg A, B, C\}$
- Uma cláusula de conflito gerada a partir de cortes que contem exatamente uma variável valorada nesse nível são *asserting*.
 - ▶ Cláusulas *asserting* são necessárias para a completude
 - ▶ O nível de asserção é o **segundo maior** nível da cláusula de conflito, -1 se não existir

Cortes



Algoritmo DPLL+

$D \leftarrow ()$

▷ sequência vazia de decisões

$\Gamma \leftarrow \{\}$

▷ nenhuma cláusula aprendida

while true do

if Unit-resolution acha uma contradição em Δ, Γ, D **then**

if $D = ()$ **then**

▷ contradição sem decisão

return UNSAT

else

▷ backtracking

$\alpha \leftarrow$ cláusula assertinda

$m \leftarrow$ nível de asserção de α

$D \leftarrow$ primeiras m decisões de D

▷ apagar decisões I_{m+1}, \dots

$\Gamma \leftarrow \{\alpha\} \cup \Gamma$

end if

else

▷ sem contradição por resolução unitária

if l é um literal \wedge nem l nem $\neg l$ são implicadas por resolução unitária

(Δ, Γ, D) **then**

$D \leftarrow D; l$

▷ nova decisão l em D

else

return SAT

end if

end if

Exemplo final

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

1 $D = (), \Gamma = \{\}$

Exemplo final

$\Delta =$

- ① $\{A, B\}$
- ② $\{B, C\}$
- ③ $\{\neg A, \neg X, Y\}$
- ④ $\{\neg A, X, Z\}$
- ⑤ $\{\neg A, \neg Y, Z\}$
- ⑥ $\{\neg A, X, \neg Z\}$
- ⑦ $\{\neg A, \neg Y, \neg Z\}$

① $D = (), \Gamma = \{\}$

② Suponha que

$D = (A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1)$

Exemplo final

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

1 $D = (), \Gamma = \{\}$

2 Suponha que

$$D = (A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1)$$

3 **Cláusula de conflito** $\{\neg A, \neg X\}$ de nível 0

Exemplo final

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

1 $D = (), \Gamma = \{\}$

2 Suponha que

$$D = (A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1)$$

3 **Cláusula de conflito** $\{\neg A, \neg X\}$ de nível 0

4 **Backtracking** ao nível de asserção :

$$D = (A \mapsto 1), \Gamma = \{\{\neg A, \neg X\}\}$$

Exemplo final

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

1 $D = (), \Gamma = \{\}$

2 Suponha que

$D = (A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1)$

3 **Cláusula de conflito** $\{\neg A, \neg X\}$ de nível 0

4 **Backtracking** ao nível de asserção :

$D = (A \mapsto 1), \Gamma = \{\{\neg A, \neg X\}\}$

5 **Resolução unitária** acha um conflito, gerando $\{\neg A\}$ com nível (-1).

Exemplo final

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

1 $D = (), \Gamma = \{\}$

2 Suponha que

$$D = (A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1)$$

3 **Cláusula de conflito** $\{\neg A, \neg X\}$ de nível 0

4 **Backtracking** ao nível de asserção :

$$D = (A \mapsto 1), \Gamma = \{\{\neg A, \neg X\}\}$$

5 **Resolução unitária** acha um conflito, gerando $\{\neg A\}$ com nível (-1).

6 **Backtracking:** $D = (), \Gamma = \{\{\neg A, \neg X\}, \{\neg A\}\}$

Exemplo final

$\Delta =$

- 1 $\{A, B\}$
- 2 $\{B, C\}$
- 3 $\{\neg A, \neg X, Y\}$
- 4 $\{\neg A, X, Z\}$
- 5 $\{\neg A, \neg Y, Z\}$
- 6 $\{\neg A, X, \neg Z\}$
- 7 $\{\neg A, \neg Y, \neg Z\}$

1 $D = (), \Gamma = \{\}$

2 Suponha que

$$D = (A \mapsto 1, B \mapsto 1, C \mapsto 1, X \mapsto 1)$$

3 **Cláusula de conflito** $\{\neg A, \neg X\}$ de nível 0

4 **Backtracking** ao nível de asserção :

$$D = (A \mapsto 1), \Gamma = \{\{\neg A, \neg X\}\}$$

5 **Resolução unitária** acha um conflito, gerando $\{\neg A\}$ com nível (-1).

6 **Backtracking:** $D = (), \Gamma = \{\{\neg A, \neg X\}, \{\neg A\}\}$

7 ...

1 Provedores SAT

2 Teorias da primeira ordem: aritmética

Teoria de primeira ordem

Definição

Uma teoria de primeira ordem T é definida por

- uma **assinatura** Σ : um conjunto de constantes, funções e símbolos de predicados
 - um conjunto de **axiomas** \mathcal{A} , um conjunto de fórmulas de primeira ordem nas quais só constantes, funções e predicados de Σ aparecem.
-
- Uma Σ -fórmula contém constantes, funções e predicados de Σ bem como conectivos lógicos e quantificadores.
 - Os axiomas \mathcal{A} dão o sentido dos símbolos de Σ

Igualdade

- 1 $\forall x \ x = x$
- 2 $\forall x, y \ x = y \Rightarrow y = x$
- 3 $\forall x, y, z \ x = y \wedge y = z \Rightarrow x = z$
- 4 para todo inteiro positivo n e símbolo de função f de aridade n
$$\forall \bar{x}, \bar{y}, \bigwedge_i x_i = y_i \Rightarrow f(\bar{x}) = f(\bar{y})$$
- 5 para todo inteiro positivo n e símbolo de predicado p de aridade n
$$\forall \bar{x}, \bar{y}, \bigwedge_i x_i = y_i \Rightarrow p(\bar{x}) \Leftrightarrow p(\bar{y})$$

Aritmética de Peano

Assinatura

$$\Sigma_{PA} = \{0, 1, +, *, =\}$$

Axiomas

- 1 $\forall x \neg(x + 1 = 0)$
- 2 $\forall x, y \ x + 1 = y + 1 \Rightarrow x = y$
- 3 $F(0) \wedge \forall x; (F(x) \Rightarrow F(x + 1)) \Rightarrow \forall x F(x)$
- 4 $\forall x \ x + 0 = x$
- 5 $\forall x, y \ (x + (y + 1) = (x + y) + 1)$
- 6 $\forall x \ x * 0 = 0$
- 7 $\forall x, y \ x * (y + 1) = x * (y + x)$

Observação sobre indução

- Indução é um esquema de axiomas: é o conjunto de axiomas obtido por a substituição de F por cada Σ_{PA} -fórmula com precisamente uma variável livre.

Interpretação e decidibilidade

Interpretação: α_I

- $\alpha_I[0] \doteq 0_{\mathbb{N}}$
- $\alpha_I[1] \doteq 1_{\mathbb{N}}$
- $\alpha_I[+] \doteq +_{\mathbb{N}}$
- $\alpha_I[*] \doteq *_{\mathbb{N}}$
- $\alpha_I[=] \doteq =_{\mathbb{N}}$

Decidibilidade

- Satisfazibilidade e validade em T_{PA} é indecidível.
- O primeiro teorema de incompletude de Gödel implica que a aritmética de Peano não captura a "aritmética verdadeira".

Aritmética de Presburger

Assinatura

$$\Sigma_{\mathbb{N}} = \{0, 1, +, =\}$$

Axiomas

- 1 $\forall x \neg(x + 1 = 0)$
- 2 $\forall x, y \ x + 1 = y + 1 \Rightarrow x = y$
- 3 $F(0) \wedge \forall x; (F(x) \Rightarrow F(x + 1]) \Rightarrow \forall x F(x)$
- 4 $\forall x \ x + 0 = x$
- 5 $\forall x, y \ (x + (y + 1) = (x + y) + 1)$

Observação sobre indução

- Indução é um esquema de axiomas: é o conjunto de axiomas obtido por a substituição de F por cada $\Sigma_{\mathbb{N}}$ -fórmula com precisamente uma variável livre.

Interpretação e decidibilidade

Interpretação : α_I

- $\alpha_I[0] \doteq 0_{\mathbb{N}}$
- $\alpha_I[1] \doteq 1_{\mathbb{N}}$
- $\alpha_I[+] \doteq +_{\mathbb{N}}$
- $\alpha_I[=] \doteq =_{\mathbb{N}}$

Decidibilidade (Presburger)

$T_{\mathbb{N}}$ é decidível.

Usar a aritmética de Presburger para \mathbb{Z}

Considere a fórmula $F_0 = \forall w, x \exists y, z (x + 2y - z - 13 > -3w - 5)$

Introdução de diferenças

$$F_1 = \forall w_n, w_p, x_n, x_p \exists y_n, y_p, z_n, z_p \\ (x_p - x_n + 2(y_p - y_n) - (z_p - z_n) - 13 > -3(w_p - w_n) - 5)$$

Eliminação de $-$

$$F_2 = \forall w_n, w_p, x_n, x_p \exists y_n, y_p, z_n, z_p \\ \$(x_p + 2y_p + z_n) + 3w_p + 5 > x_n + 2y_n + z_p + 3w_n + 13 \$$$

Eliminação de $*$

$$F_3 = \forall w_n, w_p, x_n, x_p \exists y_n, y_p, z_n, z_p \exists u \\ \neg(u = 0) \wedge (x_p + y_p + y_p + z_n + w_p + w_p + w_p + u = \\ x_n + y_n + y_n + z_p + w_n + w_n + w_n + \underbrace{1 + \dots + 1}_8)$$

Exemplos

Eliminação dos quantificadores

Admissibilidade

Uma teoria T admite eliminação dos quantificadores se existir um algoritmo que, dado uma Σ -fórmula F , calcula uma fórmula G **sem quantificadores** T -equivalente a F .

Exemplo

Seja $F = \exists x \cdot 2x = y$.

- Se F for uma $\Sigma_{\mathbb{Q}}$ -fórmula, $G = \top$
- Se F for uma $\Sigma_{\mathbb{Z}}$ -fórmula, $G = ??$

Predicado de divisibilidade

- $k|\cdot$ para $k \in \mathbb{Z}^+$ tal que $k|x$ seja verdadeira sse $x \bmod k = 0$
- $\widehat{T}_{\mathbb{Z}} = T_{\mathbb{Z}} \cup \{\}$

Método de Cooper

Objetivo

- A entrada uma $\widehat{\Sigma}_{\mathbb{Z}}$ -fórmula $\exists x F(x)$, com F uma fórmula sem quantificador, mas com outras variáveis livres que x .
- O algoritmo vai construir uma $\widehat{\Sigma}_{\mathbb{Z}}$ -fórmula **sem quantificador** $\widehat{T}_{\mathbb{Z}}$ -equivalente a $\exists x F(x)$.

Etapa 1

- Calcular a NNF de $F(x)$, chamada de $F_1(x)$.
- $\exists x F(x) \equiv_{\widehat{T}_{\mathbb{Z}}} \exists x F_1(x)$

Método de Cooper (etapa 2)

Reescrita de literais

- 1 $s = t \rightarrow s < t + 1 \wedge t < s + 1$
- 2 $\neg(s = t) \rightarrow s < t \vee t < s$
- 3 $\neg(s < t) \rightarrow t < s + 1$

Análise da saída

A saída $\exists x F_2(x) \equiv_{\widehat{T}_{\mathbb{Z}}} \exists x F_1(x)$ e contem só literais da forma

$$s < t, k|t \text{ ou } \neg(k|t)$$

Exemplo

Reescrever $\neg(x < y) \wedge \neg(x = y + 3)$.

Método de Cooper (etapa 3)

Descrição

- Reunir os termos contendo x , para que os literais tiverem a forma

$$hx < t, t < hx, k|hx + t \text{ ou } \neg(k|hx + t) \text{ com } x \neg \in t$$

Saída

$$\exists x F_3(x) \equiv \widehat{\tau_z} \exists x F_2(x)$$

Exemplo

Reunir os termos da fórmula $x + x + y < z + 3z + 2y - 4x$

Método de Cooper (etapa 4)

Descrição

Seja $\delta = \text{mmc}\{h \mid h \text{ coeficiente de } x \text{ em } F_3(x)\}$.

- 1 $hx < t \rightarrow \delta x < h't \quad h'h = \delta$
- 2 $t < hx \rightarrow h't < \delta x \quad h'h = \delta$
- 3 $k|hx + t \rightarrow h'k|\delta x + h't \quad h'h = \delta$
- 4 $\neg(k|hx + t) \rightarrow \neg(h'k|\delta x + h't) \quad h'h = \delta$

Análise da saída

- $h'k| \cdot \neg k| \cdot$
- A fórmula obtida é F'_3
 - 1 Seja $F''_3 = F'_3\{\delta x \mapsto x'\}$
 - 2 $\exists x' F_4(x') = \exists x' F''_3(x') \wedge \delta|x'$
- $\exists x' F_4(x') \equiv_{\widehat{T}_Z} \exists x F_3(x)$, com literais de F_4 da forma
(A) $x' < a$, (B) $b < x'$, (C) $h|x' + c$ ou (D) $k|x' + d$, $x \neg \in a, b, c$, ou d

Método de Cooper (etapa 5)

Descrição

- Construir a projeção esquerda infinita $F_{-\infty}(x')$ de $F_4(x')$
- Reescrita de
 - ① $x' < a \rightarrow \top$
 - ② $b < x' \rightarrow \perp$
- Seja
 - ▶ $\gamma = \text{mmc}\{h \text{ dos literais } h|x' + c, k \text{ dos literais } k|x' + d\}$
 - ▶ $B = \{b \text{ dos literais } (\mathbf{B})\}$
- Construir $F_5 = \bigvee_{j=1}^{\gamma} F_{-\infty}(j) \vee \bigvee_{j=1}^{\gamma} \bigvee_{b \in B} F_4(b + j)$

Exercícios

Assunto

Transforme com o método de Cooper as fórmulas abaixo:




- 1 $\exists x 3x - 2y + 1 > -y \wedge 2x - 6 < z \wedge 4|5x + 1$
- 2 $\exists x 2x = y$
- 3 $\exists x (3x + 1 < 10 \vee 7x - 6 > 7) \wedge 2|x$

Resumo

1 Provedores SAT

2 Teorias da primeira ordem: aritmética

Referências

-  Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh (eds.), *Handbook of Satisfiability*, IOS Press, 2009.
-  Aaron R. Bradley and Zohar Manna, *The Calculus of Computation: Decision Procedures with Applications to Verification*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
-  Daniel Kroening and Ofer Strichman, *Decision Procedures: An Algorithmic Point of View*, 1 ed., Springer Publishing Company, Incorporated, 2008.

Perguntas ?



<http://dimap.ufrn.br/~richard/dim0436>