Korektné tablové pravidlá. DPLL

7. prednáška · Matematika (4): Logika pre informatikov

Ján Kľuka, Jozef Šiška

Letný semester 2019/2020

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky
Katedra aplikovanej informatiky

Obsah 7. prednášky

Rekapitulácia

Minulý týždeň:

- Dokázali sme korektnosť tabiel.
- Preskúmame, čo vedia tablá povedať o splniteľnosti.
- Dokázali sme úplnosť tabiel.

Tento týždeň:

- Pohodlnejšie tablá pomocou ďalších korektných pravidiel.
- SAT solver a algoritmus DPLL.

Dôkazy a výrokovologické tablá

Dôkazy a výrokovologické tablá

Nové korektné pravidlá

Problémy so základnými pravidlami

Základné tablové pravidlá sú jednoduché, ľahko overiteľné a analytické — z (ne)pravdivosti zloženej formuly odvodzujú (ne)pravdivosť jej priamych podformúl.

Nie sú ale úplne pohodlné ani prirodzené, hlavne β.

Príklad 5.1

Dokážme, že pre všetky formuly A, B, C, X, Y, Z:

$$\{(A \to C), (B \to C), (C \to X), (C \to Y), ((X \land Y) \to Z)\} \\ \vdash_{\mathrm{p}} ((A \lor B) \to Z)$$

Všimnime si:

- časté použitia pravidla β na implikáciu, kde sa jedna vetva ihneď uzavrie;
- opakovanie jedného podstromu dôkazu.

Riešenie príkladu 5.1

Tablo pre

$$S^+ = \{\,\mathbf{T}(A \to C), \mathbf{T}(B \to C), \mathbf{T}(C \to X), \mathbf{T}(C \to Y), \mathbf{T}((X \land Y) \to Z),$$
$$\mathbf{F}((A \lor B) \to Z)\}$$

1. **T** $(A \to C)$	S^+
2. **T** $(B \to C)$	S^+
3. **T** $(C \to X)$	S^+
4. **T** $(C \to Y)$	S^+
5. **T** $((X \land Y) \to Z)$	S^+
6. **F** $((A \lor B) \to Z)$	S^+
7. **T** $(A \lor B)$	$\alpha 6$
8. **F** Z	$\alpha 6$

- 9. **F** $(X \land Y)$ $\beta 5$
  - 10. **T** A $\beta 7$
    - 11. **F** A $\beta 1$ — $*$ 10, 11
    - 12. **T** C $\beta 1$
      - 13. **F** C $\beta 3$ — $*$ 12, 13
      - 14. **T** X $\beta 3$
        - 15. **F** C $\beta 4$ — $*$ 12, 15
        - 16. **T** Y $\beta 4$
          - 17. **F** X $\beta 9$ — $*$ 14, 17
          - 18. **F** Y $\beta 9$ — $*$ 16, 18
  - 19. **T** B $\beta 7$
    - 20. **F** B $\beta 2$ — $*$ 19, 20
    - 21. **T** C $\beta 2$
      - 22. **F** C $\beta 3$ — $*$ 21, 22
      - 23. **T** X $\beta 3$
        - 24. **F** C $\beta 4$ — $*$ 21, 24
        - 25. **T** Y $\beta 4$
          - 26. **F** X $\beta 9$ — $*$ 23, 26
          - 27. **F** Y $\beta 9$ — $*$ 25, 27
- 28. **T** Z $\beta 5$ — $*$ 8, 28

Odstránenie problémov — nové pravidlá

Keby tablový kalkul obsahoval napríklad veľmi prirodzené pravidlá *modus ponens*, *modus tolens* a *rez*:

$$\frac{\mathbf{T}(X \to Y) \quad \mathbf{T}\,X}{\mathbf{T}\,Y} \tag{MP}$$

$$\frac{\mathbf{T}(X \to Y) \quad \mathbf{F}\,Y}{\mathbf{F}\,X} \tag{MT}$$

$$\frac{}{\mathbf{T}\,X \mid \mathbf{F}\,X} \tag{cut}$$

dôkaz v príklade by sa dal sprehľadniť a odstránila by sa duplicita.

Riešenie príkladu 5.1 s modus ponens a modus tolens

1.	$\mathbf{T}(A \to C)$	S^+
2.	$\mathbf{T}(B \to C)$	S^+
3.	$\mathbf{T}(C \to X)$	S^+
4.	$\mathbf{T}(C \to Y)$	S^+
5.	$\mathbf{T}((X \wedge Y) \to Z)$	S^+
6.	$\mathbf{F}((A \vee B) \to Z)$	S^+
7.	$\mathbf{T}(A \vee B)$	$\alpha 6$
8.	$\mathbf{F}Z$	$\alpha 6$
9.	$\mathbf{F}(X \wedge Y)$	MT $5, 8$

<table>
<tr><td colspan="2">10. $\mathbf{T}A$ $\beta 7$
11. $\mathbf{T}C$ MP $1, 10$
12. $\mathbf{T}X$ MP $3, 11$
13. $\mathbf{T}Y$ MP $4, 11$</td><td colspan="2">16. $\mathbf{T}B$ $\beta 7$
17. $\mathbf{T}C$ MP $2, 16$
18. $\mathbf{T}X$ MP $3, 17$
19. $\mathbf{T}Y$ MP $4, 17$</td></tr>
<tr><td>14. $\mathbf{F}X$ $\beta 9$
$* \; 12, 14$</td><td>15. $\mathbf{F}Y$ $\beta 9$
$* \; 13, 15$</td><td>20. $\mathbf{F}X$ $\beta 9$
$* \; 18, 20$</td><td>21. $\mathbf{F}Y$ $\beta 9$
$* \; 19, 21$</td></tr>
</table>

Riešenie príkladu 5.1 s rezom, modus ponens a modus tolens

1.	$\mathbf{T}(A \to C)$	S^+
2.	$\mathbf{T}(B \to C)$	S^+
3.	$\mathbf{T}(C \to X)$	S^+
4.	$\mathbf{T}(C \to Y)$	S^+
5.	$\mathbf{T}((X \land Y) \to Z)$	S^+
6.	$\mathbf{F}((A \lor B) \to Z)$	S^+
7.	$\mathbf{T}(A \lor B)$	$\alpha 6$
8.	$\mathbf{F}Z$	$\alpha 6$
9.	$\mathbf{F}(X \land Y)$	MT $5, 8$

- 10. $\mathbf{T}C$ cut
  11. $\mathbf{T}X$ MP $3, 10$
  12. $\mathbf{T}Y$ MP $4, 10$
  - 13. $\mathbf{F}X$ $\beta 9$
    $* \; 11, 13$
  - 14. $\mathbf{F}Y$ $\beta 9$
    $* \; 12, 14$
- 15. $\mathbf{F}C$ cut
  - 16. $\mathbf{T}A$ $\beta 7$
    17. $\mathbf{T}C$ MP $1, 16$
    $* \; 15, 17$
  - 18. $\mathbf{T}B$ $\beta 7$
    19. $\mathbf{T}C$ MP $2, 18$
    $* \; 15, 19$

Ingrediencie korektnosti a úplnosti tabiel

Všimnite si:

Na dokázanie **korektnosti** tablového kalkulu stačilo, aby mali pravidlá vlastnosť:

$$\frac{\alpha}{\alpha_1} \qquad \frac{\alpha}{\alpha_2} \qquad \frac{\beta}{\beta_1 \mid \beta_2} \qquad \frac{}{A^+}\ A^+ \in S^+$$

Nech v je ľubovoľné ohodnotenie, v ktorom je pravdivá S^+.
Ak je vo v pravdivá premisa, tak je vo v pravdivý aspoň jeden záver.

- Vďaka tejto vlastnosti zo splniteľnej množiny S^+ skonštruujeme iba splniteľné tablá.
- Netreba opačnú implikáciu (ak je vo v pravdivý aspoň jeden záver, tak je vo v pravdivá premisa).

Na dôkaz **úplnosti** stačili pravidlá (S^+), α, β, pretože stačia na vybudovanie úplného tabla.

Nové pravidlo

Čo sa stane, ak pridáme nové pravidlo, napríklad modus ponens:

$$\frac{\mathbf{T}(X \to Y) \quad \mathbf{T}\,X}{\mathbf{T}\,Y} \quad ? \qquad \text{(MP)}$$

Upravíme definíciu priameho rozšírenia:

Úprava definície tabla

… Nech $\mathcal{T}$ je tablo pre S^+ a y je nejaký jeho list. Potom tablom pre S^+ je aj každé *priame rozšírenie* $\mathcal{T}$ ktorýmkoľvek z pravidiel:

α: ⋯

⋮

MP: Ak sa na vetve π_y nachádzajú *obe* formuly $\mathbf{T}(X \to Y)$ a $\mathbf{T}\,X$, tak ako jediné dieťa y pripojíme nový vrchol obsahujúci $\mathbf{T}\,Y$.

Nové pravidlo vs. korektnosť a úplnosť

Korektnosť tabiel s MP:

Pri dôkaze lemy K1

> *Nech S^+ je množina označených formúl v jazyku $\mathcal{L}$, nech $\mathcal{T}$ je tablo pre S^+ a v je ohodnotenie pre $\mathcal{L}$. Ak sú S^+ a $\mathcal{T}$ pravdivé vo v, tak je vo v pravdivé aj každé priame rozšírenie tabla $\mathcal{T}$.*

využijeme

Tvrdenie 5.2 (Korektnosť pravidla MP)

Nech X a Y sú ľubovoľné formuly a v je ľubovoľné ohodnotenie. Ak sú vo v pravdivé $\mathbf{T}(X \to Y)$ a $\mathbf{T}\,X$, tak je vo v pravdivá $\mathbf{T}\,Y$.

Dôkaz.

Keďže $v \models_p \mathbf{T}(X \to Y)$, tak $v \models_p (X \to Y)$, teda $v \not\models_p X$ alebo $v \models_p Y$. Pretože ale $v \models_p \mathbf{T}\,X$, tak $v \models_p X$. Takže $v \models_p Y$, a teda $v \models_p \mathbf{T}\,Y$. $\square$

Dôkaz lemy K2 a samotnej vety o korektnosti — bez zmeny.

Úplnosť — bez zmeny, úplné tablo vybudujú základné pravidlá.

Tablové pravidlá vo všeobecnosti — problém

Zadefinovať vo všeobecnosti, čo je pravidlo a kedy je korektné, nie je také jednoduché.

Potrebujeme zachytiť, že pravidlo:

- má premisy, ktoré nejaký tvar a zdieľajú nejaké podformuly, napr. moduls tolens (MT) má premisy $\mathbf{T}(X \to Y)$ a $\mathbf{F}\,Y$;
- odvodzuje z nich závery, ktoré tiež zdieľajú podformuly s premisami, napr. $\mathbf{F}\,X$ (alebo medzi sebou v prípade rezu).

pre všetky možné zdieľané podformuly, v našom príklade X a Y.

Tablové pravidlá vo všeobecnosti — vzor

Pravidlo sa dá predstaviť nasledovne:

Pravidlo má vzor — dvojicu tvorenú vzormi premís a záverov, kde spoločné podformuly predstavujú konkrétne atómy, napr. vzor pravidla MT:

$$\frac{\mathbf{T}(\mathrm{p}(\mathrm{c}) \rightarrow \mathrm{q}(\mathrm{c})) \quad \mathbf{F}\,\mathrm{q}(\mathrm{c})}{\mathbf{F}\,\mathrm{p}(\mathrm{c})}$$

Tablové pravidlá vo všeobecnosti — inštancia

Každý konkrétny prípad — inštancia pravidla vznikne substitúciou ľubovoľných formúl za atómy vo vzore:

$$\frac{\begin{array}{c}\mathbf{T}(\mathtt{p}(\mathtt{c}) \to \mathtt{q}(\mathtt{c}))_{[\mathtt{p}(\mathtt{c})|(\mathtt{sedan}(\mathtt{a}) \land \mathtt{biely}(\mathtt{a})), \mathtt{q}(\mathtt{c})|\mathtt{kupi}(\mathtt{B}, \mathtt{a})]} \\ \mathbf{F}\,\mathtt{q}(\mathtt{c})_{[\mathtt{p}(\mathtt{c})|(\mathtt{sedan}(\mathtt{a}) \land \mathtt{biely}(\mathtt{a})), \mathtt{q}(\mathtt{c})|\mathtt{kupi}(\mathtt{B}, \mathtt{a})]}\end{array}}{\mathbf{F}\,\mathtt{p}(\mathtt{c})_{[\mathtt{p}(\mathtt{c})|(\mathtt{sedan}(\mathtt{a}) \land \mathtt{biely}(\mathtt{a})), \mathtt{q}(\mathtt{c})|\mathtt{kupi}(\mathtt{B}, \mathtt{a})]}}$$

$$= \frac{\begin{array}{c}\mathbf{T}((\mathtt{sedan}(\mathtt{a}) \land \mathtt{biely}(\mathtt{a})) \to \mathtt{kupi}(\mathtt{B}, \mathtt{a})) \\ \mathbf{F}\,\mathtt{kupi}(\mathtt{B}, \mathtt{a})\end{array}}{\mathbf{F}(\mathtt{sedan}(\mathtt{a}) \land \mathtt{biely}(\mathtt{a}))}$$

Tablové pravidlá vo všeobecnosti — pravidlo

Samotné pravidlo je množina všetkých inštancií vzoru:

$$\mathsf{MT} = \left\{ \left. \frac{\begin{array}{c}\mathbf{T}(\mathrm{p}(\mathrm{c}) \to \mathrm{q}(\mathrm{c}))_{[\mathrm{p}(\mathrm{c})|X,\,\mathrm{q}(\mathrm{c})|Y]} \\ \mathbf{F}\,\mathrm{q}(\mathrm{c})_{[\mathrm{p}(\mathrm{c})|X,\,\mathrm{q}(\mathrm{c})|Y]}\end{array}}{\mathbf{F}\,\mathrm{p}(\mathrm{c})_{[\mathrm{p}(\mathrm{c})|X,\,\mathrm{q}(\mathrm{c})|Y]}} \,\right|\, X, Y \in \mathcal{E}_{\mathcal{L}} \right\}$$

Samozrejme, *konkrétne* pravidlo vieme zapísať aj bez substitúcie:

$$\mathsf{MT} = \left\{ \left. \frac{\mathbf{T}(X \to Y) \quad \mathbf{F}\,Y}{\mathbf{F}\,X} \,\right|\, X, Y \in \mathcal{E}_{\mathcal{L}} \right\}$$

Definícia 5.3 (Vzor tablového pravidla)

Nech $n \geq 0$ a $k > 0$ sú prirodzené čísla, nech $P_1^+, \dots, P_n^+$, $C_1^+, \dots, C_k^+$ sú označené formuly.

Dvojicu tvorenú n-ticou $(P_1^+, \dots, P_n^+)$ a k-ticou $(C_1^+, \dots, C_k^+)$ a zapisovanú

$$\begin{array}{c|c|c} P_1^+ & \cdots & P_n^+ \\ \hline C_1^+ & \cdots & C_k^+ \end{array}$$

nazývame *vzorom tablového pravidla*.

Označené formuly $P_1^+, \dots, P_n^+$ nazývame *vzory premís*, označené formuly $C_1^+, \dots, C_k^+$ nazývame *vzory záverov*.

Definícia 5.4 (Tablové pravidlo a jeho inštancia)

Nech

$$\begin{array}{ccc} P_1^+ & \cdots & P_n^+ \\ \hline C_1^+ & \multicolumn{1}{|c|}{\cdots} & C_k^+ \end{array}$$

je vzor tablového pravidla a $a_1, \ldots, a_m$ sú všetky atómy, ktoré sa vyskytujú v označených formulách $P_1^+, \ldots, P_n^+, C_1^+, \ldots, C_k^+$.

Tablové pravidlo R je množina

$$R = \left\{ \begin{array}{ccc} P_1^+{}_{[a_1|X_1,\ldots,a_m|X_m]} & \cdots & P_n^+{}_{[a_1|X_1,\ldots,a_m|X_m]} \\ \hline C_1^+{}_{[a_1|X_1,\ldots,a_m|X_m]} & \multicolumn{1}{|c|}{\cdots} & C_k^+{}_{[a_1|X_1,\ldots,a_m|X_m]} \end{array} \;\middle|\; X_1,\ldots,X_m \in \mathcal{E}_{\mathcal{L}} \right\},$$

Každý prvok množiny R nazývame *inštanciou* pravidla R.

Keď už vieme, čo je pravidlo, môžeme povedať, kedy je korektné:

Definícia 5.5 (Tablové pravidlo a jeho korektnosť)

Tablové pravidlo R je *korektné* vtt
pre každú inštanciu pravidla R

$$\frac{P_1^+ \quad \cdots \quad P_n^+}{C_1^+ \;\big|\; \cdots \;\big|\; C_k^+}$$

a pre každé ohodnotenie v platí, že
ak sú vo v pravdivé všetky premisy $P_1^+, \ldots, P_n^+$,
tak je vo v pravdivý niektorý záver $C_1^+, \ldots, C_k^+$.

Úprava definície tabla

$\cdots$

- $\cdots$
- Nech $\mathcal{T}$ je tablo pre S^+ a y je nejaký jeho list. Potom tablom pre S^+ je aj každé ***priame rozšírenie*** $\mathcal{T}$ ktorýmkoľvek z pravidiel:

  $\vdots$

  ***R*:** Ak sa pre nejakú inštanciu pravidla R

$$\begin{array}{c}P_1^+ \quad \cdots \quad P_n^+ \\ \hline C_1^+ \mid \cdots \mid C_k^+\end{array}$$

  na vetve π_y nachádzajú *všetky* premisy $P_1^+, \ldots, P_n^+$, tak k uzlu y pripojíme k nových vrcholov obsahujúcich postupne závery $C_1^+, \ldots, C_k^+$.

Príklad: Korektnosť rezu

To, že rez

$$\frac{}{\mathbf{T}\,X \mid \mathbf{F}\,X}$$

je korektné pravidlo, dokážeme veľmi ľahko:

Tvrdenie 5.6 (Korektnosť pravidla rezu)

Nech X je ľubovoľná formula a v je ľubovoľné ohodnotenie.
Potom je vo v pravdivý niektorý zo záverov pravidla rezu
$\mathbf{T}\,X$ *alebo* $\mathbf{F}\,X$.

Dôkaz.

Formula X je vo v buď pravdivá alebo nepravdivá.
V prvom prípade $v \models_{\mathrm{p}} \mathbf{T}\,X$. V druhom prípade $v \models_{\mathrm{p}} \mathbf{F}\,X$.
Teda v oboch prípadoch platí, že vo v je pravdivý niektorý zo záverov $\mathbf{T}\,X$ alebo $\mathbf{F}\,X$ pravidla rezu. $\square$

Teoretické cvičenia a „midterm“

Teoretické cvičenia:

- Korektné pravidlá
- Tvrdenia o tablách

Domáce zadanie — tu07 + midterm:

tu07	1 úloha, 2 body (povinne opraviteľná)
midterm	2 úlohy po 5 bodov (neopraviteľný)

SAT a DPLL

SAT a DPLL

Problém výrokovologickej splniteľnosti (SAT)

Problém SAT

Definícia 6.1 (Problém SAT)

Problémom výrokovologickej splniteľnosti (*SAT*) je problém určenia toho, či je daná množina výrokovologických formúl splniteľná.

- Zvyčajne sa redukuje na problém splniteľnosti **klauzálnej** teórie (teda formuly v CNF).
- ***SAT solver*** je program, ktorý rieši problém SAT.

Príklad 6.2

Nech a, b, c sú predikátové atómy.

Nech $S = \{(a \lor b), (a \lor \neg b), (\neg a \lor b), (\neg a \lor \neg b \lor \neg c), (\neg a \lor c)\}$. Je množina klauzúl S splniteľná?

Tabuľková metóda

Tabuľková metóda:

- Skúma *všetky* ohodnotenia predikátových atómov
- Trvá $\mathrm{O}(s \cdot 2^N)$ krokov,
  - N je počet atómov a s je súčet veľkostí klauzúl
  - 2^N ohodnotení, pre každé treba zistiť, či sú všetky klauzuly pravdivé
- Zaberá priestor $\mathrm{O}(k \cdot 2^N)$
  - k je počet klauzúl
  - Pamätáme si (píšeme na papier) celú tabuľku
- Tabuľka slúži aj ako dôkaz prípadnej nesplniteľnosti

SAT a DPLL

Naivný backtracking

Naivný backtracking v Pythone

```
#!/usr/bin/env python3
class Sat(object):
    def __init__(self, n, clauses):
        self.n, self.clauses, self.solution = n, clauses, None
    def checkClause(self, v, c):
        return any( ( v[abs(lit)] if lit > 0 else not v[abs(lit)] )
                    for lit in c )
    def check(self, v):
        return all( self.checkClause(v, cl) for cl in self.clauses )
    def solve(self, i, v):
        if i >= self.n: # ohodnotili sme vsetky atomy
            if self.check(v):
                self.solution = v
                return True
            return False
        for b in [True, False]:
            v[i] = b
            if self.solve(i+1, v):
                return True
        return False
Sat(20, [[]]).solve(0, {})
```

Čas: $O(s \cdot 2^N)$, priestor: $O(s+N)$;

N — počet atómov,

s — súčet veľkostí klauzúl

Strom prehľadávania ohodnotení

$$S = \{(a \lor b), (a \lor \neg b), (\neg a \lor b), (\neg a \lor \neg b \lor \neg c), (\neg a \lor c)\}$$

× znamená $v \not\models_{\mathrm{p}} S$ $\qquad$ $f := 0, t := 1$

- $\{\}$
  - $a \mapsto 0$: $\{a \mapsto 0\}$
    - $b \mapsto 0$: $\{a \mapsto 0, b \mapsto 0\}$
      - $c \mapsto 0$: $\{a \mapsto 0, b \mapsto 0, c \mapsto 0\}$ ×
      - $c \mapsto 1$: $\{a \mapsto 0, b \mapsto 0, c \mapsto 1\}$ ×
    - $b \mapsto 1$: $\{a \mapsto 0, b \mapsto 1\}$
      - $c \mapsto 0$: $\{a \mapsto 0, b \mapsto 1, c \mapsto 0\}$ ×
      - $c \mapsto 1$: $\{a \mapsto 0, b \mapsto 1, c \mapsto 1\}$ ×
  - $a \mapsto 1$: $\{a \mapsto 1\}$
    - $b \mapsto 0$: $\{a \mapsto 1, b \mapsto 0\}$
      - $c \mapsto 0$: $\{a \mapsto 1, b \mapsto 0, c \mapsto 0\}$ ×
      - $c \mapsto 1$: $\{a \mapsto 1, b \mapsto 0, c \mapsto 1\}$ ×
    - $b \mapsto 1$: $\{a \mapsto 1, b \mapsto 1\}$
      - $c \mapsto 0$: $\{a \mapsto 1, b \mapsto 1, c \mapsto 0\}$ ×
      - $c \mapsto 1$: $\{a \mapsto 1, b \mapsto 1, c \mapsto 1\}$ ×

SAT a DPLL

Optimalizácia backtrackingu

Priebežné vyhodnocovanie klauzúl

Strom ohodnotení:

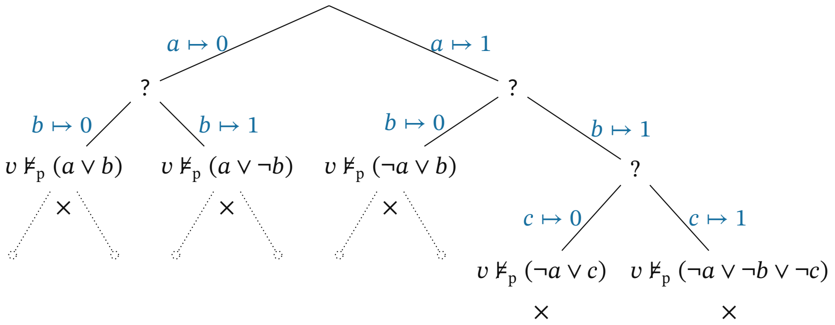
- List — ohodnotenie všetkých premenných
- Každý uzol — *čiastočné ohodnotenie*
- Ohodnotenie v uzle je *rozšírením* ohodnotenia v rodičovi
- Niektoré klauzuly sa dajú vyhodnotiť aj v čiastočnom ohodnotení
  - V čiastočnom ohodnotení $v = \{a \mapsto 0, b \mapsto 1\}$ sa dá určiť pravdivosť $(a \lor b)$, $(a \lor \neg b)$, $(\neg a \lor b)$ z našej S
- Ak nájdeme nepravdivú, môžeme hneď „backtracknúť“ — zastaviť prehľadávanie vetvy a vrátiť sa o úroveň vyššie
  - V čiastočnom ohodnotení $v = \{a \mapsto 0, b \mapsto 0\}$ je nepravdivá $(a \lor b)$ z S

Prehľadávanie s priebežným vyhodnocovaním

$$S = \{(a \lor b), (a \lor \neg b), (\neg a \lor b), (\neg a \lor \neg b \lor \neg c), (\neg a \lor c)\}$$

$\times$ znamená $v \not\models_p S$

? znamená zatiaľ žiadna nepravdivá klauzula



Zjednodušenie množiny klauzúl podľa literálu

Nech v je čiastočné ohodnotenie, v ktorom $v(a) = 1$.

V každom rozšírení ohodnotenia v:

- sú pravdivé klauzuly obsahujúce a
  - $\{a \mapsto 1, ...\} \models_{\mathrm{p}} (a \vee b)$
  - $\{a \mapsto 1, ...\} \models_{\mathrm{p}} (a \vee \neg b)$
- je pravdivá klauzula $(\ell_1 \vee \cdots \vee \neg a \vee \cdots \vee \ell_n)$ obsahujúca $\neg a$ **vtt** je pravdivá **zjednodušená** klauzulu $(\ell_1 \vee \cdots \vee \cdots \vee \ell_n)$
  - $\{a \mapsto 1, ...\} \models_{\mathrm{p}} (\neg a \vee \neg b \vee \neg c)$ vtt $\{a \mapsto 1, ...\} \models_{\mathrm{p}} (\neg b \vee \neg c)$

Takže množinu S môžeme **zjednodušiť**:

- klauzuly s a môžeme **vynechať**;
- klauzuly s $\neg a$ môžeme **zjednodušiť**.

Zjednodušenie množiny klauzúl podľa literálu

Množinu klauzúl

$$S = \{ (a \lor b), (a \lor \neg b), (\neg a \lor b), (\neg a \lor \neg b \lor \neg c), (\neg a \lor c) \}$$

môžeme ***zjednodušiť podľa*** $a \mapsto 1$ na

$$S|_{a \mapsto 1} = \{ b, (\neg b \lor \neg c), c \}.$$

Analogicky môžeme S zjednodušiť podľa $a \mapsto 0$ na

$$S|_{a \mapsto 0} = \{ b, \neg b \}.$$

Zjednodušenie množiny klauzúl podľa literálu

Definícia 6.3

Nech P je predikátový atóm a S je množina klauzúl. Potom definujeme

$$S|_{P\mapsto f} = \{\, (\ell_1 \lor \cdots \lor \cdots \lor \ell_n) \mid (\ell_1 \lor \cdots \lor P \lor \cdots \lor \ell_n) \in S \,\}$$
$$\cup \{\, c \mid c \in S, \text{ v } c \text{ sa nevyskytuje } P \text{ ani } \neg P \,\}$$
$$S|_{P\mapsto t} = \{\, (\ell_1 \lor \cdots \lor \cdots \lor \ell_n) \mid (\ell_1 \lor \cdots \lor \neg P \lor \cdots \lor \ell_n) \in S \,\}$$
$$\cup \{\, c \mid c \in S, \text{ v } c \text{ sa nevyskytuje } P \text{ ani } \neg P \,\}$$
$$S|_{\neg P\mapsto t} = S|_{P\mapsto f}$$
$$S|_{\neg P\mapsto f} = S|_{P\mapsto t}$$

Tvrdenie 6.4

Nech P je predikátový atóm a S je množina klauzúl.

Nech $b \in \{t, f\}$ a v je ohodnotenie také, že $v(P) = b$.

Potom $v \models_{\mathrm{p}} S$ vtt $v \models_{\mathrm{p}} S|_{P\mapsto b}$.

Propagácia jednotkových klauzúl

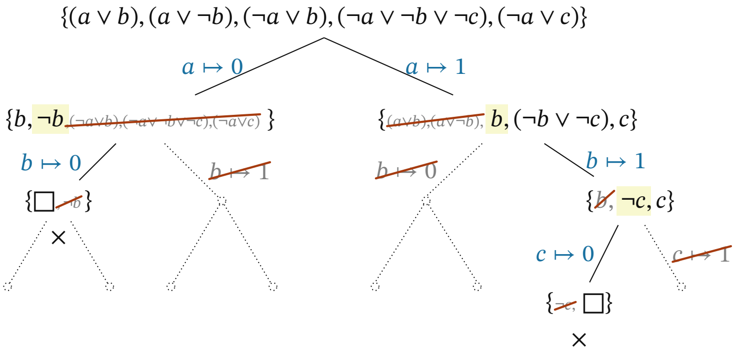
Nech $T = \{(a \lor \neg b), (a \lor b \lor c)\}$.

Začnime zjednodušením podľa $a \mapsto 0$:

- $T' := T|_{a \mapsto 0} = \{\neg b, (b \lor c)\}$
  - $\neg b$ — ***jednotková klauzula*** (*unit clause* alebo iba ***unit***)
  - T' spĺňajú **iba** ohodnotenia v, kde $v(b) = 0$
  - Takže T' zjednodušíme podľa $b \mapsto 0$
- $T'' := T'|_{b \mapsto 0} = \{c\}$
  - c — jednotková klauzula
  - T'' spĺňajú iba ohodnotenia v, kde $v(c) = 1$
  - Takže T'' zjednodušíme podľa c
- $T''' := T''|_{c \mapsto 1} = \{\}$ prázdna, pravdivá v hocijakom ohodnotení. Podľa tvrdenia 6.4:
  - T'' je pravdivá v každom ohodnotení, kde $v(c) = 1$.
  - T' je pravdivá v každom ohodnotení, kde $v(b) = 0$, $v(c) = 1$.
  - T je pravdivá v ohodnotení $v = \{a \mapsto 0, b \mapsto 0, c \mapsto 1\}$.

Prehľadávanie so zjednodušovaním klauzúl a unit propagation

Propagácia jednotkových klauzúl (*unit propagation*) je proces opakovaného rozširovania ohodnotení podľa jednotkových klauzúl a zjednodušovania.



Eliminácia nezmiešaných literálov

Všimnime si literál u v množine klauzúl:

$$T = \{(\neg a \vee \neg b \vee c), (\neg a \vee P), (\neg b \vee P), a, b, \neg c\}$$

Literál P je ***nezmiešaný*** (angl. *pure*) v T:
P sa vyskytuje v T, ale jeho komplement $\neg P$ sa tam nevyskytuje.

Nech $T' := T|_{P \mapsto 1} = \{(\neg a \vee \neg b \vee c), a, b, \neg c\}$

- Ak nájdeme ohodnotenie $v \models_{\mathrm{p}} T'$,
  tak $v_0 := v(P \mapsto 0)$ aj $v_1 := v(P \mapsto 1)$ sú modelmi T'
  a v_1 je navyše modelom T, teda T je splniteľná.
- Ak je T' nesplniteľná,
  tak je nesplniteľná každá jej nadmnožina, teda aj T.

Z hľadiska splniteľnosti sú klauzuly obsahujúce u nepodstatné.
Stačí uvažovať $T|_{P \mapsto 1}$.

Eliminácia nezmiešaných literálov

Definícia 6.5

Nech P je predikátový atóm premenná.

Komplementom literálu P je $\neg P$. *Komplementom literálu $\neg P$* je P.

Komplement literálu ℓ označujeme $\bar{\ell}$.

Definícia 6.6

Nech ℓ je literál a S je množina klauzúl.

Literál ℓ je ***nezmiešaný*** (*pure*) v S vtt ℓ sa vyskytuje v niektorej klauzule z S, ale jeho komplement $\bar{\ell}$ sa nevyskytuje v žiadnej klauzule z S.

Tvrdenie 6.7

Nech ℓ je literál a S je množina klauzúl.

Ak ℓ je nezmiešaný v S, tak S je splniteľná vtt $S|_{\ell \mapsto 1}$ je splniteľná.

SAT a DPLL

DPLL a sledované literály

DPLL

Algoritmus 6.8 (Davis and Putnam [1960], Davis et al. [1962])

def DPLL(Φ, v)**:**
    if Φ obsahuje prázdnu klauzulu**:**
        return False
    if v ohodnocuje všetky atómy**:**
        return True
    while existuje jednotková (unit) klauzula ℓ vo Φ**:**
        $\Phi, v =$ unit-propagate(ℓ, Φ, v)
    while existuje nezmiešaný (pure) literál ℓ vo Φ**:**
        $\Phi, v =$ pure-literal-assign(ℓ, Φ, v)
    $x =$ choose-branch-atom(Φ, v)
    return DPLL($\Phi|_{x \mapsto t}, v(x \mapsto t)$) **or** DPLL($\Phi|_{x \mapsto f}, v(x \mapsto f)$)

Technika sledovaných literálov (watched literals)

Aby sme nemuseli zjednodušovať množinu klauzúl:

- Pre každú klauzulu vyberieme 2 sledované literály.

  $(\neg a \lor \neg b \lor \neg c)$
- Sledovaný literál musí byť nenastavený alebo true, ak sa to dá.
- Ak sa sledovaný literál stane *true*: nič nemusíme robiť.

  $\{a \mapsto 0\} \qquad (\neg a \lor \neg b \lor \neg c)$
- Ak sa sledovaný literál stane *false*: musíme nájsť iný.

  $\{a \mapsto 1\} \qquad (\neg a \lor \neg b \lor \neg c)$

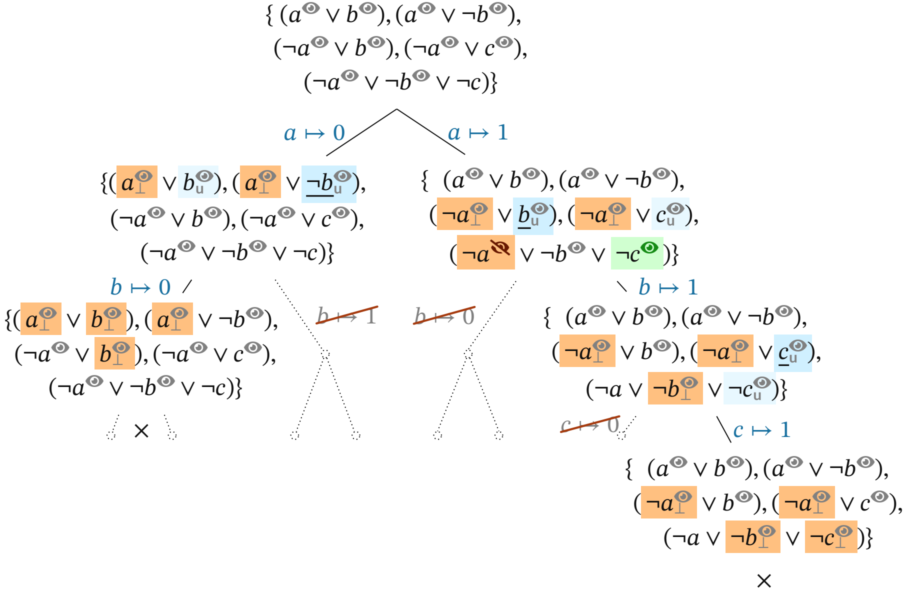
  Ak iný nie je, práve sme vyrobili jednotkovú klauzulu
  (všetky literály okrem druhého sledovaného sú *false*),

  $\{a \mapsto 1, b \mapsto 1\} \qquad (\neg a \lor \neg b_{\perp} \lor \underline{\neg c}_{\mathrm{u}})$

  alebo spor (aj druhý sledovaný je už *false*).

  $\{a \mapsto 1, b \mapsto 1, c \mapsto 0\} \qquad (\neg a_{\perp} \lor c_{\perp})$
- Keď backtrackujeme: nič nemusíme robiť
  (možno sa niektoré sledované literály stanú *nenastavenými*).

Prehľadávanie s unit propagation a sledovaním



Moderné SAT solvery:

- algoritmus DPLL:
  backtracking + propagácia jednotkových klauzúl;
- sledovanie literálov

Tento týždeň na praktických cvičeniach:
reprezentácia klauzúl, ohodnotení, sledovanie literálov

Budúci týždeň:
DPLL — propagácia jednotkových klauzúl, backtracking

Súťaž o najrýchlejší SAT solver — do konca výučby.
Bonus až 6 bodov (podľa umiestnenia)

Literatúra

Martin Davis and Hillary Putnam. A computing procedure for quantification theory. *J. Assoc. Comput. Mach.*, 7:201–215, 1960.

Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994. ISBN 978-0-201-53082-7.

Raymond M. Smullyan. *Logika prvého rádu*. Alfa, 1979. Z angl. orig. *First-Order Logic*, Berlin-Heidelberg: Springer-Verlag, 1968 preložil Svätoslav Mathé.