CURSUL #9

Tuesday, December 13, 2022 2:06 PM

UNDE SUNT INSTANȚELE "DIFICILE" PT O PROBLEMĂ NP-completă?

3-SAT Parametru $c > 0$

Generez instanțe aleatoare pt 3-SAT cu parametrul c

$$\begin{cases} \#\text{variabile} = n \\ \#\text{clauze} = c \cdot n \end{cases}$$ ← aleg $c \cdot n$ clauze la întâmplare

$8\binom{n}{3}$ = nr total de clauze posibile peste n variabile

fiecare variabilă x; $\bar{x}$ — 3 variabile din n

INTUIȚIE Multe probleme NP-complete au tranziție de fază

Cele mai "grele" cazuri se întâlnesc la tranziția de fază → ALGORITMUL ?!

(DP) **DPLL** Davis – Putnam Logemann – Loveland

→ alg. pt SAT (backtracking)

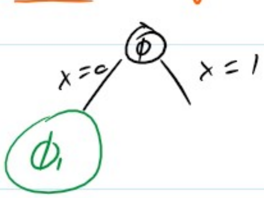
completi → răspunsul DA/NU este întotdeauna corect

SAT $\phi(x_1, \ldots x_n)$ formă normală conjunctivă (CNF)

$$\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$$

clauză

Clauză $\bar{x} \vee y \vee \bar{z}$

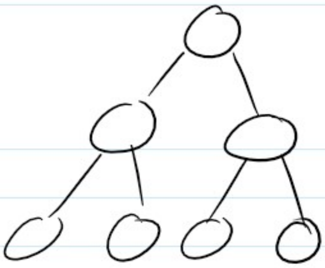


eliminate

$$\phi \begin{bmatrix} x \vee \bar{y} \vee \bar{z} \\ \bar{x} \vee y \vee t \\ \bar{x} \vee \bar{y} \vee z \\ z \vee \bar{t} \vee \bar{\xi} \end{bmatrix}$$

$$\phi_1 = \begin{Bmatrix} \bar{y} \vee \bar{z} \\ z \vee \bar{t} \vee \bar{\xi} \end{Bmatrix}$$

- Creez arbore de backtracking.
  - Aleg variabila după care fac backtracking a.î. să îi pot da val. corectă

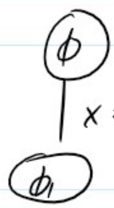


SITUAȚII ÎN CARE VALOAREA CORECTĂ ESTE CLARĂ

① x literal pur

x=0 satisface toate clauzele în care apar $x, \bar{x}$

$$\phi = \begin{Bmatrix} \bar{x} \vee \bar{y} \vee \bar{z} \\ \bar{x} \vee \bar{t} \vee \bar{z} \\ a \vee \bar{b} \vee c \end{Bmatrix}$$

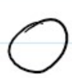


Def x literal pur dacă x apare doar pozitiv sau doar negativ în ϕ

REGULA 1 Dacă x literal pur îl aleg pe x ca variabilă și nu fac backtracking după x.

② x literal unitar.

$$C_1 \quad \frac{x \vee \bar{y} \vee \bar{z}}{\begin{matrix} y \vee C_1 \\ z \vee C_2 \end{matrix}}$$



$x \vee \cancel{\bar{y}} \vee \cancel{\bar{z}}$

$C_1 \rightarrow C_1' = \{x\}$

$\emptyset_{z=1}$

$\emptyset$

$C_1 \to C_1' = \{x\}$

$\boxed{x = TRUE}$

Dacă prin simplificări rezultă o clauză de lungime 1 val. variabilei din clauză unic definită

REGULA 2 Dacă vreo variabilă apare într-o clauză unitară (de lungime 1) → $\boxed{\text{val. bine definită}}$

{DPLL} backtracking
+
clauze pure
+
clauze unitare

tranziția de fază → algoritmul de tip DPLL

Sat solvere

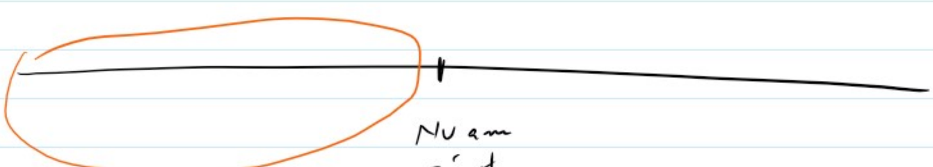
minisat
glucose
lingeling.

Bazate pe DPLL

Idee fundamentală pt accelerarea alg. de tip DPLL

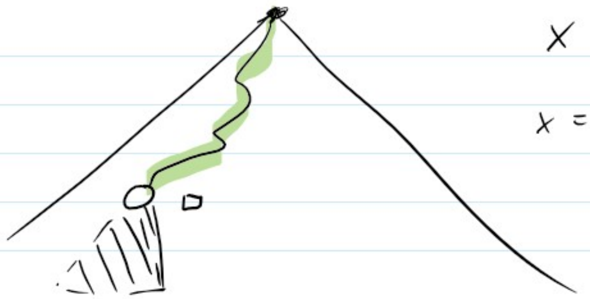
CDCL

"conflict-driven clause learning"



Nu am găsit soluție!

Pot învăța ceva din asta?



$$\left.\begin{array}{l} X \\ x = \text{FALSE} \end{array}\right\} \to \bot$$

$$\underbrace{x = T \quad y = T \quad z = F \quad t = T}$$

$$\downarrow$$

creează o contradicție

Pot învăța faptul că orice sol pt ϕ satisface

$$C' = X \vee y \vee \bar{z} \vee t$$

$$\Downarrow$$

adaug C' la ϕ când explorez restul arborelui de backtracking.

<u>Concluzii</u>

1. Alg. practici pt SAT-solving bazați pe paradigma DPLL

2. <u>Clause learning</u> → (CDCL)
   principala idee experimentată

3. Multe instanțe de SAT care vin "din practică" pot fi rezolvate cu solverele actuale

   (chiar dacă $P \neq NP$)

4. Există clase de formule pe care metodele de tip DPLL <u>nu</u> scalează bine

4. Există clase de formule pe care metodele de tip DPLL **nu** scalează bine (polinomial)

DE CE EXISTĂ CLASE DE FORMULE PE CARE DPLL nu merg bine ?

Exemplu (Principiul Cutiei / principiul lui Dirichlet)

(n) porumbei în n-1 căsuțe, $\rightarrow$ imposibil fără a avea 2 porumbei în aceeași căsuță.

Formula propozițională PHP_n^{n-1} nesatisfiabilă

exprimă

{Până de curând $n=15$ prea mult pt solvere actuale.}

$PHP_{n,n-1}$

$$X_{i,j} = \begin{cases} 1 & i \rightarrow j \\ 0 & \text{altfel} \end{cases} \quad i=\overline{1,n} \quad j=\overline{1,n-1}$$

Exp $n=15$ $\underbrace{15 \times 14}_{210}$ variabile

- $X_{i,1} \vee X_{i,2} \ldots \vee X_{i,n-1}$ por i merge cel puțin într-o căsuță
- $\overline{X}_{i,a} \vee \overline{X}_{i,b}$ por i nu merge simultan în a, b

$i=\overline{1,n}$

$1 \leq a < b \leq n-1$

- $\overline{X}_{i,a} \vee \overline{X}_{j,a}$ por i și j nu merg ...

$$\left\lfloor \; - \quad \bar{X}_{i,a} \vee \bar{X}_{j,a} \right.$$

pt. i și j nu merg simultan în aceeași căsuță a.

$1 \le i < j \le n$

$1 \le a \le n-1$

$$\boxed{PHP_n^{n-1} \notin SAT}$$

Timpul de rulare $DP(PHP_n^{n-1})$ scalează exponențial cu n.

<u>Explicație</u> COMPLEXITATEA DEMONSTRAȚIILOR PRIN REZOLUȚIE

<u>Rezoluție</u>

$$\frac{x \vee C_1 \qquad \bar{x} \vee C_2}{C_1 \vee C_2}$$

(T) $\phi \notin SAT \iff \phi \vdash_{\text{rezoluție}} \square$

<u>Mă interesează</u>

(ϕ_n) Exp. (PHP_n^{n-1})

L_n = # clauze în cea mai scurtă demonstrație de nesatisfiabilitate prin rezoluție

C_1, C_2 ... C_m ... $\text{rez}(C_i, C_j)$ ... $-$ ... Π

$$\underbrace{C_1 \; C_2 \; \ldots \; C_m \quad \mathrm{rez}(C_i, C_j) \; \ldots \; \square}_{L_n}$$

T $\exists\, c > 0$ a.î. $\boxed{L_n(PHP_n^{n+1}) \geq 2^{c \cdot n}}$

("complexitatea demonstrațiilor")

$\Downarrow$

DPLL?

T $L_n \leq$ running time pt oricare alg. de tip DPLL

Dem.

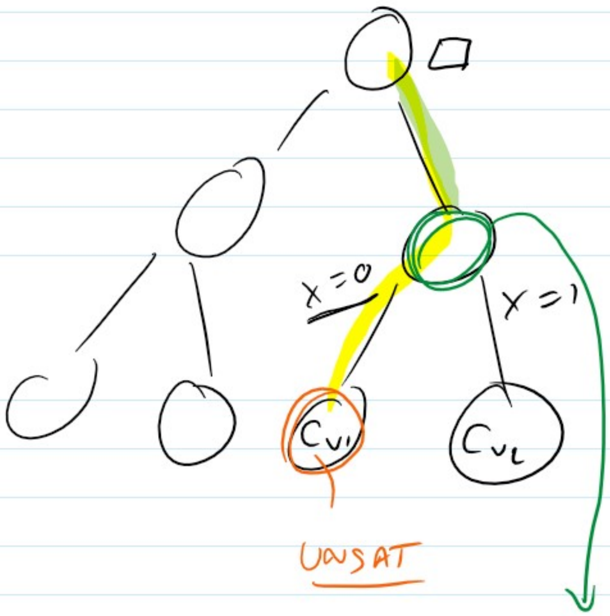
IDEEA Iau un arbore de backtracking pt

$A(\phi_n)$

$\Downarrow$

Construiesc o dem de nesatisfiabilitate pt ϕ_n prin rezoluție

$\boxed{\#\text{ clauze} = \#\text{ noduri în arb. backtracking}}$



Pt fiecare nod frunză v

există o clauză C_v în ϕ

nesatisfăcută de sol. respectivă

$$C_{v_1} = x \vee D_1$$

$$C_{v_2} = \bar{x} \vee D_2$$

clauză $D_1 \vee D_2$ $(res(C_{v_1}, C_{v_2}))$

nesatisfăcută de

În felul ăsta am generat o dem. de nesat. prin rezoluție