



Algorithmen & Datastrukturen 2

WPO – Conflict-driven Clause Learning

DPLL maakt regelmatig “dezelfde fouten”

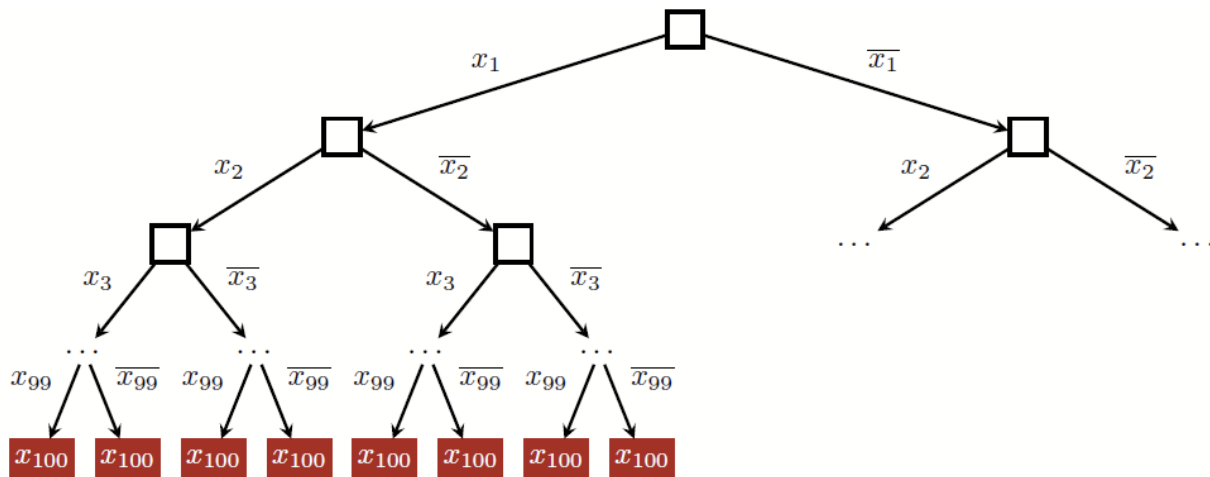
Conflicten kunnen herhaaldelijk terugkeren!

$$\overline{x_1} \vee x_{99} \vee x_{100}$$

$$\overline{x_1} \vee \overline{x_{99}} \vee x_{100}$$

$$\overline{x_1} \vee x_{99} \vee \overline{x_{100}}$$

$$\overline{x_1} \vee \overline{x_{99}} \vee \overline{x_{100}}$$



`cnf-samples/example-19_18.cnf`

Resolutie to the rescue

- Nieuwe clauses afleiden op basis van 2 bestaande clauses in de formule
 - Zie ook cursus Artificiële Intelligentie
- Formeel: Concreet:

$$\frac{c \cup \{l\} \quad c' \cup \{\bar{l}\}}{c \cup c'}$$

$$\frac{a \vee b \vee c \quad d \vee e \vee \bar{c}}{a \vee b \vee d \vee e}$$

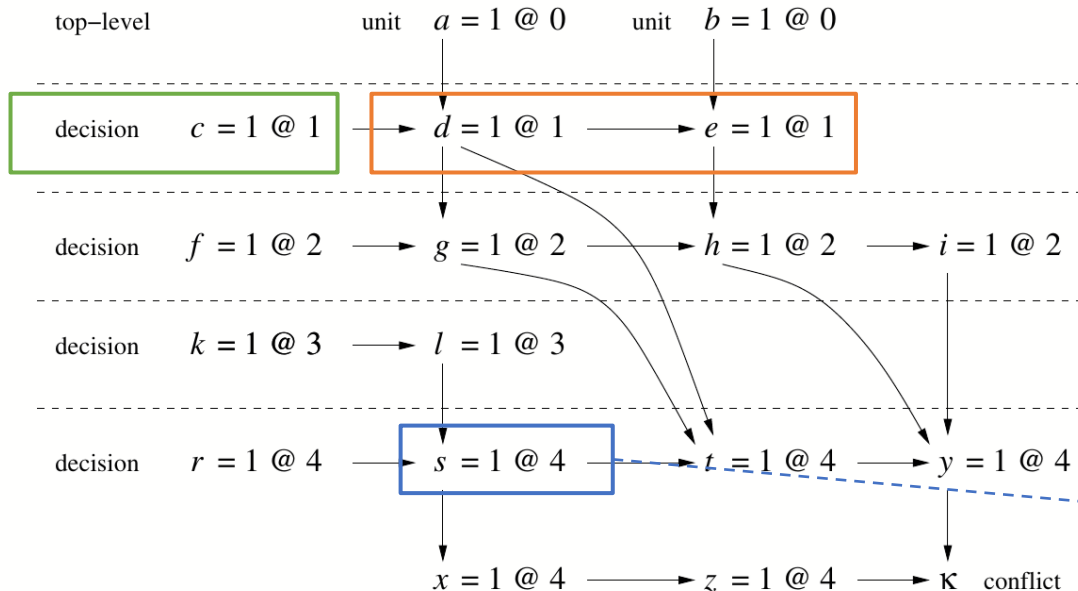
Resolutie in Scheme



```
(define (resolve cls1 cls2 l not-l)
  (let ((merged-lits (remove-duplicates (append (remap (lambda (lit)
                                                         (same-literal? lit l))
                                                         (literals cls1))
                                                         (remap (lambda (lit)
                                                         (same-literal? lit not-l))
                                                         (literals cls2)))
                                                         same-literal?))))
    (make-clause merged-lits (length merged-lits))))
```

Implicatiegraaf

Visuele voorstelling van CDCL's zoekproces



a

b

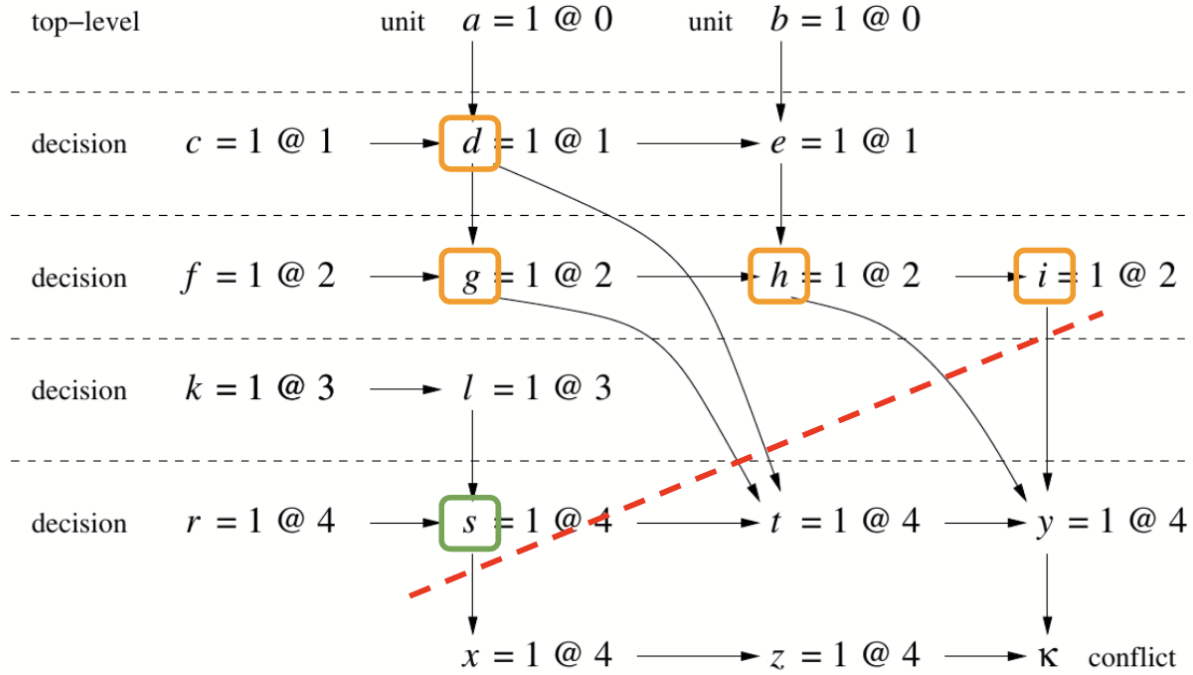
$\bar{a} \vee \bar{c} \vee d$

$\bar{b} \vee \bar{d} \vee e$

...

Eerste unieke implicatiepunt, 1-UIP

1-UIP Clause



$$\bar{d} \vee \bar{g} \vee \bar{h} \vee \bar{i} \vee \bar{s}$$

Deze clause vervat het conflict

Conflict-analyse d.m.v. resolutie

Conflicterende clause resoluten met de reden van propagatie van de “jongste” literal

Bijkomende boekhouding nodig voor “tijdstip” dat een literal propageerde

1 ANALYZECONFLICT(c_{confl}):

Input : A Conflicting clause c_{confl}

Output: The 1UIP clause

2 while c_{confl} contains at least two literals from the current decision level
do

3 | $\ell \leftarrow$ the youngest literal in c_{confl} ;

4 | $c_{confl} \leftarrow (c_{confl} \cup \bar{\ell}.reason) \setminus \{\ell, \bar{\ell}\}$;


5 end

6 return c_{confl} ;

Backjumpen

Nadat de 1UIP-clause is geleerd moet er teruggesprongen worden naar het niveau waar deze kan propageren

Wat als de geleerde clause
maar 1 literal heeft?



```
1 BACKJUMP(c):  
   Input : A conflicting clause c  
2 lev ← the level of the second youngest literal in c;  
3 while assignmentStack.TOP().level > lev do  
4   | ℓ ← assignmentStack.POP();  
5   | ℓ.reason ← nil;  
6   | ℓ.value ← u;  
7   | ℓ.level ← nil;  
8 end  
9 currentDecisionLevel ← l;  
10 ℓ ← the only unassigned literal in c;  
11 MAKETRUE(ℓ, c);  
12 return;
```


Standaard CDCL implementatie

Zie [a-d/sat/cdc1/standard.rkt](#)

Alternatieve conflict-analyse

Niet meer zoeken naar de jongste literal

Doet al deels backjumping

Moet \bar{l} zijn!

```
1 ANALYZECONFLICT'(cconf):  
  Input  : A Conflicting clause cconf  
  Output: The 1UIP clause  
2 while cconf contains at least two literals from the current decision level  
  do  
3   |  $\ell \leftarrow \text{assignmentStack.POP}();$   
4   | if  $\ell \in c$  then  
5   |   |  $c_{\text{conf}} \leftarrow c_{\text{conf}} \cup \bar{\ell}.\text{reason} \setminus \{\ell, \bar{\ell}\};$   
6   | end  
7   |  $\ell.\text{reason} \leftarrow \text{nil};$   
8   |  $\ell.\text{value} \leftarrow \mathbf{u};$   
9   |  $\ell.\text{level} \leftarrow \text{nil};$   
10 end  
11 return cconf;
```

Opdracht

Alternatieve conflict-analyse voor CDCL implementeren

- Standaard CDCL bestuderen
- Zie pseudocode vorige slide of cursustekst
- Voorzie bestand
`a-d/sat/cdcl/alternative.rkt`

Test jouw implementatie via `test.rkt`

Zie bijgeleverde folder `cnf-samples`

- `example-19_18.cnf`
- `example-19_21.cnf`
- `kit-a3-ex1.cnf`

Denkvraagje (extra oefening)

- CDCL gebruikt Quicksort om de 2e jongste literal te bepalen in het backjumpen
 - Kan dit niet anders aangepakt worden?