

Algorithmisches Beweisen LAB

CDCL

Kaspar Kasche

FSU Jena

30.05.2024

- Implementierung von SAT-Lösern
 - 2-SAT
 - Hornformeln
 - DPLL
 - CDCL
 - clause learning
 - **watched literals**
 - decision heuristics
 - restart strategy

CDCL Pseudocode

Eingabe: KNF φ

```
1: decision-level  $\leftarrow 0$ 
2: while Es existieren nicht belegte Variablen do
3:   decision-level++
4:   decide()
5:    $C_{\text{conflict}} \leftarrow \text{propagate}()$ 
6:   while  $C_{\text{conflict}}$  is not null do
7:     if decision-level = 0 then return UNSAT
8:     end if
9:      $C_{\text{learned}} \leftarrow \text{analyze-conflict}(C_{\text{conflict}})$ 
10:     $\varphi \leftarrow \varphi \wedge C_{\text{learned}}$ 
11:    backtrack( $C_{\text{learned}}$ )
12:     $C_{\text{conflict}} \leftarrow \text{propagate}()$ 
13:  end while
14:  apply-restart-policy()
15: end while
```

Watched Literals

- Verbessertes Verfahren für Unit-Propagation
- Bei Konfliktsuche sind nur Unit-Klauseln relevant
- Klausel ist Unit, wenn
 - ein Literal ist nicht belegt **und**
 - alle anderen Literale sind falsch
- Neue Idee: Genügt, zwei Literale pro Klausel zu betrachten
- Folgende Invariante muss immer gelten:
 - Entweder beide angeschauten Literale sind nicht belegt,
 - oder mindestens eins der beiden ist erfüllt.
- Wichtig: Falls beide Literale belegt sind, und eins ist falsch:
Dessen decision-level darf nicht niedriger sein als das erfüllte.

- Liste aller angeschauten Klauseln pro Literal
- Queue aller belegter und nicht bearbeiteter Variablen
- Bei Belegung einer Variable:
 - Betrachtung aller angeschauten Klauseln für das negierte Literal
 - Sicherstellen, dass Invariante gilt
 - Falls Klausel Unit wird, hinzufügen des Literals in Queue
- Sollte Invariante nicht mehr erfüllbar sein: Konflikt!

Beispiel

(Notation: x unwatched, \bar{x} watched)

$$(\bar{x}_1 \vee x_2 \vee \neg \bar{x}_3) \wedge (\bar{x}_1 \vee \neg \bar{x}_2) \wedge (\neg \bar{x}_1 \vee \neg \bar{x}_3), \quad \{\}$$

Entscheidung: $x_3 \mapsto 1$

$$(\bar{x}_1 \vee x_2 \vee \neg \bar{x}_3) \wedge (\bar{x}_1 \vee \neg \bar{x}_2) \wedge (\neg \bar{x}_1 \vee \neg \bar{x}_3), \quad \{x_3\}$$

$$(\bar{x}_1 \vee \bar{x}_2 \vee \neg \bar{x}_3) \wedge (\bar{x}_1 \vee \neg \bar{x}_2) \wedge (\neg \bar{x}_1 \vee \neg \bar{x}_3), \quad \{\neg x_1\}$$

$$(\bar{x}_1 \vee \bar{x}_2 \vee \neg \bar{x}_3) \wedge (\bar{x}_1 \vee \neg \bar{x}_2) \wedge (\neg \bar{x}_1 \vee \neg \bar{x}_3), \quad \{\neg x_1\}$$

$$(\bar{x}_1 \vee \bar{x}_2 \vee \neg \bar{x}_3) \wedge (\bar{x}_1 \vee \neg \bar{x}_2) \wedge (\neg \bar{x}_1 \vee \neg \bar{x}_3), \quad \{x_2, \neg x_2\}$$

Widerspruch!

Aufgabe: CDCL

- Implementierung der Watched Literals
- Vergleichen Sie die Performance mit einfacher propagation
- Ausgabe einiger Statistiken:
 - Zeit
 - Speicherbedarf
 - Anzahl Unit Propagations
 - Anzahl Entscheidungen
 - Anzahl Konflikte
 - etc.