Algorithmisches Beweisen LAB

CDCL

Luc Spachmann

FSU Jena

30.05.2022

Ziele

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

CDCL Pseudocode

Eingabe: KNF φ

```
 1: decision-level ← 0
 2: while Es existieren nicht belegte Variablen do
 3:     decision-level++
 4:     decide()
 5:     C_conflict ← propagate()
 6:     while C_conflict is not null do
 7:         if decision-level= 0 then return UNSAT
 8:         end if
 9:         C_learned ← analyze-conflict(C_conflict)
10:         φ ← φ ∧ C_learned
11:         backtrack(C_learned)
12:         C_conflict ← 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.

Ideen zur Implementierung

- 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

$$(\overline{x_1} \vee x_2 \vee \overline{\neg x_3}) \wedge (\overline{x_1} \vee \overline{\neg x_2}) \wedge (\overline{\neg x_1} \vee \overline{\neg x_3}), \quad \{\}$$

Entscheidung: $x_3 \mapsto 1$

$$(\overline{x_1} \vee x_2 \vee \overline{\neg x_3}) \wedge (\overline{x_1} \vee \overline{\neg x_2}) \wedge (\overline{\neg x_1} \vee \overline{\neg x_3}), \quad \{x_3\}$$

$$(\overline{x_1} \vee \overline{x_2} \vee \neg x_3) \wedge (\overline{x_1} \vee \overline{\neg x_2}) \wedge (\overline{\neg x_1} \vee \overline{\neg x_3}), \quad \{\neg x_1\}$$

$$(\overline{x_1} \vee \overline{x_2} \vee \neg x_3) \wedge (\overline{x_1} \vee \overline{\neg x_2}) \wedge (\overline{\neg x_1} \vee \overline{\neg x_3}), \quad \{\neg x_1\}$$

$$(\overline{x_1} \vee \overline{x_2} \vee \neg x_3) \wedge (\overline{x_1} \vee \overline{\neg x_2}) \wedge (\overline{\neg x_1} \vee \overline{\neg x_3}), \quad \{x_2, \neg x_2\}$$

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.