Algorithmisches Beweisen LAB

DPLL

Luc Spachmann

FSU Jena

09.05.2022

Ziele

- Implementierung von SAT-Lösern
  - 2-SAT
  - Hornformeln
  - DPLL
  - CDCL (Schrittweise)

DPLL

- Davis–Putnam–Logemann–Loveland Algorithmus
- Lösen beliebiger k-KNF
- Rekursiver Algorithmus

Pure Literals

$$(x \vee \neg y) \wedge (x \vee y \vee \neg z) \wedge (z)$$

- Ein Literal l heißt *pur* in einer KNF φ falls:
    - l kommt φ vor
    - $\neg l$ kommt nicht in φ vor
- Falls φ erfüllbar ist, so hat es eine erfüllende Belegung, welche alle puren Literale erfüllt
- Diese können also direkt belegt werden

Wiederholung: Unit Propagation

$$(\neg x_1) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5)$$

- Eine Unit-Klausel beinhaltet genau ein Literal $(\neg x_1)$
- Eine Unit-Belegung erfüllt eine Unit-Klausel: $x_1 \mapsto 0$
- Unit Propagation ist eine sukzessive und vollständige Anwendung von Unit-Belegungen

Algorithmus

Eingabe: KNF φ

$\varphi \leftarrow$ unit-propagate(φ)
$\varphi \leftarrow$ pure-literal-elimination(φ)
if φ ist die leere Formel **then**
    return true
end if
if φ enthält die leere Klausel **then**
    return false
end if
$x \leftarrow$ get-decision-variable(φ)
return DPLL($\varphi[x \mapsto 0]$) or DPLL($\varphi[x \mapsto 1]$)

Anmerkungen

- “get-decision-variable“ gibt beliebige nicht gesetzte Variable zurück
- Bei Rückgabe “false“ muss φ identisch zu Funktionsanfang sein
- Wie werden Einschränkungen gespeichert?
    - Änderung der Formel direkt
    - Formel konstant und zusätzliche Speicherung der Belegung (empfohlen)

Aufgabe: DPLL

- Implementierung des Algorithmus für DPLL
- Testen des Programms anhand zufälliger k-SAT Formeln. Wie entwickelt sich die Laufzeit bei Veränderung der Parameter?
- Ausgabe einiger Statistiken:
    - Zeit
    - Speicherbedarf
    - Anzahl Unit Propagations
    - Anzahl Entscheidungen
    - etc.