SAT-Solving und Anwendungen

CDCL SAT-Solving

Prof. Dr. Wolfgang Küchlin
Dipl. Inf. Christoph Zengler

Universität Tübingen

26. April 2011

Aktivitätsheuristiken

Aktivitätsheuristiken

Wähle die Variable, die am aktivsten ist.

- Aktivität kann verschieden definiert sein
- Häufig: Variable, die kürzlich in vielen Konflikten (empty clauses) vorkam

Idee:

- Variablen, die oft in Konflikten vorkommen, spielen eine herausragende Rolle und sollten zuerst belegt werden
- Variablen haben verschieden großen Einfluss auf eine Formel (Bsp: Softwareverifikation, Variable die über einen großen `if-then-else`-Branch entscheidet beeinflusst Ergebnis mehr als andere Variablen)

Beispiel für Aktivitätsheuristik:

- **VSIDS** (variable state independent decaying sum)

VSIDS

Berechnung:

- Jede Variable bekommt eine *Aktivität act* zugewiesen
- Initial ist *act* Anzahl der Vorkommen der Variable (positiv + negativ)
- Für jede gelernte Klausel $(x_1 \vee x_2 \vee ... \vee x_n)$: erhöhe *act* für alle x_i um konstanten Betrag c (1)
- Teile periodisch alle Aktivitäten durch einen Faktor f (2)
- Wähle Variable mit höchster Aktivität

Erklärung:

(1) Aktivität einer Variable wird höher, je häufiger sie in Konflikten auftaucht

(2) Aktivität aller Variablen wird von Zeit zu Zeit verringert, d.h. Konflikte die in der nahen Vergangenheit aufgetreten sind, werden bevorzugt

Damit:

- Wahl der Variablen, die **aktuell** (im aktuellen Suchbaum) am **häufigsten in Konflikten** vorkam

Erweiterung der Aktivität auf Klauseln (Clause Deletion)

Problem:

- Während eines CDCL Durchgangs werden viele neue Klauseln gelernt
- Viele dieser Klauseln werden im Folgenden nicht mehr benutzt

$\rightarrow$ Klauselmenge bläht sich unnötig auf

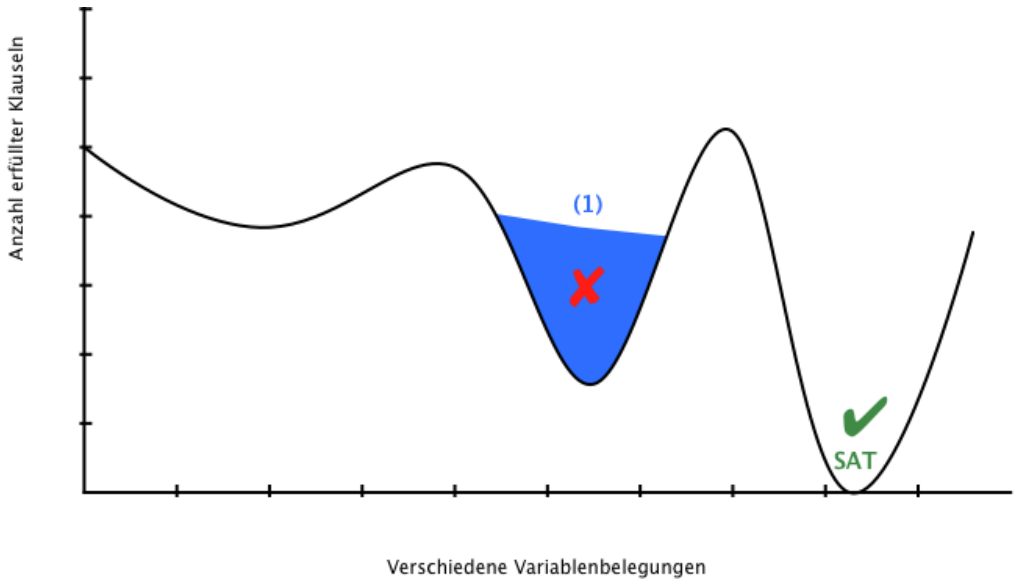
Lösung:

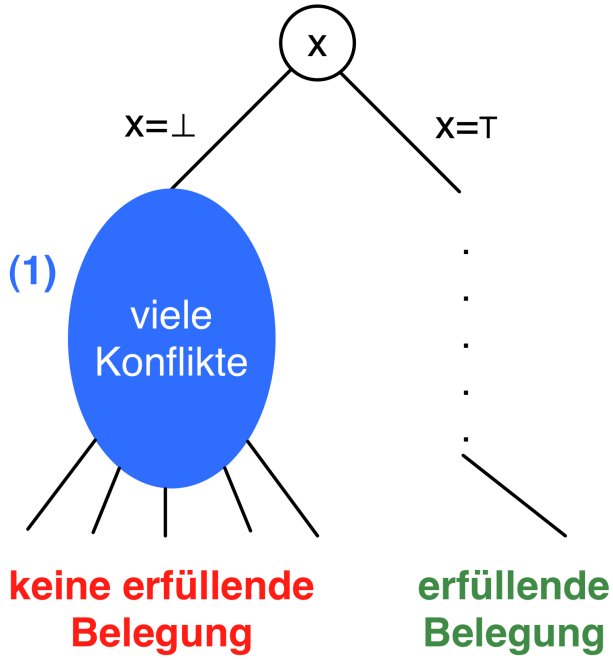
- Bewerte neu gelernte Klauseln ebenfalls mit Aktivität
- Initiale Aktivität einer neu gelernten Klausel ist eine Konstante c
- Jedesmal, wenn die Klausel in der Berechnung einer neuen Klausel vorkommt (d.h. in der Resolution benutzt wird), wird die Aktivität erhöht
- Periodisch werden alle Aktivitäten verringert
- Periodosch werden alle Klauseln mit geringer Aktivität (unter einem vorher bestimmten Schwellenwert) gelöscht

Effekt:

- Wird eine neu gelernte Klausel nie oder selten benutzt, wird sie nach einer gewissen Zeit wieder gelöscht

Gefahr von lokalen Minima





Lokale Minima

Mit VSIDS besteht die Gefahr des „Festfressens“ in einem lokalen Minimum (1):

- Heuristik wählt immer aktuelle Konfliktvariablen aus, jedoch besteht ein größerer Konflikt ganz am Anfang des Suchbaums
- Bei einer großen Anzahl von Variablen kann ein Verlassen des lokalen Minimums sehr lange dauern

Lösung für lokale Minima: Restarts

Restart

Der SAT Solver wird zurückgesetzt und neu gestartet, d.h.

- Alle Variablenbelegungen werden rückgängig gemacht
- Alle Aktivitäten von Variablen und Klauseln werden auf initialen Wert gesetzt
- **Aber:** Gelernte Klauseln bleiben erhalten

- Kriterium für Restart: Anzahl der gelernten Klauseln
- Nach bestimmter Anzahl von gelernten Klauseln erfolgt Restart
- Um Termination zu gewährleisten: Erhöhen dieses Restart-Schwellenwertes nach jedem Restart
- D.h. 1. Restart nach 100 gelernten Klauseln, 2. Restart nach 150, 3. Restart nach 300, usw...

Zusammenfassung

Alles, was man für einen effizienten state-of-the-art SAT Solver braucht (so zu finden in z.B. MiniSAT):

1. Klauselllernen
2. Nicht-chronologisches Backtracking
3. Gute Auswahlheuristiken (z.B. VSIDS)
4. Effiziente Unit Propagation (watched literals - dazu später mehr)
5. Restarts mit Clause Deletions

Fazit:

- Im Gegensatz zu anderen Problemen (z.B. lineare Optimierung mit Simplex) kann SAT Solving auf wenige Kernpunkte reduziert werden
- Einer der besten aktuellen SAT Solver - MiniSAT - kommt mit < 1000 Zeilen Code aus