SAT-Solving und Anwendungen
Auswahlheuristiken

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

Universität Tübingen

10. Juni 2009

Bereits bekannt

- $numpos(x)$ = Anzahl der positiven Vorkommen der Variable x in unerfüllten Klauseln
- $numneg(x)$ = Anzahl der negativen Vorkommen der Variable x in unerfüllten Klauseln

DLCS - Dynamic Largest Combined Sum

Wähle als nächstes Variable x mit der größten Summe:
$numpos(x) + numneg(x)$

DLIS - Dynamic Largest Individual Sum

Wähle als nächste Variable x mit dem größten $numpos(x)$ oder $numneg(x)$

Idee von DLCS und DLIS: Variablen die besonders oft vorkommen haben mehr Einfluss auf das Gesamtergebnis als Variablen, die selten vorkommen.

Bereits bekannt (ctd.)

MOM - **M**aximum **o**ccurrence in clauses of **m**inimal size

Wähle die Variable, deren Vorkommen in kurzen Klauseln maximal ist.

Idee von MOM:

- Kurze Klauseln haben mehr Aussagekraft als lange Klauseln, da sie schneller zu UP und somit zu neuen Variablenbelegungen führen
- Variablen die besonders oft in kurzen Klauseln vorkommen haben großen Einfluss auf das Gesamtergebnis

Berechnung:

- l minimale Größe einer unerfüllten Klausel
- $numpos(x)_l$, $numneg(x)_l$ Anzahl der positiven/negativen Vorkommen der Variable x in unerfüllten Klauseln von minimaler Länge x
- Wähle Variable mit maximalem $(numpos_l + numneg_l) \cdot 2^k + (numpos_l \cdot numneg_l)$
- k beliebig gewählter großer Wert, so dass der erste Term den zweiten dominiert
- Damit Präferenz: Variable mit maximalem Vorkommen in Klauseln minimaler Länge $(numpos_l + numneg_l)$, gibt es mehrere mit gleichem Vorkommen, so wähle diejenige mit der kleinsten Differenz zwischen $numpos_l$ und $nummeg_l$ (2. Term, das Quadrat ist das Rechteck mit der größten Fläche)

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 einen *score 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 scores durch einen Faktor f (2)
- Wähle Literal mit höchstem score

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

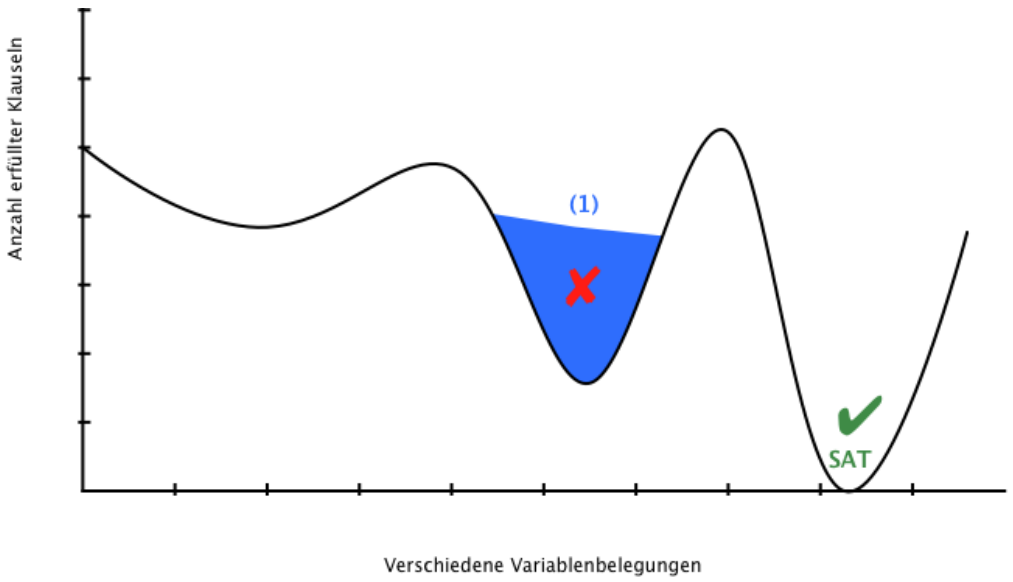
Lösung:

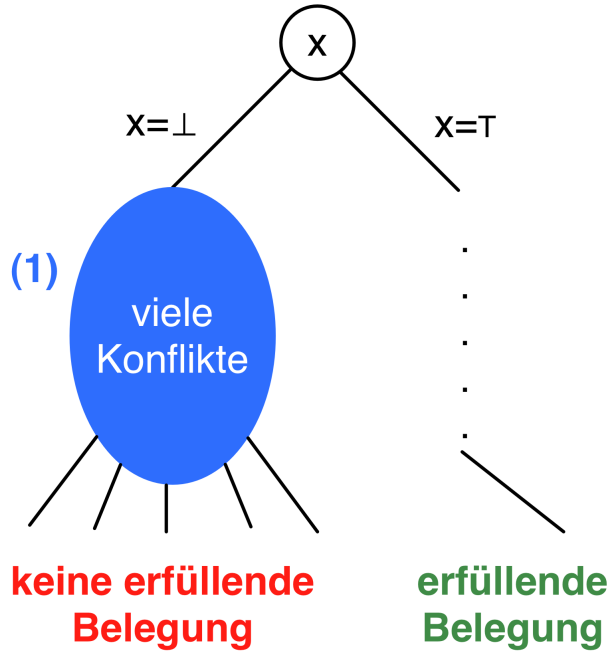
- Bewerte neu gelernte Klauseln ebenfalls mit score
- Initialer score einer neu gelernten Klausel ist eine Konstante c
- Jedesmal, wenn die Klausel in der Berechnung einer neuen Klausel vorkommt (d.h. in der Resolution benutzt wird), wird score erhöht
- Periodisch werden alle scores verringert
- Periodosch werden alle Klauseln mit geringer score (unter 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

CDCL wird resettet 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. nach 150, 3. 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 - siehe Übung)
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 aktuelln SAT Solver - MiniSAT - kommt mit < 3000 Zeilen Code aus