



Formale Systeme, WS 2008/2009

Praxisaufgabe 1: SAT Solver – Light Up lösen mit MiniSat

Abgabe der Lösungen bis zum 31.12.2008 über die Webseite zur Vorlesung

<http://i12www.ira.uka.de/~beckert/Lehre/Formale-Systeme/>

Für die vollständige Lösung dieser Praxisaufgabe erhalten Sie 1,5 Bonuspunkte für die Abschlussklausur (bitte beachten Sie die Erläuterung zu Bonuspunkten auf der Webseite zur Vorlesung).

Das Puzzle Light Up

Das Logik-Puzzle „Light Up“ ist eine NP-vollständige¹ Denksportaufgabe, die wir in dieser Praxisaufgabe mit Hilfe eines aussagenlogischen SAT(isfiability) Solvers lösen wollen.

Die Aufgabe in Light Up besteht darin, auf einem quadratischen Spielbrett Lampen so auf den Spielfeldern anzubringen, dass alle Felder ausgeleuchtet werden, aber zwei Lampen sich nie gegenseitig beleuchten. Auf dem Spielfeld sind Wände (*blocks*) platziert, die die Reichweite der Lampe beschränken. Auch können Wände mit einer Randbedingung versehen werden, wie viele Lampen in ihrer Umgebung aufgestellt werden müssen. Formaler beschrieben:

Spezifikation: Gegeben ist ein schachbrettartiges Spielbrett der Seitenlänge N . Die Felder des Brettes sind zu Beginn entweder *leer* oder *mit einer Wand* belegt. Wände sind entweder nicht gekennzeichnet oder mit einer Ziffer zwischen 0 und 4 markiert.

Platzieren Sie auf die leeren Felder Lampen, so dass gilt:

- Zu jedem anfänglich leeren Feld gibt es eine Lampe, die in derselben Zeile oder derselben Spalte steht, ohne dass eine Wand zwischen dem Feld und der Lampe liegt.
- Es gibt keine zwei Lampen, die in der selben Zeile oder Spalte stehen, ohne dass eine Wand zwischen ihnen liegt.
- Jede mit einer Ziffer z markierte Wand grenzt *genau* an z mit Lampen besetzte Felder. Zwei Felder grenzen aneinander, wenn sie nebeneinander oder übereinander liegen (jedoch nicht diagonal zu einander).

Abbildung 1 zeigt ein Beispiel für ein Spielbrett der Größe $N = 5$. Neben der Ausgangssituation (a) ist eine Situation mit einer platzierten Lampe (b), sowie eine Lampenplatzierung, die alle Bedingungen aus der Spezifikation erfüllt, dargestellt.

Der SAT Solver MiniSat

Für diese Aufgabe verwenden wir das Werkzeug MiniSat², da es einerseits ein wegen seiner Leistung preisgekrönter SAT Solver und andererseits ein sehr simples Werkzeug mit einer überschaubaren Schnittstelle ist.

Auf der Homepage von MiniSat können Sie die aktuelle Version³ herunterladen und übersetzen. Al-

¹Siehe dazu <http://www.cs.umass.edu/~mcpheilb/papers/2005lightup.pdf>

²<http://minisat.se/>

³<http://minisat.se/downloads/minisat2-070721.zip>

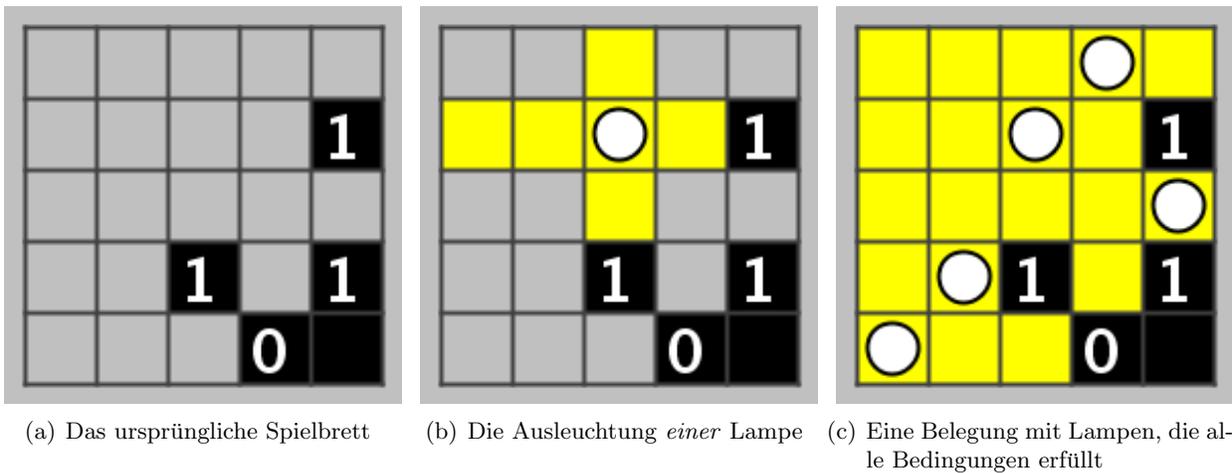


Abbildung 1: Ein Beispiel-Spielbrett für $N = 5$

ternativ dazu steht Ihnen auf den Studentenrechnern der ATIS eine Installation dieses Praxistools zur Verfügung. Details hierzu finden Sie am Ende dieses Dokuments.

Eingabeformat für MiniSat

MiniSat benutzt für seine Ein- und Ausgabe das sogenannte DIMACS-Format, das auf sehr simple Weise die Formulierung (großer) aussagenlogischer Probleme in Klauselform erlaubt.

Dieses Format besteht aus einer Kopfzeile und mehreren Klauselzeilen. Die Kopfzeile enthält neben Schlüsselwörtern (“p cnf” für “problem in conjunctive normal form”) die Anzahl der verwendeten AL-Variablen und die Zahl der Klauseln. Es folgen die Klauselzeilen, von denen jede aus einer leerzeichenseparierten Liste von Literalen (von 0 verschiedene ganze Zahlen) gefolgt von einer abschließenden 0 besteht. Negative Zahlen stehen dabei für die negierten Variablen. Beispielsweise entspricht die Eingabe

```
p cnf 3 2
-1 2 0
1 -3 -2 0
```

der aussagenlogischen Formel $(\neg P_1 \vee P_2) \wedge (P_1 \vee \neg P_3 \vee \neg P_2)$.

Ausgabe von MiniSat

MiniSat liefert das Ergebnis in einer Datei zurück. Ist die Klauselmenge unerfüllbar, so lautet die erste und einzige Zeile UNSAT. Für den Fall der Erfüllbarkeit der Klauselmenge lautet diese Zeile SAT und die zweite Zeile enthält eine Beschreibung der erfüllenden Interpretation. Dabei werden diejenigen AL-Variablen, die als wahr interpretiert werden, durch eine positive ganze Zahl und diejenigen, die zu falsch ausgewertet werden durch eine negative dargestellt.

Die Aufgabe

Die Praxisaufgabe besteht darin, ein Java-Programm zu schreiben, das

1. Light-Up-Spielbretter in aussagenlogische Klauselmengen im DIMACS-Format transformiert, so dass jede erfüllende Belegung der Klauselmenge einer Lösung des Puzzles entspricht,
2. MiniSat aufruft, um eine solche erfüllende Belegung zu finden,

3. die gefundene Belegung in eine Lösung des Puzzles zurückübersetzt.

Die Bedingungen an die Lampenplatzierung müssen dazu aussagenlogisch formalisiert werden. Sie müssen entscheiden, was Sie durch AL-Variablen modellieren und die Forderungen aus der Problemstellung in Klauseln formulieren. Typischerweise wird eine Formalisierung mit $O(N^2)$ Variablen und $O(N^3)$ Klauseln auskommen.

Bitte beachten Sie: Die Klauselmenge wird von dem Java-Programm für jede Instanz (jedes Spielbrett) jeweils neu erstellt. Sie kodiert sowohl die allgemeinen Regeln des Spiels als auch die Lage der Wände auf dem gegebenen Spielbrett.

Kodierung der Spielbretter (Ein- und Ausgabe)

Spielbretter werden für diese Aufgabe durch Zeichenketten repräsentiert, die als Eingabe und Ausgabe für Ihr Java-Programm dienen. Die unten erwähnten Java-Klassen ermöglichen es Ihnen, diese Zeichenketten zu parsen und so die Spielbretter in Ihr Programm einzulesen. Auch können Sie damit die Lösung, das heißt ein Spielbrett mit platzierten Lampen, als Zeichenkette darstellen.

Sie finden auf der Webseite zur Vorlesung. Zu zweien davon sollen Sie die mit Ihrem Programm und MiniSat generierte Lösung abgeben.

Abgabe der Lösung

Die Abgabe der Lösung dieser Aufgabe erfolgt ebenfalls über die Webseite zur Vorlesung. Bitte registrieren Sie sich dort unter Ihrer Matrikelnummer und melden Sie sich am System an. Laden Sie für die beiden geforderten Aufgaben Lösung, das heißt die Zeichenketten für die Bretter mit platzierten Lampen, hoch.⁴

Bitte geben Sie zusätzlich den Quelltext (als ASCII-Text) Ihres Java-Programms auf der Seite ab. Auch für Programme, die nicht völlig korrekte Ergebnisse liefern, werden Punkte anteilig vergeben.

Infrastruktur

Wir haben für Sie etwas Infrastruktur vorbereitet, um Ihnen die Bearbeitung dieser Aufgabe zu erleichtern.

Java-Klassen

Auf den Webseiten zur Vorlesung finden Sie ein Paket von Java-Klassen, das es Ihnen erleichtern soll, mit Spielbrettern umzugehen. Sie können damit u.a.

- Zeichenketten in Spielbrettobjekte konvertieren und umgekehrt,
- den Status einzelner Felder abfragen und setzen,
- sich Spielbretter grafisch anzeigen lassen,
- Klauselmengen erstellen, die ausdrücken, dass genau n aus vier AL-Variablen wahr sind.

Auf den Seiten finden Sie außerdem etwas Dokumentation in JavaDoc. Bitte konsultieren Sie diese. Die Klassen können Ihnen die Arbeit stark erleichtern.

Sie sind nicht an die Programmiersprache Java gebunden. Da jedoch das Rahmenwerk in Java vorliegt, empfehlen wir stark, Java zu verwenden.

⁴Die Möglichkeit, sich zu registrieren und Lösungen hochzuladen, wird es auf der Webseite ab 20.11.2008 geben.

Installation von MiniSat

Wie oben bereits erwähnt, finden Sie das MiniSat-Werkzeug installiert im Studentenpool der ATIS. Die Rechner im Pool sind zu erreichen im Poolraum im Untergeschoss des Informatikgebäudes oder via SSH auf den Rechner `i08fs1.ira.uka.de`. Ein Konto für diese Rechner können alle Studenten der Informatik erhalten⁵.

Um das Werkzeug benutzen zu können, geben Sie bitte nach dem Einloggen auf der Konsole Folgendes ein:

```
source ~/s_ulbric/FORMSYS
```

Danach lässt sich MiniSat durch den Aufruf `minisat` starten. Zwei Kommandozeilenargumente werden dabei üblicherweise übergeben: Das erste ist die Datei mit der DIMACS-Eingabe und das zweite ist der Dateinamen, unter dem MiniSat das Ergebnis speichern soll.

```
minisat infile out
```

MiniSat gibt dann auf die Standardausgabe noch eine Menge Statusinformationen über die Anzahl der verwendeten Klauseln, benötigter Zeit, Speicher usw. aus

⁵Studenten anderer Fakultäten in Ausnahmefällen wie diesem auch