

Übungen zur Vorlesung Formale Spezifikation und Verifikation

Blatt 2

Aufgabe 2-1 Entscheiden Sie mit dem DPLL-Verfahren, ob folgende Formel erfüllbar ist. Berechnen Sie eine erfüllende Belegung, falls möglich.

$$D \wedge (\neg D \vee \neg A \vee B) \wedge (\neg B \vee \neg E) \wedge (E \vee A \vee \neg D) \wedge (\neg F \vee G) \\ \wedge (\neg G \vee (\neg D \vee \neg C)) \wedge (C \vee \neg H) \wedge (H \vee F) \wedge (D \vee A \vee B)$$

Aufgabe 2-2 In der Vorlesung wurde Sudoku in Aussagenlogik kodiert. Dazu wurden Variablen x_{ijk} für $i, j \in \{0, \dots, 8\}$ und $k \in \{1, \dots, 9\}$ gewählt. Es wurde eine Formel konstruiert, deren erfüllende Belegungen genau die Lösungen des Sudokus repräsentieren. Eine erfüllende Belegung macht die Variable x_{ijk} wahr genau dann wenn in der repräsentierten Lösung die Zahl k in Zelle (i, j) steht.

Andere Kodierungen des Problems sind ebenso möglich. Geben Sie eine Formel an, deren erfüllende Belegungen die Lösungen folgendermaßen kodieren: Die Variable x_{ijk} ist wahr genau dann wenn in Zelle (i, j) eine Zahl *kleiner-gleich* k steht.

Aufgabe 2-3 Das n -Damen-Problem besteht darin, n Damen auf einem Schachbrett der Größe $n \times n$ aufzustellen, so dass keine Dame eine andere schlagen kann. Zur Erinnerung: Damen können sich horizontal, vertikal und diagonal bewegen.

Die Zahl n sei beliebig gegeben. Geben Sie eine aussagenlogische Formel an, die erfüllbar ist genau dann wenn das n -Damen-Problem lösbar ist. Von einer erfüllenden Belegung sollte man leicht eine Lösung des Problems ablesen können. Erklären Sie, wie eine erfüllende Belegung eine Lösung des Problems kodiert.

Aufgabe 2-4 Schreiben Sie ein Programm, welches das n -Damen-Problem mit einem SAT-Solver löst. Eingabe soll die Zahl n sein. Wenn eine Lösung existiert, so soll eine Lösung ausgegeben werden. Für $n = 8$ könnte eine Lösung zu Beispiel so angezeigt werden:

```
. . . . . D . .
. . . D . . . .
. . . . . D .
D . . . . . .
. . . . . D
. D . . . . .
. . . . D . .
. . D . . . .
```

Auf der Vorlesungshomepage finden Sie Programme in Java und Haskell, welche die Tseitin-Transformation implementieren und die einen SAT-Solver direkt aufrufen können. Ihr Programm muss also nur für gegebenes n die Formel aus Aufgabe 2-3 ausrechnen. Mit der Funktion `satisfiable` können Sie dann Erfüllbarkeit testen.

Abgabe: Sie können ihre Lösungen bis Donnerstag, den 27.4., um 16:00 Uhr über UniWorX abgeben.