Matrikelnummer	Name

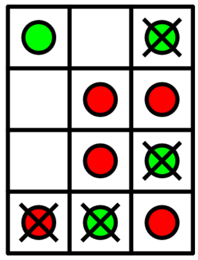
Aufgabenblatt 9 - Solitaire (cont'd)

Theoretische Informatik 1, SS15

Ausgabe: 29.05.2015 Abgabe: 02.07.2015

Die Sprache `SOLITAIRE` sei definiert wie in Aufgabenblatt 6:

$$\texttt{SOLITAIRE} := \{\langle G\rangle \mid G \text{ ist ein lösbares } \textit{Solitaire}\text{-Spiel}\}\ .$$



Ein *Solitaire*-Spiel

1. Zeigen Sie, dass `SOLITAIRE` NP-vollständig ist. **(5 Punkte)**

Hinweis: Reduzieren Sie hierzu `3SAT` auf `SOLITAIRE`. D.h. zeigen Sie, dass ein effizientes Verfahren R existiert, dass jede 3-KNF ϕ in ein *Solitaire*-Spiel G überführt, welches genau dann lösbar ist wenn für ϕ eine erfüllende Belegung existiert. Beschreiben Sie das Verfahren R, und erläutern Sie den Zusammenhang zwischen den Literalen, Variablen und Klausel in ϕ und den Zeilen, Spalten und Spielsteinen in G.

2. Implementieren Sie die Reduktion R in JFLAP. **(10* Punkte)**

Entwerfen Sie eine deterministische Turingmaschine $\mathcal{T}_R$, die eine 3-KNF ϕ als Eingabe nimmt und in ein *Solitaire*-Spiel G, wie in Punkt 1. umwandelt. Erweitern Sie ihre Implementierung des Enscheiders für `SOLITAIRE` aus dem Aufgabenblatt 6 mit der Maschine $\mathcal{T}_R$, sodass die Gesamtkonstruktion `3SAT` entscheidet.

Sie können davon ausgehen, dass die Eingabe die Codierung einer gültigen 3-KNF $\langle\phi\rangle$ mit k Variablen und l Klauseln ist. $\langle\phi\rangle$ hat folgendes Format:

$$\langle\phi\rangle \;=\; \texttt{\#}\langle L_{11}\rangle\texttt{\#}\langle L_{12}\rangle\texttt{\#}\langle L_{13}\rangle\texttt{\#\$\#}\langle L_{21}\rangle\texttt{\#}\langle L_{22}\rangle\texttt{\#}\langle L_{33}\rangle\texttt{\#\$\#}\ \ldots\ \texttt{\#\$\#}\langle L_{l1}\rangle\texttt{\#}\langle L_{l2}\rangle\texttt{\#}\langle L_{l3}\rangle\texttt{\#\$}$$

wobei L_{ij} das j-te Literal in der i-ten Klausel bezeichnet. Jedes $\langle L_{ij}\rangle$ hat die Form $bin(x)-f$, wobei $bin(x)$ ein eindeutiges binäres Muster bezeichnet, das eine Input-Variable von ϕ codiert. f ist `0` wenn die Variable negiert ist und `1` sonst. Sie können davon ausgehen, dass die binären Muster der Variablen aufsteigend sortiert sind. Weiters können Sie davon ausgehen, dass die binären Muster für alle Variablen die gleiche Länge haben.

Beispiel:

$$\phi(x_0, x_1, x_2, x_3) \;=\; (\neg x_0 \vee x_1 \vee \neg x_2) \wedge (x_0 \vee x_2 \vee \neg x_3)$$

$$\langle\phi\rangle \;=\; \texttt{\#00-0\#01-1\#10-0\#\$\#00-1\#10-1\#11-0\#\$}$$

Weiters soll folgende Konvention gelten: Wenn dem Input ein Fragezeichen vorgestellt ist, also die Eingabe `?`$\langle\phi\rangle$ anliegt soll die Maschine $\langle G\rangle$ als Output generieren und halten (ohne den Entscheider für `SOLITAIRE` auf $\langle G\rangle$ auszuführen).

Tipp: Sie können in einem bestehenden Turingmaschinenprogramm die Anzahl an Bändern ändern indem Sie die `jff`-Datei in einem Texteditor öffnen und die Zeile `<tapes>` ... `</tapes>` anpassen.