Philipps Universität Marburg

Übungen zu „Konzepte von Programmiersprachen“, WS 2010/11

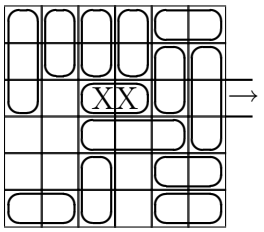
Prof. Dr. R. Loogen · Fachbereich Mathematik und Informatik · Hans-Meerwein-Straße, D-35032 Marburg

Nr. 8, Abgabe: Dienstag, 14. Dezember 2010 vor der Vorlesung

20. Rush Hour-Problem — 9 Punkte

Das Rush Hour-Problem[1] besteht darin, auf einem 6×6-Gitter vorgegebene PKWs und LKWs so zu bewegen, dass ein speziell markierter PKW zur „Ausfahrt“ gebracht wird. PKWs belegen dabei zwei Zellen, LKWs drei Zellen. Sie sind vertikal oder horizontal ausgerichtet und können dementsprechend nur in vertikaler oder horizontaler Richtung bewegt werden. Pro Zug kann ein Fahrzeug um eine Position verschoben werden.

Folgende Abbildung zeigt eine Beispiel-Startkonfiguration. Rechts daneben ist eine Nummerierung der Gitterplätze mit Zahlen zwischen 1 und 41 angegeben, die nicht durch 7 teilbar sind. Die Ausfahrt hat die Nummer 20.



1	2	3	4	5	6
8	9	10	11	12	13
15	16	17	18	19	**20**
22	23	24	25	26	27
29	30	31	32	33	34
36	37	38	39	40	41

Jedes Fahrzeug kann eindeutig durch das geordnete Paar der Randnummern der von ihm belegten Zellen dargestellt werden. Obige Beispielkonfiguration wird dann durch folgende Liste von Positionen beschrieben. Als erstes wird die Position des Fahrzeugs aufgeführt, das zur Ausfahrt bewegt werden soll:

[(17,18), (1,15), (2,9), (3,10), (4,11), (5,6), (12,19),
(13,27), (24,26), (31,38), (33,34), (36,37), (40,41)]

Entwickeln Sie ein Haskell-Programm, das eine Startkonfiguration aus einer als Aufrufparameter angegebenen Datei einliest, falls existent, eine Zugfolge, die die Start- in eine Zielkonfiguration überführt, in eine Ausgabedatei schreibt und auf der Standardausgabe die Länge der gefundenen Zugfolge ausgibt. / 2

Definieren Sie eine Funktion `findSolution :: Grid -> Maybe [Move]`, wobei die folgenden Typfestlegungen verwendet werden sollen: / 3

```
type Pos  = (Int, Int) -- Fahrzeug-Randpositionen
type Grid = [Pos]      -- Liste der Fahrzeug-Randpositionen
type Move = (Pos,Pos)  -- Positionswechsel eines Fahrzeugs
```

Hinweis: Folgende Funktionen sind zur Problemlösung hilfreich:

`solved :: Grid -> Bool` testet, ob eine Konfiguration Zielkonfiguration ist. / 0,5

`move :: Grid -> Move -> Grid` beschreibt den durch einen Zug bestimmten Konfigurationsübergang. / 0,5

`nextMoves :: Grid -> [Move]` bestimmt zu einer Konfiguration alle möglichen Züge. / 3

21. Typinferenz für Ausdrücke — 3 Punkte

Bestimmen Sie (falls möglich) den allgemeinsten Typ der folgenden Ausdrücke:

(a) `map foldr` (b) `foldr map` (c) `foldr foldr`

[1] siehe auch `http://www.puzzleworld.org/SlidingBlockPuzzles/rushhour.htm`