XI. Übungsblatt "Informatik I" WS 95/96

Christian Schulte, Prof. Gert Smolka

Ausgabe: 12. Januar 1995

Abgabe: 19. Januar 1996 vor der Vorlesung (Programme per E-Mail)

Aufgabe XI.1 (Isomorphie von Lokationen (6)) Sei Z ein Zustand gemäß der Semantik von Scheme 2. Zwei Lokationen l_1 und l_2 heißen *isomorph in* Z, falls eine der folgenden Bedingungen gilt:

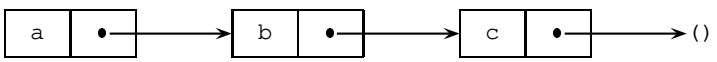
1. Z ist weder auf l_1 noch auf l_2 definiert.
2. $Z(l_1) = Z(l_2)$ und $Z(l_1)$ ist kein Paar.
3. $Z(l_1)$ und $Z(l_2)$ sind beides Paare, so daß in Z die erste Komponente von $Z(l_1)$ isomorph zu der ersten Komponente von $Z(l_2)$ ist und die zweite Komponente von $Z(l_1)$ isomorph zu der zweiten Komponente von $Z(l_2)$ ist.

1. Zeigen Sie, daß Isomorphie in Z eine Äquivalenzrelation ist.
2. Schreiben Sie eine Prozedur `(isomorph? x y)`, die testet ob `x` und `y` isomorph sind. Dabei können Sie annehmen, daß der von `x` erreichbare Graph nicht zirkulär ist.

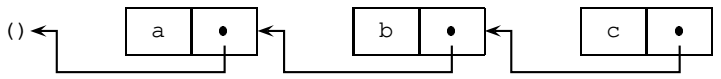
Aufgabe XI.2 (Umdrehen von Listen mit konstantem Speicher (6))

1. Schreiben Sie eine Prozedur `(reverse xs)`, die die Liste `xs` in umgekehrter Reihenfolge zurückliefert. Dabei dürfen weder cons-Zellen noch Prozeduren erzeugt werden. Die Liste `xs` darf jedoch verändert werden.

   Hinweis: Verwenden Sie dazu die folgende Idee. Die Liste `(a b c)` kann so skizziert werden:

   Durch die Verwendung der Prozedur `set-cdr!` kann man die "Verzeigerung" der Liste gerade invertieren:

2. Schreiben Sie eine Prozedur `(reverse xs)`, deren Anwendung die Liste `xs` umdreht.

Aufgabe XI.3 (Schaltkreissimulation (6)) In der Vorlesung haben Sie einen Schaltkreissimulator kennengelernt, dessen Programm mit Beispielen auf der WWW-Seite

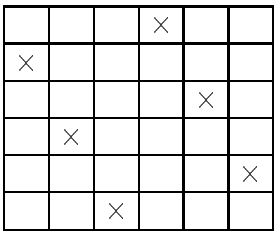
`http://ps-www.dfki.uni-sb.de/courses/info-i/programme.html`

zu finden ist. Schreiben Sie Programme für die folgenden Schaltungen und simulieren Sie diese:

1. Einen Taktgeber `(clock sk a)`, der alle zehn Zeiteinheiten seinen Ausgang `a` verändert. Dabei sollen die Verzögerungszeiten für alle Gatter zwei sein.
2. Einen Taktgeber `(clock-n sk a)`, der alle zehn Zeiteinheiten seinen Ausgang `a` verändert und nur Nichtgatter mit einer Verzögerungszeit von zwei verwendet.

3. Einen Multiplexer `(multiplexer sk e0 e1 c a)`, der in Abhängigkeit vom Wert von `c` entweder den Eingang `e0` oder `e1` auf den Ausgang `a` durchschaltet.

Aufgabe XI.4 (N-Damenproblem (8)) Das N-Damenproblem stammt von dem Mathematiker Carl Friedrich Gauss: Man muß N-Damen auf einem N×N-Schachbrett so plazieren, daß keine Dame eine andere bedroht. Dabei bedrohen sich zwei Damen, wenn sie in der selben Zeile, Spalte oder Diagonale plaziert sind. Eine Lösung für das 6-Damenproblem ist zum Beispiel:



Wir wollen dieses Problem in Scheme lösen. Dazu repräsentieren wir das N×N-Schachbrett durch eine Kette von N+1 Damenobjekten

$$\mathtt{d}_0 \quad \leftarrow \quad \mathtt{d}_1 \quad \leftarrow \quad \cdots \quad \leftarrow \quad \mathtt{d}_N$$

Dabei kennt jede Dame nur seine direkte Nachbarin.
Jede Dame $\mathtt{d}_k$ soll die folgenden Methoden haben:

- `((`$\mathtt{d}_k$ `'erste))` berechne die erste Teillösung (falls sie existiert).
- `((`$\mathtt{d}_k$ `'naechste))` berechne die nächste Teillösung (falls sie existiert).
- `((`$\mathtt{d}_k$ `'bedroht?) z s)` testet, ob das Plazieren einer Dame in der `z`-ten Zeile und `s`-ten Spalte eine der Damen $\mathtt{d}_k, \ldots, \mathtt{d}_1$ bedroht.
- `((`$\mathtt{d}_k$ `'belegung))` evaluiert zur aktuellen Belegung dargestellt als Liste von Paaren aus Zeile und Spalte der Damen $\mathtt{d}_k, \ldots, \mathtt{d}_1$.

Schreiben Sie eine Klasse

`(dame k n)`

die für das `n`-Damen Problem die Damen $\mathtt{d}_k, \ldots, \mathtt{d}_0$ erzeugt und $\mathtt{d}_k$ zurückgibt. Zunächst sollen alle Damen in der ersten Zeile stehen.
Eine Lösung für das 8-Damenproblem kann dann wie folgt berechnet werden:

```
(begin
  (define achte-dame (dame 8 8))
  ((achte-dame 'erste))
  ((achte-dame 'belegung)))
```