XIII. Übungsblatt "Informatik I" WS 95/96

Christian Schulte, Prof. Gert Smolka

Ausgabe: 26. Januar 1996

Abgabe: 2. Februar 1996 vor der Vorlesung (Programme per E-Mail)

Aufgabe XIII.1 (Queue (3)) Schreiben Sie die Klasse `queue` aus der Vorlesung (Sie finden diese auf der WWW Seite: `http://ps-www.dfki.uni-sb.de/courses/info-i/programme.html`) für den Scheme 2 Interpretierer um. Anstatt eine Fehlermeldung mittels `error` auszugeben, soll in einer Fehlersituation das Symbol `error` zurückgeliefert werden.

Aufgabe XIII.2 (Statistik (8)) Erweitern Sie den Scheme 2 Interpretierer so, daß gezählt wird:

1. Wieviel Prozeduren erzeugt werden.
2. Wieviel Paare erzeugt werden.
3. Wieviel Applikationen ausgeführt werden.

Diese Zahlen sollen durch eine Methode `statistik` in einer Form wie zum Beispiel:

```
*** Statistik Scheme 2 Interpretierer
*** Anzahl erzeugte Prozeduren: 4
*** Anzahl erzeugte Paare:      5
*** Anzahl Applikationen:       42
```

ausgegeben werden. Gleichzeitig sollen alle Zähler dabei auf null zurückgesetzt werden.

Aufgabe XIII.3 (Sequentialisierung (4)) Erweitern Sie den Scheme 2 Interpretierer so, daß er anstatt eines zweistelligen Sequentialisierungskonstruktes (`seq` A_1 A_2) ein Sequentialisierungskonstrukt beliebiger Stelligkeit (`seq` $A_1 \ldots A_n$) auswerten kann.

Aufgabe XIII.4 (Syntax- und Laufzeitfehler (4)) Der Interpretierer ist unter Zuhilfenahme des Konstruktes `abort`, das in Scheme 2 nicht definiert ist, implementiert. Dieses Konstrukt läßt sich eliminieren, wenn man einen speziellen Wert `abort` einführt: Evaluiert ein Teilausdruck zu `abort`, so evaluiert der gesamte Ausdruck zu `abort`.

1. Führen Sie im Interpretierer den speziellen Wert `abort` ein. Gehen Sie dabei wie für den Wert `undef` vor.
2. Schreiben Sie `werte-aus-konditional` und `werte-aus-sequentialisierung` um, so daß der Wert `abort` berücksichtigt wird.

Aufgabe XIII.5 (Fehlermeldungen (4)) Neben `abort` macht der Scheme 2 Interpretierer auch von `display` und `newline` Gebrauch. Redefinieren Sie `display` und `newline` so, daß der Interpretierer im Fehlerfall eine Liste (`"error"` $\cdots$ `"newline"` $\cdots$) zurückliefert.

Zusatzaufgabe XIII.6 (Paare als Paare (8)) Der Scheme 2 Interpretierer implementiert Paare in Scheme 2 als dreielementige Listen, also mit drei Paaren. Wie muß der Interpretierer geändert werden, so daß Paare durch jeweils nur ein Paar realisiert werden? Sorgen Sie dafür, daß die primitive Prozedur `pair?` in Scheme 2 wirklich nur auf **echte** Paare testet.

Den Scheme 2 Interpretierer finden Sie auf der WWW-Seite:

`http://ps-www.dfki.uni-sb.de/courses/info-i/interpretierer.scm`