Institut für Informatik
und Praktische Mathematik
der Universität Kiel

Kiel, 8.05.2002

Übungen zu Organisation und Architektur von Rechnern
SS 2002
Serie 3

Aufgabe 9 (10 Punkte)

Implementieren Sie ein Trace-Tool für die SEMCD-Maschine aus Serie 2. Definieren Sie dazu eine Funktion `trace-semcd`, die einen Ausdruck schrittweise transformiert und bestimmte Zustände der SEMCD-Maschine mit Hilfe der Funktion `print-expr` aus Aufgabe 4 in eine Liste von Strings konvertiert.

Um die auszugebenden Zustände bezeichnen zu können, werden diese mit 0 beginnend (der zu berechnende Ausdruck liegt auf dem C-Stack) durchnumeriert. Die Funktion `trace-semcd` erhält dann neben dem zu evaluierenden Ausdruck eine Liste von Zustandsnummern als zweites Argument.

Sorgen Sie dafür, daß `trace-semcd` für Zustandsbezeichnungen, die größer als die Anzahl der möglichen Transformationsschritte des gegebenen Ausdruckes sind, den letzten gültigen Zustand ausgibt.

Hinweis: Es kann davon ausgegangen werden, daß die Liste von Zustandsbezeichnungen streng monoton steigend sortiert ist.

Schreiben Sie weiterhin eine Funktion `from-to`, die zwei Parameter `m` und `n` hat und eine aufsteigende Liste der Zahlen von `m` bis `n` erzeugt.

Aufgabe 10 (20 Punkte)

Erweitern Sie Ihre SEMCD-Maschine um die primitiven Operationen `+`, `-`, `*`, `/`, `<=`, `=`, die Bool'schen Konstanten `#t` und `#f`, sowie um Konditionale wie in der Vorlesung vorgestellt.

Definieren Sie dazu zunächst die notwendigen abstrakten Datentypen:

`(define-structure (Prf name))`	repräsentiert	eine Primitive Funktion *name*
`(define-structure (Cond e1 e2))`	repräsentiert	ein Conditional: $\Delta^{(2)}\, e1\;\; e2$
`(define-structure (Bool val))`	repräsentiert	eine Bool'sche Konstante *val*

Repräsentieren Sie die Namen der primitiven Funktionen durch die entsprechenden Strings. Ergänzen Sie die Ausgabefunktion `print-expr` entsprechend und testen Sie Ihre Implementierung ausgiebig!!

Untersuchen Sie unter anderem folgenden Ausdruck:

$$
\begin{array}{l}
\overline{@}^{(2)}\ \texttt{3} \\
\qquad \overline{@}^{(2)}\ \lambda^{(2)}\ \texttt{fac}\ \lambda^{(2)}\ \texttt{n}\ \overline{@}^{(2)}\ \overline{@}^{(3)}\ \texttt{1 n <=} \\
\hspace{10cm} \Delta^{(2)}\texttt{1} \\
\hspace{11cm} \overline{@}^{(3)}\ \texttt{n}\ \overline{@}^{(2)}\ \overline{@}^{(3)}\ \texttt{1 n - fac *} \\
\qquad\qquad \texttt{Y}
\end{array}
$$

wobei `Y` folgendermassen zu definieren ist:

$$
\begin{array}{l}
\lambda^{(2)}\ \texttt{y}\ \overline{@}^{(2)}\ \lambda^{(2)}\ \texttt{x}\ \overline{@}^{(2)}\ \lambda^{(2)}\ \texttt{z}\ \overline{@}^{(2)}\ \texttt{z}\ \overline{@}^{(2)}\ \texttt{x x y} \\
\hspace{2.5cm} \lambda^{(2)}\ \texttt{x}\ \overline{@}^{(2)}\ \lambda^{(2)}\ \texttt{z}\ \overline{@}^{(2)}\ \texttt{z}\ \overline{@}^{(2)}\ \texttt{x x y}
\end{array}
$$

Aufgabe 11 (10 Punkte)

Um rekursive Ausdrücke leichter handhaben zu können, soll ein primitiver Rekursionsoperator $\alpha^{(2)}\texttt{f e}$ eingeführt werden. Er berechnet sich zu einem Ausdruck `e'` der dadurch aus `e` entsteht, daß alle freien Vorkommen der Variablen `f` in `e` durch $\alpha^{(2)}\texttt{f e}$ ersetzt werden.

Beispiel:

$$
\begin{array}{c}
\alpha^{(2)}\texttt{f}\ \lambda^{(2)}\ \texttt{n}\ \overline{@}^{(2)}\ \texttt{n f} \\
\Downarrow \\
\lambda^{(2)}\ \texttt{n}\ \overline{@}^{(2)}\ \texttt{n}\ \alpha^{(2)}\texttt{f}\ \lambda^{(2)}\ \texttt{n}\ \overline{@}^{(2)}\ \texttt{n f}
\end{array}
$$

Überlegen Sie sich, um welche Regeln man die SEMCD-Maschine erweitern muß, um die Auswertung des Rekursionsoperators zu integrieren. Achten Sie dabei darauf, daß der Rumpf des Rekursionsoperators relativ freie Variablen enthalten kann!!

Implementieren Sie die von Ihnen entworfenen Regeln basierend auf folgender Darstellung für den Rekursionsoperator:

`(define-structure (Rec fun expr))` repräsentiert $\alpha^{(2)}\textit{fun expr}$

Testen Sie Ihre Implementierung unter anderem mit folgenden Beispielen:

- $$
\begin{array}{l}
\overline{@}^{(2)}\ \texttt{3} \\
\qquad \alpha^{(2)}\texttt{fac}\ \lambda^{(2)}\ \texttt{n}\ \overline{@}^{(2)}\ \overline{@}^{(3)}\ \texttt{1 n <=} \\
\hspace{6.5cm} \Delta^{(2)}\texttt{1} \\
\hspace{7.5cm} \overline{@}^{(3)}\ \texttt{n}\ \overline{@}^{(2)}\ \overline{@}^{(3)}\ \texttt{1 n - fac *}
\end{array}
$$

- $$
\begin{array}{l}
\overline{@}^{(2)}\ \texttt{\#f} \\
\qquad \overline{@}^{(2)}\ \texttt{1}\ \lambda^{(2)}\ \texttt{m}\ \alpha^{(2)}\texttt{f}\ \Delta^{(2)}\texttt{m} \\
\hspace{7.5cm} \overline{@}^{(2)}\ \texttt{2}\ \lambda^{(2)}\ \texttt{m}\ \overline{@}^{(2)}\ \texttt{\#t f}
\end{array}
$$

Abgabe: Di., 28.05.2002 in den Übungen.