Andreas Podelski
Fachbereich 14 – Informatik
Universität des Saarlandes,
Max-Planck-Institut für Informatik

Programmiersprachen (WS 1996/97)

28. November 1996

Übungsblatt 6

Abdelwaheb Ayari
Hubert Baumeister

Aufgabe 1: (4 Punkte)

1. Gegeben sei die Funktion `secl`

   ```
   val secl = fn x => fn f => fn y => f(x,y)
   ```

   Was ist der Sinn dieser Funktion?
   Zeige mit Hilfe des Typsystems, daß gilt

   $\vdash$ `secl` : $\forall$`'a.` $\forall$`'b.` $\forall$`'c. 'a -> ('a * 'b -> 'c) -> 'b -> 'c`

2. Bestimme den jeweils allgemeinsten Typ folgender Ausdrücke mit dem in der Vorlesung vorgestellten Typinferenzalgorithmus:

   (a) **let** $\mathtt{o} = \lambda f.\lambda g.\lambda x.f(g\ x)$ **in** $\mathtt{o}\ \mathtt{o}$

   (b) $(\lambda\mathtt{o}.\mathtt{o}\ \mathtt{o})\ (\lambda f.\lambda g.\lambda x.f(g\ x))$

   (c) **let** $\mathtt{eins} = \lambda f.\lambda u.f\ u$ **in let** $\mathtt{zwei} = \lambda f.\lambda u.f\ (f\ u)$ **in** $\mathtt{eins}\ \mathtt{zwei}$

Aufgabe 2: (3 Punkte)
Implementiere mit Hilfe von Referenzen einen Ringbuffer, dargestellt als doppelt verkettete zirkuläre Liste. Gegeben seien die Datentypen

```
datatype 'a buf = Empty | Node of 'a buf ref * 'a * 'a buf ref;
datatype 'a T = Ptr of 'a buf ref;
```

Implementiere die Funktionen
`left: 'a buf -> 'a buf ref`, `right: 'a buf -> 'a buf ref` um das linke bzw. rechte Listenelement zu erreichen,
`empty : unit -> '1a T`, die einen leeren Buffer erzeugt,
`null : 'a T -> bool`, die testet ob ein Buffer leer ist,
`label : 'a T -> 'a`, um das Element im aktuellen Listenelement zu bekommen,
`moveleft : 'a T -> unit`, `moveright : 'a T -> unit` welches den linken bzw. rechten Nachbar zum aktuellen Listenelement macht,
`insert : '1a T * '1a -> unit` um ein Element in den Buffer einfügen, und
`delete : 'a T -> 'a` um das aktuelle Element aus dem Buffer zu löschen.

Anwendung: `n` Todeskandidaten sollen hingerichtet werden und müssen sich in einem Kreis aufstellen. Es wird jeweils der `k`.te Kandidat hingerichtet und aus dem Kreis entfernt. Ausgehend von dem nächsten Kandidaten wird wieder der `k`.te Kandidat hingerichtet und so weiter. Der letzte Kandidat wird begnadigt. Wo sollte man am besten stehen bei `n = 12` und `k = 7` ? Implementiere mit Hilfe des Ringbuffers eine Funktion `where : int * int -> int` die bei Eingabe von `(n,k)` den besten Platz ermittelt.

Aufgabe 3: (3 Punkte)
Implementiere eine Funktion `wc: string -> unit` welche in einem Textfile, dessen Name sie als Parameter erhält, die Anzahl an Zeichen, Wörtern und Zeilen bestimmt. Die Ausgabe soll dann ähnlich dem Unixtool `wc` erfolgen, zum Beispiel:

```
- wc "/usr/man/cat1/csh.1";
1914   9340   76816   /usr/man/cat1/csh.1
val it = () : unit
cip48 [19:19] 8 ~ -> wc /usr/man/cat1/csh.1
    1914    9340   76816 /usr/man/cat1/csh.1
```

Man braucht dazu die Funktionen:

`open_in : string -> instream` öffnet ein File und liefert einen Eingabestrom zurück.

`close_in : instream` schließt einen Eingabestrom.

`input instream * int -> string` liest aus einem Eingabestrom $n (> 0)$, das Argument, Zeichen. Bei Fileende wird `""` gelesen.

`print : int|string -> unit` gibt sein Argument auf der Standardausgabe aus.

Als Wortbegrenzer sind `" "`, `"\t"` und `"\n"` zu erkennen, als Zeilenbegrenzer `"\n"`.