

Übungen zu „Prinzipien von Programmiersprachen“

Blatt 6

Aufgabe 20. Formalisiere die folgenden Schleifenkonstrukte. Lege die statische Semantik jeweils mit Hilfe von Typregeln und die dynamische Semantik mit Hilfe von Auswertungsregeln fest.

1. Endlosschleife der Form **forever** e **end**;
2. **repeat**-Schleife der Form **repeat** e_1 **until** e_2 ;
3. **for**-Schleife der Form **for** $x = e_1$ **to** e_2 **do** e_3 **end**: überlege, ob die Laufvariable x an dieser Stelle gebunden wird, ob x in e_3 modifiziert werden darf und wie oft e_1 und e_2 ausgewertet werden.
4. **break**: der Ausdruck terminiert die nächste umgebende Schleife. Stelle mit Hilfe der statischen Semantik sicher, dass **break** tatsächlich nur innerhalb einer Schleife verwendet wird. Mit Hilfe von **forever** und **break** lassen sich Schleifen mit „mittigem Ausstieg“ realisieren.

```
forever ...
    if e then break
    else ...
end
```

Aufgabe 21. Formalisiere die folgenden Erweiterungen von **try** und **throw**:

1. Um Informationen über die Fehlerursache an den Handler kommunizieren zu können, wird **throw** um ein Argument vom Typ *Exception* erweitert:

```
| C e -- Konstruktion einer Ausnahme
| throw e -- Werfen einer Ausnahme
| try e catch C1 x1 ⇒ e1 | C2 x2 ⇒ e2 end -- Ausnahmebehandlung
```

Das **catch**-Konstrukt führt (ähnlich wie **case**) wie eine Fallunterscheidung auf dem Datentyp *Exception* durch; gibt es keine passende Klausel, wird die Ausnahme „weitergeworfen“.

2. Mit Hilfe des **finally**-Konstrukts kann ein Ausdruck angegeben werden, der unabhängig vom Ausgang des **try**-Ausdrucks auf jeden Fall ausgeführt wird.

```
| try e catch C1 x1 ⇒ e1 | C2 x2 ⇒ e2 finally e' end
```

Überlege insbesondere, was passiert, wenn e_1 oder e_2 ihrerseits Ausnahmen werfen. Hat der **try**-Ausdruck mit der **finally**-Klausel die gleiche Semantik wie das folgende Codefragment, in dem e' in alle Zweige hineingezogen wurde?

```
try let x = e in e'; x
catch C1 x1 ⇒ let x1 = e1 in e'; x1
| C2 x2 ⇒ let x2 = e2 in e'; x2
end
```