

Übungen zur Vorlesung
Semantik von Programmiersprachen
Blatt 3

Aufgabe 1. In der Vorlesung wurden für die Small-Step-Semantik Regeln für drei mögliche Auswertungsstrategien für Boolesche Ausdrücke angegeben: vollständig, linkssequentiell und parallel.

Formulieren Sie Regeln in der Big-Step-Semantik für die linkssequentielle sowie die parallele Auswertung von Booleschen Ausdrücken.

Aufgabe 2. Betrachten Sie eine Erweiterung von IMP um Exceptions, in der es zwei neue Anweisungen gibt:

$$\text{Com} \ni c ::= \dots \mid \mathbf{throw} \ E \mid \mathbf{try} \ c \ \mathbf{with} \ E \Rightarrow c$$

Der Einfachheit halber soll es nur eine einzige Exception namens E geben.

Ein mögliches Beispielprogramm wäre:

```

try (while true do
  if  $x > 0$  then skip else (throw  $E$ );
   $z := z + y$ ;
   $x := x - 1$ )
with  $E \Rightarrow \mathbf{skip}$ 

```

Sein Endergebnis entspricht dem von **while** $x > 0$ **do** ($z := z + y$; $x := x - 1$).

Geben Sie eine operationelle Semantik (small-step oder big-step) für eine sinnvolle Interpretation von IMP mit Exceptions an.

Können Sie ihre Semantik so verändern, dass auch Exceptions mit einer ganzen Zahl als Parameter behandelt werden können?

$$\text{Com} \ni c ::= \dots \mid \mathbf{throw} \ E(a) \mid \mathbf{try} \ c \ \mathbf{with} \ E(x) \Rightarrow c$$

Aufgabe 3. Implementieren Sie einen Interpreter für IMP, indem Sie Small-Step- oder Big-Step-Semantik direkt implementieren.

Auf der Vorlesungshomepage gibt es dazu Programmfragmente in Haskell und SML.

Abgabe: Vor der Vorlesung am 12.5.