Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Clemens Ballarin

Vorlesung Semantik
Sommersemester 2007
Übungsblatt 6
Besprechung am 12. Juni 2007

Aufgabe 1 (H) (*Methodenaufruf*)

Gegeben Sei das Programm P aus Übungsblatt 5. Bestimmen Sie für den Ausdruck

$$e = \texttt{x := new Z; Var x. flip()}$$

die Auswertung ausgehend vom leeren Heap und der leeren Variablenumgebung:

$$P \vdash \langle e, ([], [])\rangle \Rightarrow \langle e', s'\rangle$$

Geben Sie die vollständige Herleitung in der Big-Step-Semantik an.

Aufgabe 2 (Ü) (*Finale Ausdrücke und Exceptions*)

In der Vorlesung wurden für die Big-Step-Semantik von Jinja mit Exceptions finale Ausdrücke folgendermaßen definiert:

$$\mathit{final}\ e \equiv (\exists v.\, e = \mathit{Val}\ v) \vee (\exists a.\, e = \mathit{Throw}\ a)$$

Wie muss die Definition für *finals* es aussehen, so dass das folgende Lemma gilt:

> Wenn $P \vdash \langle e, s\rangle \Rightarrow \langle e', s'\rangle$, dann *final* e' und
> wenn $P \vdash \langle es, s\rangle [\Rightarrow] \langle es', s'\rangle$, dann *finals* es'.

Geben Sie einen Beweis für das Lemma mit Regelinduktion über die Relationen $\Rightarrow$ und $[\Rightarrow]$ an.

Hinweis: zur Bestimmung von *finals* es genügt es, die Regeln auf der Folie "Exceptions: Method call" zu betrachten.

Aufgabe 3 (Ü) (*Unerreichbare Anweisungen*)

In Java werden Ausdrücke vom Compiler zurückgewiesen, wenn Sie *unerreichbare* Teilausdrücke enthalten. Dabei heisst ein Teilausdruck unerreichbar, wenn ein vorheriger Teilausdruck mit Sicherheit zu einer Exception ausgewertet wird. Beispielsweise ist in

$$\texttt{throw}\, e_1; e_2$$

der Teilausdruck e_2 unerreichbar.

a) Erweitern Sie das Typsystem von Jinja um einen Typ *Exn*, so dass aus $P, E \vdash e :: \mathit{Exn}$ folgt, dass e mit Sicherheit (also für jeden beliebigen Startzustand) zu einer Exception auswertet.

b) Geben Sie eine Funktion $\mathcal{U}$ an, die zu einem Programm P und Ausdruck e die Menge der unerreichbaren Teilausdrücke zurückgibt.

Aufgabe 4 (P) (*Deklarationsinformation in Prolog*)

In dieser und den Programmieraufgaben der folgenden Übungsblätter wird die Semantik von Jinja in Prolog implementiert. Als ersten Schritt hierzu implementieren Sie die Prädikate, mit denen Informationen aus einem Jinja-Programm extrahiert wird, insbesondere $P \vdash C \preceq^* D$, $P \vdash C$ has-fields *FDTs*, $P \vdash C$ sees $F : T$ in D, $P \vdash C$ has $F : T$ in D und $P \vdash C$ sees $M : Ts \to T = B$ in D.

Das Prädikat has wurde in der Vorlesung nicht eingeführt. $P \vdash C$ has $F : T$ in D bedeutet, dass im Programm P eine (nicht notwendig echte) Oberklasse D von C ein Feld F vom Typ T hat, unabhängig davon, ob das Feld von C aus sichtbar ist, oder nicht.

Einen Programmrahmen finden Sie im Verzeichnis

```
/usr/proj/semantik/prog/prolog/jinja-declare
```

in der Datei `declare.pl`. Geben Sie Ihre modifizierte Datei ab.

Hinweise: Benutzen Sie für Maps Assoziationslisten und verwenden Sie das Prädikat `lookup`, das in `map.pl` vorgegeben ist, um einen Eintrag aus einer Map auszulesen. Das in `init.pl` deklarierte Prädikat `prog1` ermöglicht es Ihnen, Ihre Implementierung an dem Beispielprogramm aus der Vorlesung zu testen.