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

Vorlesung Semantik
Sommersemester 2004
Übungsblatt 6
Besprechung am 2. Juni 2004

Aufgabe 1 (H) (*Deklarationsinformation*)

Gegeben ist folgendes Jinja-Programm `P` mit den Klassen `A`, `B`, `C` und `D`:

```
class A extends Object { field n: Intg; method m: Intg -> Intg = bodyA }
class B extends A { field n: Bool }
class C extends B { method m: () -> Bool = bodyC }
class D extends A { method m: () -> Intg = bodyD }
```

a) Geben Sie für alle Klassen C ein $FDTs$ an, so dass $\texttt{P} \vdash C$ has-fields $FDTs$ gilt.

b) Bestimmen Sie alle Tupel (C, F, T, D), für die $\texttt{P} \vdash C$ sees $F : T$ in D gilt.

c) Bestimmen Sie alle Tupel (C, F, T, D), für die $\texttt{P} \vdash C$ has $F : T$ in D gilt.

d) Bestimmen Sie alle Tupel (C, M, Ts, T, B, D), für die $\texttt{P} \vdash C$ sees $M : Ts \to T = B$ in D gilt.

Aufgabe 2 (Ü) (*Binäre Operatoren in Jinja*)

Zur Auswertung von binären Ausdrücken gibt es in Jinja die Operation *binop*. Erweitern Sie dessen Definition:

a) Um die Operation `<=` für Integer. Dabei steht $\leq$ auf dem Typ der ganzen Zahlen zur Verfügung.

b) Um die Operation `div` mit Hilfe der Division /. Definieren Sie die Semantik von `div` so, dass Division durch Null undefiniert ist (es wird also keine Ausnahme erzeugt).

Aufgabe 3 (Ü) (*Natürliche Zahlen in Jinja*)

Geben Sie eine Deklaration der Klasse `Nat` an, die die natürlichen Zahlen implementiert, ohne dabei die in Jinja vorhandenen Integer-Zahlen zu benutzen. Verwenden Sie stattdessen Zeiger. Der Nullzeiger *Null* repräsentiert die Null, ein Zeiger auf die Zahl n repräsentiert $n + 1$. Zum Beispiel ist

$$\bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow (\mathit{Null})$$

die Darstellung von drei. Geben Sie Methoden für die Nachfolgeroperation `suc`, Addition `add` und den Vergleich zweier Zahlen `eq` an.

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 Operationen, 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.

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.