

### Aufgabe 1 (H) (*Das Typsystem von Jinja*)

- Geben Sie ein Programm  $P$ , eine Typumgebung  $E$  und einen Ausdruck  $e$  ohne Variable an, so dass  $e$  gemäß der Typisierungsregeln von Jinja kein Typ zugeordnet wird.
- Zeigen Sie, dass der Typ, der einem Ausdruck  $e$  für ein Programm  $P$  und eine Typumgebung  $E$  zugeordnet wird, eindeutig ist.  
Hinweis: Führen Sie den Beweis mit struktureller Induktion über den Ausdruck. Zur Vereinfachung betrachten Sie nicht alle Ausdrücke, sondern nur Val, binäre Operation, Sequenz, If und Methodenaufruf.

### Aufgabe 2 (Ü) (*Subsumption*)

In den Typisierungsregeln von Jinja kommt an mehreren Stellen die Teiltypen-Relation  $\leq$  vor. Die Regeln sollen nun vereinfacht werden. Hierzu wird die Subsumptions-Regel

$$[e : T; T \leq T'] \implies e : T'$$

zu den Typisierungsregeln hinzugenommen. Diese Regel besagt, dass wann immer einem Ausdruck der Typ  $T$  zugeordnet wird, ihm auch jeder Obertyp  $T'$  von  $T$  zugeordnet wird.

- Vereinfachen Sie die Typisierungsregeln für Variablenzuweisung, Feldzuweisung, If, Binäre Operation und Typcast.
- Diskutieren Sie, welche Änderungen sich hierdurch für das Typsystem insgesamt und für bestimmte Ausdrücke ergeben.

### 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

`throw e1; e2`

der Teilausdruck  $e_2$  unerreichbar.

- Erweitern Sie das Typsystem von Jinja um einen Typ  $Exn$ , so dass aus  $P, E \vdash e :: Exn$  folgt, dass  $e$  mit Sicherheit (als für jeden beliebigen Startzustand) zu einer Exception ausgewertet.
- Geben Sie eine Funktion  $\mathcal{U}$  an, die zu einem Programm  $P$  und Ausdruck  $e$  die Menge der unerreichbaren Teilausdrücke zurückgibt.

(Bitte wenden!)

#### Aufgabe 4 (P) (*Big-Step-Semantik von Jinja*)

Implementieren Sie die Regeln der Big-Step-Semantik (ohne Exceptions) von Jinja in Prolog. Sie finden einen Rahmen zu dieser Aufgabe im Verzeichnis

```
/usr/proj/semantik/prog/prolog/jinja-big/.
```

Ergänzen Sie die Regelrümpfe in der Datei `semantik.pl` und geben Sie Ihre Datei ab. Einige Testbeispiele finden Sie wieder in `dialog.txt`.

Hinweis: Die Datei `declare.pl` wird nach der Abgabefrist der vorherigen Programmieraufgabe freigegeben.