

Aufgabe 1 (H) (*Definite Assignment*)

Sei

$$P \vdash \langle e, (h, l) \rangle \rightarrow \langle e', (h', l') \rangle$$

eine Reduktion in der Small-Step-Semantik von Jinja ohne Exceptions. Zeigen Sie

- a) $\text{dom } l \subseteq \text{dom } l'$
- b) $\text{dom } l \cup \mathcal{A}e \subseteq \text{dom } l' \cup \mathcal{A}e'$

Hinweis: Regelinduktion über \rightarrow .

Aufgabe 2 (Ü) (*Arrays*)

In dieser und einer Aufgabe auf dem nächsten Übungsblatt wird Jinja um Arrays erweitert. Dabei sind sowohl Arrays von primitiven Typen und Objekten, als auch Arrays von Arrays zugelassen, so dass mehrdimensionale Arrays realisiert werden können. In dieser Aufgabe wird die Semantik erweitert, in der folgenden Aufgabe das Typsystem.

- a) Erweitern Sie zunächst das formale Modell. Hierzu müssen die zulässigen Werte (*val*) und Typen (*ty*) erweitert werden; ausserdem der Heap, so dass er neben Objekten auch Arrays aufnehmen kann.
- b) Geben Sie jetzt Regeln in der Big-Step-Semantik für die Arrayoperationen

<code>new T[e]</code>	Array mit Elementtyp <i>T</i> der Größe <i>e</i> erzeugen
<code>e₁[e₂]</code>	Zugriff im Array <i>e₁</i> auf das Element mit Index <i>e₂</i>
<code>e₁[e₂] := e₃</code>	Zuweisung von <i>e₃</i> im Array <i>e₁</i> auf das Element mit Index <i>e₂</i> .

an. Gehen Sie davon aus, dass für Ausnahmesituationen, die bei diesen Operationen auftreten, entsprechende Systemexceptions vorhanden sind.

Aufgabe 3 (P) (*Das Typsystem von Jinja*)

Implementieren Sie das Typsystem von Jinja. Hierzu finden Sie einen Rahmen im Verzeichnis

```
/usr/proj/semantik/prolog/jinja-type/.
```

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