

System Architecture - SS15
Exercise Sheet 9(due: June 22, 2015)

Wichtig:

- Sie Benötigen 50% aller Übungsblätter die für Klausur X relevant sind, um zu Klausur X zugelassen zu werden. Dieses Blatt ist Relevant für Haupt- und Nachklausur.
- Das Übungsblatt muss stets am Montag nach der Vorlesung bei mir in der Office Hour oder, falls zeitgleich, in der Übungsgruppe Ihrer Tutorin abgegeben werden.
- Geben Sie stets Ihren Namen, Ihre Matr. Nr., und den Namen ihrer Tutorin auf der vordersten Seite oben rechts an.
- Sie dürfen Ergebnisse von vorherigen Aufgaben verwenden, auch wenn Sie diese nicht gelöst haben. Markieren sie Gleichungen, in denen Sie ein vorheriges Ergebniss benutzen, mit dem Kürzel E+Aufgabenblatt+Aufgabennummer.
- Wenn Sie sich nicht für die Klausur vorbereiten möchten, aber trotzdem zugelassen werden möchten, schreiben Sie einfach Ihren Namen und Ihre Matrikelnummer auf die Lösung einer kompetenten Mitstudentin. Es besteht auch keine Anwesenheitspflicht in den Übungsgruppen.

Tutor: _____

Namen, Matr. Nummern: _____

Aufgabe 1:

(10)

In folgender Konfiguration c im Programm von Aufgabe 3, Blatt 8 sei

$$top(c) = (f, -) :$$

System Architecture - SS15
 Exercise Sheet 9(due: June 22, 2015)

```

t1(label) = 'Prog'
t1(father) = null
t1(fson) = 1
t1(bro) = null
t2(label) = 'FuDS'
t2(father) = 0
t2(fson) = ...
t2(bro) = null

c.m(x) = {
  null  x = top(c).result
  1     x = top(c).j
  0     x = top(c).p
  0     x = gm.programtree
  t1    x = 0
  t2    x = 1
  ...   ...

c.st(c.rd) = nson
c.ht(0) = DTE
c.ht(1) = DTE
c.nh = 2

```

Bestimmen Sie wenn möglich $etype(e, f)$, $lv(e, c)$, $va(e, c)$ von folgenden Ausdrücken e . Wenn es nicht möglich ist, $etype(e, f)$, $lv(e, c)$, oder $va(e, c)$ zu bestimmen, erklären Sie warum.

- (a) (1 point) $e = \text{true}$
- (b) (1 point) $e = \text{p*}$
- (c) (1 point) $e = \text{p*.fson*.label}$
- (d) (1 point) $e = \text{j} \neq 0$
- (e) (1 point) $e = \text{result\&}$
- (f) (1 point) $e = \text{programtree}$
- (g) (1 point) $e = \text{p*.bro\&}$
- (h) (1 point) $e = \text{programtree} == \text{p}$
- (i) (1 point) $e = \text{programtree*} == \text{p*}$
- (j) (1 point) $e = \text{programtree\&} == \text{p\&}$
- (k) (1 point (bonus)) $e = \text{p*.bro*}$
- (l) (1 point (bonus)) $e = \text{j} \neq \text{false}$
- (m) (1 point (bonus)) $e = \text{seq}$
- (n) (1 point (bonus)) $e = \text{null}$

System Architecture - SS15
Exercise Sheet 9(due: June 22, 2015)

Solution: Wir machen nur einige Teilaufgaben.

$e = \text{true}$: Konstante, $etype(e, f) = \text{bool}$, $va(e, c) = 1$, $lv(e, c) = \perp$.

$e = \text{p*.fson*.label}$: Subvariable,

$$etype(e, f) = t_i$$

wenn

$$etype(\text{p*.fson*}, f) = \{t_1 n_1; \dots; t_s n_s\},$$

und

$$n_i = \text{label}.$$

Dabei ist

$$etype(\text{p*.fson*}, f) = t$$

wenn

$$etype(\text{p*.fson}, f) = t^*,$$

und

$$etype(\text{p*.fson}, f) = t'_j$$

wenn

$$etype(\text{p*}, f) = \{t'_1 n'_1; \dots; t'_{s'} n'_{s'}\},$$

und

$$n'_j = \text{fson}.$$

Und

$$etype(\text{p*}, f) = t'$$

wenn

$$etype(\text{p}, f) = t' * .$$

Jetzt sind wir bei einer Variable angelangt und wissen:

$$\text{p} = n''_1$$

mit

$$\$cf(c) = \{t''_1 n''_1; \dots; t''_3 n''_3\},$$

da

$$\$cf(c) = \{ \text{DTEp p}; \text{uint j}; \text{DTEp result} \} .$$

Jetzt arbeiten wir uns schrittweise wieder hoch.

Also ist

$$etype(\text{p}, f) = t''_1 = \text{DTEp} = \text{DTE} * .$$

Daraus folgt direkt:

$$t' = \text{DTE} = \{ \text{uint label}; \text{DTEp father}; \text{DTEp fson}; \text{DTEp bro} \},$$

System Architecture - SS15
Exercise Sheet 9(due: June 22, 2015)

und deshalb $j = 3$ und

$$etype(p*.fson, f) = t'_j = DTE_p = DTE * .$$

Daraus wiederum folgt

$$etype(p*.fson*, f) = t = DTE = \{ \text{uint label}; \dots \},$$

und mit $i = 1$ folgt

$$etype(e, f) = \text{uint}.$$

Für den left value müssen wir nur folgendes machen:

$$lv(e, c) = lv(p*.fson*, c).label = va(p*.fson).label.$$

Der Wert von $p*.fson$ ist wie folgt definiert:

$$va(p*.fson, c) = va(p*, c)(fson) = c.m(va(p, c))(fson).$$

Wie zuvor stellen wir fest dass $p = n''_1$ mit

$$cf(c) = \{t''_1 n''_1; \dots; t''_3 n''_3\}$$

ist und daher

$$va(p, c) = c.m(top(c).p) = 0.$$

Daraus folgern wir direkt

$$lv(e, c) = c.m(0)(fson).label = t_1(fson).label = 1.label.$$

Es bleibt noch den Wert des Ausdrucks zu berechnen, dieser ist aber genau der Wert der Subvariable $lv(e, c)$ in c :

$$va(e, c) = c.m(lv(e, c)) = c.m(1)(label) = t_2(label) = 'FuDS'$$

$e = \text{result\&}$: Wie wir oben sehen können gilt

$$\text{result} = n''_3$$

mit

$$cf(c) = \{t''_1 n''_1; \dots; t''_3 n''_3\},$$

und daher ist dieser Ausdruck nicht gültig (Adressen von Stackvariablen sind ungültig).

$e = \text{seq}$: Diese Variable ist weder im topframe noch im global memory, und daher sind alle drei Funktionen undefiniert.

System Architecture - SS15
Exercise Sheet 9(due: June 22, 2015)

Geben Sie eine Simulationsrelation \sim_{tree}^c zwischen Ableitungsbäumen und Instanzen des Typs *DTE* in einer Konfiguration c an. Benutzen Sie 'x', um Label x als `uint` zu kodieren.

Tipp: Definieren Sie für $v \in ra(DTE)$, $i \in \mathbb{N}$ und C0-Konfigurationen c

$$v \circ_c i \in ra(DTE).$$

Solution: Für einen Knoten $u \in T.A$ und einen Wert $v \in ra(DTE)$ definieren wir:

$$u \sim_{tree}^c v \iff \forall i. u \circ i \in T.A \rightarrow u \circ i \sim_{tree}^c v \circ_c i \wedge c.m(v \circ_c i(\mathbf{father})) = v$$

$$\wedge 'T.l(u)' = v(\mathbf{label}),$$

$$v \circ_c i = b_c^i(c.m(v(\mathbf{fson}))),$$

$$b_c(v) = c.m(v(\mathbf{bro})).$$

Es gibt auch Bisimulationsrelationen, in denen keinen zusätzlichen Brüder, Söhne, oder Väter vorhanden sein dürfen. Für unsere Zwecke reicht diese Definition.