

System Architecture - SS15
Exercise Sheet 10(due: June 29, 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: **(2)**

Erweitern Sie die Sprache C0 um ein `if` für Pointer. Definieren sie insbesondere Semantik für

$$\text{if } e \{s_1\}\text{else}\{s_2\}$$

wenn $\text{etype}(e, f) = t*$. Dabei soll gelten die Statementsequenz s_1 ausgeführt werden wenn e nicht zum Nullpointer ausgewertet, sonst s_2 .

Solution: Da die if-statements in der Grammatik nur boolesche Ausdrücke erlauben, fügen wir die Produktion

$$\langle St \rangle \rightarrow \text{if } \langle F \rangle \{ \langle StS \rangle \} \text{ else } \{ \langle StS \rangle \}$$

hinzu.

Wir definieren nun c' wenn

$$c.pr[1] = \text{if } e \{s_1\}\text{else}\{s_2\},$$

wobei wir unspezifizierte Komponenten unverändert lassen.

Man beachte, dass die Grammatik an dieser Stelle ambig ist, da sich identifier nun als BE oder als F ableiten lassen. Anhand der Typprüfung von Ausdrücken können wir jedoch einfach zwischen den beiden unterscheiden.

Ist also

$$e = \text{null} \vee \text{etype}(e, c) = t*,$$

System Architecture - SS15
Exercise Sheet 10(due: June 29, 2015)

dann gilt $va(e, c) \in SV(c) \cup \{null\}$ und wir definieren:

$$c'.pr = \begin{cases} s_1; \text{tail}(c.pr) & va(e, c) \neq null \\ s_2; \text{tail}(c.pr) & \text{o.w.} \end{cases}$$

Ist anderenfalls

$$etype(e, c) = bool,$$

dann gilt wie zuvor

$$c'.pr = \begin{cases} s_1; \text{tail}(c.pr) & va(e, c) = 1 \\ s_2; \text{tail}(c.pr) & \text{o.w.} \end{cases}$$

Alle anderen Fälle schliessen wir per Kontextbedingung aus.

Aufgabe 2:

(2)

Definieren Sie die Semantik des return Statements!

Solution: Sei

$$c.pr[1] = \text{return } e,$$

und $etype(e, c) = vtype(c.rds(c.rd), c)$. Wir definieren c' , wobei wir wie zuvor unspezifizierte Komponenten unverändert lassen.

$$\begin{aligned} c'.pr &= \text{tail}(c.pr) \\ c'.rd &= c.rd - 1 \\ c'.m(c.rds(c.rd)) &= va(e, c) \end{aligned}$$

Man beachte, dass sich die Funktionen $c'.st$ und $c.st$ sowie $c'.rds$ und $c.rds$ jeweils im Typ unterscheiden, aber dennoch

$$c'.F(x) = c.F(x)$$

für alle $x \in \text{Dom}(c'.F)$.

Aufgabe 3:

(4)

Im Program von Aufgabe 3, Blatt 8, wollen wir die Funktion `se` implementieren. Dabei soll mit der Simulationsrelation aus Aufgabe 2, Blatt 9, gelten dass wenn immer

$$c.pr[1] = e = \text{se}(p, i) \wedge u \sim_{tree}^c c.m(va(p, c)),$$

dann auch

$$se(u, va(i, c)) \sim_{tree}^{c'} c'.m(va(e, c')),$$

wobei c' die Konfiguration nach der Rückkehr aus der Funktion `se` ist.

System Architecture - SS15
 Exercise Sheet 10(due: June 29, 2015)

Solution: Wir erinnern uns:

$$se(u, i) = u2^{i-1}0$$

Das programmieren wir direkt:

```
DTEp se(DTEp seq, uint index)
{
  while index > 1 {
    seq = nson(seq,2) ;
    index = index - 1;
  };
  seq = nson(seq,0);
  return seq
}
```

Aufgabe 4:

(8)

Bestimmen Sie das Memory-Layout in folgender Konfiguration c im Programm von Aufgabe 3, Blatt 8, für alle explizit angegebenen Variablen. (Achtung: $c.m(top(c).p) = 0$, statt wie auf Blatt 9!)

$$\begin{aligned}
 t_1(\text{label}) &= \text{'Prog' } \\
 t_1(\text{father}) &= \text{null} \\
 t_1(\text{fson}) &= 1 \\
 t_1(\text{bro}) &= \text{null} \\
 t_2(\text{label}) &= \text{'FuDS' } \\
 t_2(\text{father}) &= 0 \\
 t_2(\text{fson}) &= \dots \\
 t_2(\text{bro}) &= \text{null} \\
 c.m(x) &= \begin{cases} \text{null} & x = \text{top}(c).\text{result} \\ 1 & x = \text{top}(c).\text{j} \\ 0 & x = \text{top}(c).\text{p} \\ 0 & x = \text{gm.}\text{programtree} \\ t_1 & x = 0 \\ t_2 & x = 1 \\ \dots & \dots \end{cases} \\
 c.st(c.rd) &= \text{nson} \\
 c.ht(0) &= \text{DTE} \\
 c.ht(1) &= \text{DTE} \\
 c.nh &= 2
 \end{aligned}$$

System Architecture - SS15
Exercise Sheet 10(due: June 29, 2015)

Solution:

0_{32}	$hbase +_{32} 28_{32}$
$ba(\dots, c)$	$hbase +_{32} 24_{32}$
$hbase$	$hbase +_{32} 20_{32}$
$'FuDs'_{32}$	$hbase +_{32} 16_{32}$
0_{32}	$hbase +_{32} 12_{32}$
$hbase +_{32} 16_{32}$	$hbase +_{32} 8_{32}$
0_{32}	$hbase +_{32} 4_{32}$
$'Prog'_{32}$	$hbase$
\dots	\dots
0_{32}	$ba(top(c), c) +_{32} 8_{32}$
1_{32}	$ba(top(c), c) +_{32} 4_{32}$
0_{32}	$ba(top(c), c)$
\dots	\dots
$ST(0, c)$	$sbase +_{32} 12_{32}$
0_{32}	$sbase$