Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl.-Ing. Christoph Baumann

10. Übungsblatt Informatik II

(Abgabe: 07.07.2008)

Aufgabe 1: (Ableitungsbäume) **(7 Punkte)**

Wir betrachten zu einer kontextfreien Grammatik $G = (T, N, s, P)$ einen Ableitungsbaum mit Wurzel s und Blattwort uvw. Es gilt also

$$s \longrightarrow_G^* uvw$$

mit $u, w \in (N \cup T)^*$ und $v \in N$. Weiterhin erlaubt die Grammatik die Ableitung

$$v \longrightarrow_G^* x$$

für $x \in (N \cup T)^*$ in endlich vielen Ableitungsschritten. Beweise formal:

$$s \longrightarrow_G^* uxw$$

Aufgabe 2: (Kontextbedingungen) **(4+2+3+5+2 Punkte)**

In der Vorlesung wurden Kontextbedingungen erwähnt, die C_0-Programme erfüllen müssen um ausgeführt werden zu können. Wir wollen nun einige dieser Bedingungen für Variablendeklarationen und Zuweisungen formalisieren.

a) Gegeben sei eine Variablendeklaration

```
t na
```

Damit Variable na in die Symboltabelle $st \in \{st.gm, ft(f).st\}$ eingetragen werden kann, muss der Typ t in der Typtabelle vorhanden sein. Gib eine mathematische Definition dieser Bedingung an! Spezifiziere die Symboltabelle st', die sich ergibt, wenn na in st eingetragen wird!

b) Im Folgenden seien Typtabelle tt, Funktionstabelle ft und die Symboltabelle des global memory $st.gm$ vollständig angelegt. Wir betrachten die Zuweisung

$$id = e$$

von Ausdruck e an Bezeichner id im Rumpf der Funktion f. Damit die Zuweisung zulässig ist, muss id eine deklarierte Variable oder Subvariable sein. Formalisiere diese Bedingung!

c) Seien x und y deklarierte Variablen und `x=y` eine Zuweisung im Rumpf der Funktion f. Der Ausdruck lässt sich syntaktisch bis in die Form $\langle id \rangle = \langle id \rangle$ zurückführen. Von dieser Stelle an legt die Grammatik nicht mehr eindeutig fest, ob die rechte Seite einen arithmetischen Ausdruck $\langle A \rangle$ oder einen booleschen Ausdruck $\langle BA \rangle$ darstellt. Formuliere eine Kontextbedingung, unter der `x=y` als $\langle id \rangle = \langle BA \rangle$ interpretiert werden muss.

d) Nun sollen formale Kontextbedingungen für die Typkorrektheit von Zuweisungen erstellt werden. Wir betrachten erneut die Zuweisung $id = e$. Formuliere Bedingungen an $typ(id)$ für folgende Fälle von e.

- $e \equiv x$
- $e \equiv -7$
- $e \equiv 1$
- $e \equiv$'c'
- $e \equiv x\&$

x sei dabei ein Identifier.

e) Betrachte die folgende syntaktisch korrekte Zuweisung:

```
x&=y&
```

Warum darf diese semantisch gesehen nicht zulässig sein? Woran scheitert die Ausführung zur Laufzeit, wenn man diesen Fehler nicht abfangen würde?

Aufgabe 3: (CO-Anweisungsausführung) **(7+10 Punkte)**

In Abschnitt 4.5 der Vorlesungsmitschrift vom SS07 wird an verschiedenen Beispielen erläutert, wie mit Hilfe der formalen C_0-Semantik die Ausführung von Code simuliert und über Eigenschaften von C_0-Programmen Beweise geführt werden können. Gegeben sei nun folgendes C_0-Programm:

```
int a;                int function(int a) {
bool* p;              int b;
                      b = a;
int main(){           while b > 10 do {b = b - 9 };
int b;                p* = 0;
int c;                return b}
p = new bool*;
a = 27;
b = function(a);
return 0};
```

a) Gib für das C_0-Programm die Initialkonfiguration c^0, die Typtabelle und die Funktionstabelle an!

b) Beginne das Programm von c^0 ausgehend durchzuführen! Gib dazu nach jeder Ausführung die neue Konfiguration an!

Aufgabe 4 BONUS: (While-Schleife) **(10* Punkte)**

Betrachte das C_0-Programm mit der while-Schleife in Abschnitt 4.5.4 der Vorlesungsmitschrift vom SS07!

Sei $va(c^\alpha, \texttt{result}) = K$ und $va(c^\alpha, \texttt{n}) = M$ und $c^\alpha.pr = \texttt{while n>0 do } \{...\}$.
Dann soll nach M weiteren Durchläufen der Schleife gelten:

- $va(c^{\alpha+3M}, \texttt{result}) = K + M + (M-1) + ... + 1$
- $va(c^{\alpha+3M}, \texttt{n}) = 0$
- $c^{\alpha+3M}.pr : \texttt{while n>0 do } \{...\}$

Beweise die Behauptung durch vollständige Induktion über M!