Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Theoretische Informatik

Sommersemester 2023
7. Aufgabenblatt

Aufgabe 26 (Semantik von Programmen [klausurähnlich], 1+1+1+1 Punkte)
Gegeben seien die folgenden zwei Programme, wobei angenommen werden kann, dass alle Variablen einen ganzzahligen Typen haben.

P1 ≡
```
x := y;
y := x;
```

P2 ≡
```
z := 1;
while(not (x == 0)){
   z := z + 1;
   x := x - z;
}
```

Betrachtet werden die folgenden zwei Zustände z1 und z2 mit: z1(x)=2, z1(y)=3 und z2(x)=1, z2(y)=4. Berechnen Sie für alle Programme (i=1,2) und Zustände (j=1,2) SemPart(Pi,zj). Es reicht jeweils die Angabe des Ergebnisses.
[in der Klausur könnte auch nach Sem(Pi,zj) gefragt werden]

Aufgabe 27 (Semantik von Programmen [klausurähnlich], 1+1+1+1 Punkte)
Gegeben Seien die folgenden zwei Programme, wobei angenommen werden kann, dass alle Variablen einen ganzzahligen Typen haben.

P1 ≡
```
u := 1;
if(x > u) {
     y := y - 1;
} else {
     y := y + 1;
}
```

P2 ≡
```
while(y > x){
     x := x - 1;
     y := y - 2;
}
```

Betrachtet werden die folgenden zwei Zustände z1 und z2 mit: z1(x)=2, z1(y)=3 und z2(x)=1, z2(y)=4. Berechnen Sie für alle Programme (i=1,2) und Zustände (j=1,2) SemPart(Pi,zj). Es reicht jeweils die Angabe des Ergebnisses.
[in der Klausur könnte auch nach Sem(Pi,zj) gefragt werden]

Aufgabe 28 (formale Ausführung von Programmen)

a) Führen Sie die beiden Programme aus der Aufgabe 26 mit den beiden Zuständen schrittweise mit den Transitionsregeln solange aus, bis das Ergebnis feststeht.
b) Lesen Sie als Hintergrund Kapitel 4.4 der Dokumentation zur theoriesammlung von der Veranstaltungswebseite. Führen Sie die Programme aus den vorherigen Aufgaben soweit sinnvoll mit der Bibliothek aus.

Aufgabe 29 (systematische Auswertung von Zusicherungen)
Gegeben seien folgende Zusicherungen:

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

$p_1 \equiv$ x+y < y+y $p_2 \equiv$ x+y = y+x $p_3 \equiv$ (x>7 $\vee$ y>6) $\rightarrow$ (x*y >42)

a) Berechnen Sie schrittweise die Semantik der Zusicherungen für den Zustand z mit z(x)=6 und z(y)=7.
b) Berechnen Sie für jede der Zusicherungen p_i die resultierende Zusicherung für p_i[y:=1] und p_i[y:=x] und vereinfachen Sie wenn möglich.
c) Geben Sie möglichst kompakte Beschreibungen für Sem(p_i) an.

Aufgabe 30 (Erweiterung von Syntax und Semantik)
Lesen Sie als Hintergrund Kapitel 4.4 und 4.5 der Dokumentation zur theoriesammlung von der Veranstaltungswebseite.
Erweitern Sie die Basisprogrammiersprache um die Regel
`Befehl -> do{Sequenz}while(Bedingung)`. Der einfachste Ansatz ist es, eine Kopie der Klasse SemantikBasissprache zu machen, sich den dortigen Ansatz anzusehen, bei dem alle Regeln programmiert werden und ihnen dann eine Berechnung mit zugeordnet wird. Sie ergänzen dann im nächsten Schritt die neue Regel und die zugehörige Semantik, genauer ein passendes Berechnungsobjekt mit `sem.umsetzungHinzu(regel, berechnung);` zum Semantik-Objekt sem. Am Ende sollte das folgende Programm laufen.

```
public static void main(String[] args) {
  SemantikBasissprachеMitDo.ausfuehren("""
      x := 3;
      u := 0;
      z := 0;
      do{
        u := u + x;
        z := z + 1;
      } while (z < x)
      """);
}
```

Das Programm sollte folgende Ausgabe produzieren.

```
:= :     {x=3}
:= :     {u=0, x=3}
:= :     {u=0, x=3, z=0}
do:      {u=0, x=3, z=0}
:= :     {u=3, x=3, z=0}
:= :     {u=3, x=3, z=1}
do t:    {u=3, x=3, z=1}
:= :     {u=6, x=3, z=1}
:= :     {u=6, x=3, z=2}
do t:    {u=6, x=3, z=2}
:= :     {u=9, x=3, z=2}
:= :     {u=9, x=3, z=3}
do f:    {u=9, x=3, z=3}
```

Hinweis: Generell ist auch eine Grammatik in Textform nutzbar, bei der alle Regeln durch voranstehende Namen identifiziert werden.

```
r1:: AX -> BX | a
r3:: BX -> /eps
```

Mit der Methode `g.regelZuId("r1_0")` bzw. `g.regelZuId("r1_1")` bzw. `g.regelZuId("r3")` kann dann das passende Regelobjekt der Grammatik g erhalten werden.