4. Übung zur
Konstruktion fehlerfreier Software

Frank Pilhofer

Montag 13-14 Uhr

0. Zuweisungen in einer Datenumgebung

Sei $d^0 = [(x,R,9.3),(y,R,2),(z,Z,3),(z,Z,4),(w,Z,6)]$. Wir nehmen an, daß eine reelle Zahl bei Zuweisung an eine ganzzahlige Variable zur nächsten ganzen Zahl auf- bzw. abgerundet wird.

0.0 $x := 1.4y + z$

$d^1 = [(x,R,5.8),(y,R,2),(z,Z,3),(z,Z,4),(w,Z,6)]$

0.1 $z := z + y$

$d^1 = [(x,R,9.3),(y,R,2),(z,Z,5),(z,Z,4),(w,Z,6)]$

0.2 $z := x + y$

$d^1 = [(x,R,9.3),(y,R,2),(z,Z,11),(z,Z,4),(w,Z,6)]$

0.3 $(z,w) := (y+z+w, 2y+3z+4w)$

$d^1 = [(x,R,9.3),(y,R,2),(z,Z,11),(z,Z,4),(w,Z,37)]$

0.4 $(z,z) := (y+z+w, 2y+3z+4w)$

d^1 ist nicht definiert, da z zwei verschiedene Werte gleichzeitig zugewiesen werden sollen.

0.5 $(z,z) := (y+z+w, y \cdot w - 1)$

$d^1 = [(x,R,9.3),(y,R,2),(z,Z,11),(z,Z,4),(w,Z,6)]$

0.6 $(y,w) := (w,y)$

$d^1 = [(x,R,9.3),(y,R,6),(z,Z,3),(z,Z,4),(w,Z,2)]$

1. Mehr Zuweisungen in einer Datenumgebung

Sei $d^0 = \begin{bmatrix} (k,Z,2),(y(0),R,2),(y(1),R,3.2),(j,Z,1), \\ (y(2),R,5.1),(j,Z,3),(y(3),R,7.5),(y(2),Z,9) \end{bmatrix}$.

1.0 $y(j) := y(k) + 3$

$(y(j) := y(k) + 3).d^0 = (y(j.d) := y(k.d) + 3).d^0 = (y(1) := y(2) + 3).d^0$

$= \quad d^1 = \begin{bmatrix} (k,Z,2),(y(0),R,2),(y(1),R,8.1),(j,Z,1), \\ (y(2),R,5.1),(j,Z,3),(y(3),R,7.5),(y(2),Z,9) \end{bmatrix}$

1.1 $(j, y(j)) := (j+1, y(k)+3)$

$((j, y(j)) := (j+1, y(k)+3)).d^0$

= $((j, y(j.d)) := (j.d+1, y(k.d)+3)).d^0$

= $((j, y(1)) := (1+1, y(2)+3)).d^0$

= $((j, y(1)) := (2, 8.1)).d^0$

= $d^1 = \begin{bmatrix} (k,Z,2),(y(0),R,2),(y(1),R,8.1),(j,Z,2), \\ (y(2),R,5.1),(j,Z,3),(y(3),R,7.5),(y(2),Z,9) \end{bmatrix}$

2. if-Anweisungen

Sei $d^0 = [(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$.

2.0 if $x < 0$ then $y := -x$ else $y := x$ endif

$d^1 = [(x,R,9.3),(y,R,9.3),(z,Z,3),(w,Z,6)]$

2.1 if $x > 0$ then $y := -x$ else $y := x$ endif

$d^1 = [(x,R,9.3),(y,R,-9.3),(z,Z,3),(w,Z,6)]$

2.2 if $r > 0$ then $y := -x$ else $y := x$ endif

Undefiniert, da *r* nicht definiert ist.

2.3 if $x > 0$ then $r := -x$ else $y := x$ endif

Undefiniert, da *r* nicht definiert ist.

2.4 if $x < 0$ then $r := -x$ else $y := x$ endif

$d^1 = [(x,R,9.3),(y,R,9.3),(z,Z,3),(w,Z,6)]$

3. Folgen von Anweisungen

Sei $d^0 = [(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$.

3.0 $z := y+z+w;\quad w := 2y+3z+4w$

$(z := y+z+w;\quad w := 2y+3z+4w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$

= $(w := 2y+3z+4w).((z := y+z+w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)])$

= $(w := 2y+3z+4w).[(x,R,9.3),(y,R,2),(z,Z,11),(w,Z,6)]$

= $d^1 = [(x,R,9.3),(y,R,2),(z,Z,11),(w,Z,61)]$

3.1 $x := y \cdot z; \quad y := w$

$$(x := y \cdot z; \quad y := w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad (y := w).((x := y \cdot z).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)])$$
$$= \quad (y := w).[(x,R,6),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad d^1 \quad = \quad [(x,R,6),(y,R,6),(z,Z,3),(w,Z,6)]$$

3.2 $y := w; \quad x := y \cdot z$

$$(y := w; \quad x := y \cdot z).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad (x := y \cdot z).((y := w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)])$$
$$= \quad (x := y \cdot z).[(x,R,9.3),(y,R,6),(z,Z,3),(w,Z,6)]$$
$$= \quad d^1 \quad = \quad [(x,R,18),(y,R,6),(z,Z,3),(w,Z,6)]$$

3.3 $x := x + w \cdot z; \quad y := w$

$$(x := x + w \cdot z; \quad y := w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad (y := w).((x := x + w \cdot z).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)])$$
$$= \quad (y := w).[(x,R,27.3),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad d^1 \quad = \quad [(x,R,27.3),(y,R,6),(z,Z,3),(w,Z,6)]$$

3.4 $y := w; \quad x := x + w \cdot z$

$$(y := w; \quad x := x + w \cdot z).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad (x := x + w \cdot z).((y := w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)])$$
$$= \quad (x := x + w \cdot z).[(x,R,9.3),(y,R,6),(z,Z,3),(w,Z,6)]$$
$$= \quad d^1 \quad = \quad [(x,R,27.3),(y,R,6),(z,Z,3),(w,Z,6)]$$

3.5 $x := y \cdot z; \quad y := w; \quad z := 2x - w$

$$(x := y \cdot z; \quad y := w; \quad z := 2x - w).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]$$
$$= \quad (z := 2x - w).((y := w).((x := y \cdot z).[(x,R,9.3),(y,R,2),(z,Z,3),(w,Z,6)]))$$
$$= \quad (z := 2x - w).((y := w).[(x,R,6),(y,R,2),(z,Z,3),(w,Z,6)])$$
$$= \quad (z := 2x - w).[(x,R,6),(y,R,6),(z,Z,3),(w,Z,6)]$$
$$= \quad d^1 \quad = \quad [(x,R,6),(y,R,6),(z,Z,6),(w,Z,6)]$$

4. Definitionsbereich einer Zuweisung

Damit eine Zuweisung $x := A$ in einer Datenumgebung d definiert ist, muß

- eine Variable mit dem Namen x in d existieren: $\exists i: N_i = x$
- der Ausdruck A muß in d definiert sein: $d \in \mathrm{Def}_A$
- $A.d$ muß zum Typ des ersten x in d kompatibel sein: $A.d \in W_i \wedge N_i = x$

Insgesamt: $\mathrm{Def}_{x:=A} = \left\{ d^j \in D \middle| d^j \in \mathrm{Def}_A \wedge \exists i: \left(N_i^j = x \wedge \left(\forall k: \left(k \geq i \vee N_k^j \neq x \right) \right) \wedge A.d \in W_i \right) \right\}$.

5. Definitionsbereich einer mehrfachen Zuweisung

Damit eine mehrfache Zuweisung $(x_1 \dots x_n) := (A_1 \dots A_n)$ definiert ist, müssen natürlich sämtliche Einzelzuweisungen in der Datenumgebung definiert sein. Da einer Variablen nicht gleichzeitig zwei Werte zugewiesen werden können, müssen die einzelnen Variablennamen paarweise unterschiedlich sein.

Daher: $\mathrm{Def}_{(x_1 \dots x_n) := (A_1 \dots A_n)} = \left(\forall (i,j) : \left(i = j \vee x_i \neq x_j\right)\right) \bullet \bigcap_{i=1}^{n} \mathrm{Def}_{x_i := A_i}$

Falls der logische Term falsch ist, so ist der Definitionsbereich leer.

6. Definitionsbereich einer if-Anweisung

- Die Bedingung muß definiert sein.
- Falls die Bedingung wahr ist, so muß der then-Zweig definiert sein.
- Falls die Bedingung falsch ist, so muß der else-Zweig definiert sein.

Also: $\mathrm{Def}_{\text{if } B \text{ then } S_1 \text{ else } S_2 \text{ endif}} = \left\{ d^j \in D \middle| d^j \in \mathrm{Def}_B \wedge \left(\left(B.d^j\right) \bullet \left(d^j \in \mathrm{Def}_{S_1}\right) \vee \overline{\left(B.d^j\right)} \bullet \left(d^j \in \mathrm{Def}_{S_2}\right) \right) \right\}$

7. Definitionsbereich einer Folge von zwei Anweisungen

Das Ergebnis einer Folge $(S_1, S_2).d$ von Anweisungen ist definiert, wenn $S_1.d = d'$ und $S_2.d'$ definiert sind. Also: $\mathrm{Def}_{S_1,S_2} = \left\{ d^j \in D \middle| d^j \in \mathrm{Def}_{S_1} \wedge S_1.d^j \in \mathrm{Def}_{S_2} \right\}$.

Diese Definition läßt sich leicht rekursiv auf eine Folge von Anweisungen beliebiger Länge erweitern: $\mathrm{Def}_{S_1,\dots,S_n} = \left\{ d^j \in D \middle| d^j \in \mathrm{Def}_{S_1} \wedge S_1.d^j \in \mathrm{Def}_{S_2,\dots,S_n} \right\}$.