Übungsblatt 3

Abgabe: 9.1.2008

Wir betrachten wieder das Programm

```
P1: if(x % 2){
P2:    x = x + 1;
    }
p3: exit;
```

Als Menge der Locations Loc betrachten wir

$$Loc =_{def} \{P_1, P_2, P_3\}$$

wobei jedes Element lediglich eine Abkürzung für den in der jeweiligen Zeile noch auszuführenden Programmtext darstellt.

Als Menge von Symbolen X verwenden wir für unser konkretes Programm lediglich:

$$X =_{def} \{x\}$$

Die Variable x stellt einen Integer dar. Da sie als Eingabe für das Programm verwendet wird ist ihr Wert immer definiert. Somit gilt für die Menge der Valuationsfunktionen für die konkrete Semantik des Programmes:

$$\Sigma =_{def} \{x\} \to int$$

Der Zustandsraum der konkreten Semantik dieses Programmes ist somit:

$$S =_{def} Loc \times \Sigma$$

Als Referenz für abstrakte Interpretationen verwenden wir als Zustandsraum jedoch den Potenzmengenverband über den konkreten Zustandsraum, also:

$$S_{\mathbb{P}} =_{def} \mathbb{P}(Loc \times \Sigma)$$

Für die Zustandsübergangsrelation $\longrightarrow_{\mathbb{P}} \in S_{\mathbb{P}} \times S_{\mathbb{P}}$ ergeben sich folgende operative Regeln:

- (I)

$$\frac{A \subseteq \Sigma}{\begin{array}{c}\{(P_1, \sigma) \mid \sigma \in A\} \\ \longrightarrow_{\mathbb{P}} \\ \{(P_2, \sigma) \mid \sigma \in A \wedge \sigma(x) \ \% \ 2 = 1\} \cup \\ \{(P_3, \sigma) \mid \sigma \in A \wedge \sigma(x) \ \% \ 2 = 0\}\end{array}}$$

- (II)

$$\frac{A \subseteq \Sigma}{\begin{array}{c}\{(P_2, \sigma) \mid \sigma \in A\} \\ \longrightarrow_{\mathbb{P}} \\ \{(P_3, \sigma[x \mapsto \sigma(x) + 1]) \mid \sigma \in A\}\end{array}}$$

- (III)

$$\frac{A \subseteq \Sigma}{\{(P_3, \sigma) \mid \sigma \in A\} \longrightarrow_{\mathbb{P}} \{(P_3, \sigma) \mid \sigma \in A\}}$$

Der Programmierer des Programms behauptet, dass für Eingaben von

$$x = 1, x = 2, \ldots, x = 10$$

folgendes gilt:

1. x ist nach Ausführung eine gerade Zahl
2. x liegt nach Ausführung zwischen 2 und 10

Aufgabe 1: Berechnung im Potenzmengenverband (10%)

- Findet einen Startzustand $S_{0_{\mathbb{P}}} \in S_{\mathbb{P}}$, der den Einschränkungen des Programmierers über mögliche Eingabedaten entspricht.
- Führt die Berechnung durch

Aufgabe 2: Feinkörnige Abstraktion über $\mathbb{I}(int)$ (40%)

Für Werte von x betrachten wir nun anstatt von Integern Intervalle über Integer, also den Verband:

$$L_1 =_{def} (\mathbb{I}(int), \subseteq)$$

Entsprechend kann eine neue Menge von abstrakten Valuationsfunktionen eingeführt werden:

$$\Sigma_{L_1} =_{def} \{x\} \to \mathbb{I}(int)$$

Unsere abstrakte Interpretation über den Intervallverband verwendet dann als Zustandsraum:

$$S_{L_1} =_{def} \mathbb{P}(Loc \times \Sigma_{L_1})$$

Die Transitionsrelation $\longrightarrow_{L_1} \in S_{L_1} \times S_{L_1}$ wird nun nach den in der Vorlesung vorgestellten Regeln konstruiert. Hierzu wird jedoch noch etwas Handwerkszeug benötigt:

Aufgabe 2.1: Galois-Verbindungen

- Konstruiert eine Galois-Verbindung zwischen $\mathbb{P}(int)$ und $\mathbb{I}(int)$
- Konstruiert damit eine Galois-Verbindung zwischen $S_{\mathbb{P}}$ und S_{L_1}
- Startzustand $S_{0_{L_1}} = \{(P_1, [x \mapsto [1,1]]), \ldots, (P_1, [x \mapsto [10,10]])\}$ entsteht aus ${S_{0_{\mathbb{P}}}}^{\triangleright}$. Weist dies nach

Die bestmögliche Zustandsübergangsrelation $\longrightarrow_{L_1} \in S_{L_1} \times S_{L_1}$ erfüllt folgender Regeln ($p \in S_{\mathbb{P}}, q \in S_{L_1}$):

- (IV)

$$\frac{p^{\triangleright\triangleleft} \longrightarrow_{\mathbb{P}} p'}{p^{\triangleright} \longrightarrow_{L_1} p'^{\triangleright}}$$

- (V)

$$\frac{q^{\triangleleft} \longrightarrow_{\mathbb{P}} p'}{q \longrightarrow_{L_1} p'^{\triangleright}}$$

Diese Regeln sind zur Berechnung einer (abstrakten) Programmausführung jedoch ungeeignet. Beide Regeln haben als Prämisse jedoch einen Zustandsübergang in $\longrightarrow_{\mathbb{P}}$, und entsprechend können die Regeln für $\longrightarrow_{\mathbb{P}}$ verwendet werden, um explizite Regeln für $\longrightarrow_{L_1}$ herzuleiten.

Für die vorliegende Aufgabe reichen die Regeln:

- (VI)

$$\frac{}{\begin{array}{c}\{(P_1,[x\mapsto[1,1]]),\ldots,(P_1,[x\mapsto[10,10]])\}\\ \longrightarrow_{L_1}\\ \{(P_2,[x\mapsto[1,1]]),(P_2,[x\mapsto[3,3]])\}\ldots,(P_2,[x\mapsto[9,9]])\}\cup\\ \{(P_3,[x\mapsto[2,2]]),(P_3,[x\mapsto[4,4]])\ldots,(P_3,[x\mapsto[10,10]])\}\end{array}}$$

- (VII)

$$\frac{}{\begin{array}{c}\{(P_2,[x\mapsto[1,1]]),(P_2,[x\mapsto[3,3]])\}\ldots,(P_2,[x\mapsto[9,9]])\}\\ \longrightarrow_{L_1}\\ \{(P_3,[x\mapsto[2,2]]),(P_3,[x\mapsto[4,4]])\ldots,(P_3,[x\mapsto[10,10]])\}\end{array}}$$

Aufgabe 2.2: Zustandsübergangsrelation $\longrightarrow_{L_1}$

- Zeigt, dass Regel (VI) die Regel (IV) erfüllt
- Zeigt, dass Regel (VI) die Regel (V) erfüllt
- Zeigt, dass Regel (VII) die Regel (IV) erfüllt
- Zeigt, dass Regel (VII) die Regel (V) erfüllt

Aufgabe 2.3: Berechnung in S_{L_1}

- Führt die Berechnung ausgehend von Startzustand $S_{0_{L_1}}$ durch.

Aufgabe 3: Grobkörnige Abstraktion über $\mathbb{I}(int)$ (40%)

Für Werte von x betrachten wir weiter Intervalle, aber Locations fallen jetzt zusammen. Unsere abstrakte Interpretation über den Intervallverband verwendet dann als Zustandsraum:

$$S_{L_2} =_{def} Loc \not\to \Sigma_{L_1}$$

Die Transitionsrelation $\longrightarrow_{L_2} \in S_{L_2} \times S_{L_2}$ wird nach den in der Vorlesung vorgestellten Regeln konstruiert. Hierzu wird jedoch noch etwas Handwerkszeug benötigt:

Aufgabe 3.1: Galois-Verbindungen

- Konstruiert eine Galois-Verbindung zwischen S_{L_1} und S_{L_2}
- Startzustand $S_{0_{L_2}} = [P_1 \mapsto [x \mapsto [1, 10]]]$ entsteht aus ${S_{0_{L_1}}}^{\triangleright}$. Weist dies nach

Die bestmögliche Zustandsübergangsrelation $\longrightarrow_{L_2} \in S_{L_2} \times S_{L_2}$ erfüllt folgender Regeln ($p \in S_{L_1}, q \in S_{L_2}$):

- (VIII)
$$\frac{p^{\triangleright\triangleleft} \longrightarrow_{L_1} p'}{p^{\triangleright} \longrightarrow_{L_2} p'^{\triangleright}}$$
- (IX)
$$\frac{q^{\triangleleft} \longrightarrow_{L_1} p'}{q \longrightarrow_{L_2} p'^{\triangleright}}$$

Diese Regeln sind zur Berechnung einer (abstrakten) Programmausführung jedoch ungeeignet. Beide Regeln haben als Prämisse jedoch einen Zustandsübergang in $\longrightarrow_{L_1}$, und entsprechend können die Regeln für $\longrightarrow_{L_1}$ verwendet werden, um explizite Regeln für $\longrightarrow_{L_2}$ herzuleiten.

Für die vorliegende Aufgabe reichen die Regeln:

- (X)
$$\frac{}{[P_1 \mapsto [x \mapsto [1, 10]]] \longrightarrow_{L_2} [P_2 \mapsto [x \mapsto [1, 9]], P_3 \mapsto [x \mapsto [2, 10]]]}$$
- (XI)
$$\frac{}{[P_2 \mapsto [x \mapsto [1, 9]]] \longrightarrow_{L_2} [P_3 \mapsto [x \mapsto [2, 10]]]}$$

Aufgabe 3.2: Zustandsübergangsrelation $\longrightarrow_{L_2}$

- Zeigt, dass Regel (X) die Regel (VIII) erfüllt
- Zeigt, dass Regel (X) die Regel (IX) erfüllt
- Zeigt, dass Regel (XI) die Regel (VIII) erfüllt
- Zeigt, dass Regel (XI) die Regel (IX) erfüllt

Hinweis: Hierzu wird man möglicherweise weitere Übergänge in $\longrightarrow_{L_1}$ mithilfe der Regeln für $\longrightarrow_{\mathbb{P}}$ als legitim nachweisen müssen.

Aufgabe 3.3: Berechnung in S_{L_2}

- Führt die Berechnung ausgehend von Startzustand $S_{0_{L_2}}$ durch.

Aufgabe 4: Einschätzung (10%)

Wir haben nun Berechnungen in den Verbänden

- $\mathbb{P}(Loc \times (\{x\} \to int))$
- $\mathbb{P}(Loc \times (\{x\} \to \mathbb{I}(int)))$
- $Loc \not\to (\{x\} \to int))$

durchgeführt. Wir betrachten nun die Aussagen des Programmierers aus der Einführung. Beantwortet folgende Fragen:

- Welche Verbände sind geeignet, um Aussage 1 zu überprüfen?
- Welche Verbände sind geeignet, um Aussage 2 zu überprüfen?
- Welcher Verband ist am effektivsten, um Aussage 1 zu überprüfen?
- Welcher Verband ist am effektivsten, um Aussage 2 zu überprüfen?

Begründet Eure Antworten.