

Übungen zur Semantik von Programmiersprachen

Aufgabe 1 (H) (*Regelinduktion*) (Abgabe: Montag, 5.11.01)

In Aufgabe 3, Blatt 2 wurde folgende Beziehung informell anhand des zugehörigen Ableitungsbaums begründet.

$$\begin{array}{l} \text{while } b_0 \vee b_1 \text{ do } s \\ \sim \text{ while } b_0 \text{ do } s; \text{ while } b_0 \vee b_1 \text{ do } s \end{array}$$

Beweisen Sie diese Aussage nun formal, indem Sie für beide Richtungen Regelinduktion anwenden.

Aufgabe 2 (Ü) (*Regelinduktion*)

Gegeben sei eine Relation $\rightarrow \subseteq Y \times Y$.

- (a) Definieren Sie formal mit Hilfe von Induktionsregeln die transitive, reflexive Hülle \rightarrow^* der Relation \rightarrow .
- (b) Zeigen Sie mit Regelinduktion, dass gilt:

$$\forall a, b, c. a \rightarrow^* b \wedge b \rightarrow^* c \Rightarrow a \rightarrow^* c$$

Aufgabe 3 (Ü) (*Äquivalenz von Auswertungssemantik und Transitionsemantik*)

Für die Sprache WHILE wurde in der Vorlesung eine alternative operationelle Semantik definiert. Hier soll nun eine Richtung der Äquivalenz von Auswertungssemantik \rightarrow und Transitionsemantik \rightarrow_1 gezeigt werden.

Beweisen Sie:

$$\forall s, \sigma, \sigma'. [\langle s, \sigma \rangle \rightarrow \sigma' \Rightarrow \langle s, \sigma \rangle \rightarrow_1^* \langle \text{skip}, \sigma' \rangle]$$

Aufgabe 4 (P) (*Transitionssemantik in PROLOG*)

(Abgabe: Montag, 5.11.01, 14 Uhr an kleing@in.tum.de)

- (a) In Aufgabe 5, Blatt 2 wurde die Sprache WHILE um ein *for*-Konstrukt erweitert. Neben *while* und *for* gibt es in imperativen Programmiersprachen oft auch ein *repeat*-Konstrukt, bei dem die Abbruchbedingung erst nach Ausführung des Schleifenkörpers getestet wird.

Die Syntax der Sprache WHILE aus der Vorlesung sei nun um

repeat *s* until *b*

erweitert.

Geben Sie die entsprechenden Regeln für die Transitionssemantik an. Die neuen Regeln sollen dabei nicht auf das **while** der Sprache zurückgreifen.

- (b) Die Transitionssemantik der erweiterten WHILE-Sprache soll in PROLOG programmiert werden. Es gelten die gleichen Vereinfachungsregeln für die abstrakte Syntax wie in Aufgabe 5, Blatt 2. Geben Sie Ihr Programm und ein Protokoll der Beispiele in der Datei `dialog.txt` ab.

Einen Rahmen zur Lösung dieser Programmieraufgabe finden Sie im Verzeichnis

`/usr/proj/semantik/prog/prolog/small-step/`

Hinweis:

Zum Anfertigen des Protokolls ist das Programm `script <dateiname>` hilfreich. Das Programm öffnet eine neue Shell, in der Sie dann den Prolog-Interpreter starten können. Alles, was in dieser Shell auf dem Bildschirm ausgegeben wird, schreibt `script` in die Datei `<dateiname>`.