Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Olaf Müller

WS 1998/99
Übungsblatt 8
21. 12. 1998

Übungen zur Semantik von Programmiersprachen

Aufgabe 1 **(H)** *(IMP ohne if-then-else)* **Abgabetermin: Dienstag, 19.1.99, 12 Uhr**

Zeigen Sie, daß man das *if-then-else*-Konstrukt auf die *while*-Schleife, die Zuweisung und die Komposition reduzieren kann.

- Sei $c :=$ **if** b **then** c_0 **else** c_1. Erstellen Sie dazu ein Programm c', das dieselbe Bedeutung besitzt wie c, aber nicht auf die Fallunterscheidung zurückgreift. Sie dürfen in c' eine neue Hilfsvariable einführen.
- Zeigen sie nun formal, daß die denotationale Semantik von c bezüglich der Variablen aus c mit derjenigen von c' übereinstimmt.

Aufgabe 2 **(H)** *(Auswertungssemantik und Typsicherheit)*

In der Vorlesung wurde eine Transitionssemantik für TIMP angegeben. Entwickeln Sie eine entsprechende Auswertungssemantik und beweisen Sie die Typsicherheit von TIMP bezüglich dieser Semantik.

Aufgabe 3 **(Ü)** (*Statische Modifikation von Typen*)

Das Typsystem der Sprache TIMP soll derartig erweitert werden, daß Typen von Variablen im Programmverlauf verändert werden können. Beispielsweise soll folgendes Programm als typkorrekt angesehen werden: $X := 5; X := true$.

- Entwickeln Sie Typisierungsregeln für ein solches Typsystem. Beachten Sie dabei, daß sich der Deklarationskontext Γ während des Programmverlaufs verändern kann. Deshalb sollen Typisierungsregeln die Form $\{\Gamma\}\mathit{konstrukt}\{\Gamma'\}$ besitzen mit der Bedeutung, daß *konstrukt* typkorrekt ist, falls Γ vor der Ausführung und Γ' nach der Ausführung gilt.
- Geben Sie eine Auswertungssemantik für die erweiterte Sprache an.
- Beweisen Sie die Typsicherheit.

Aufgabe 4 **(P)** (*Typchecking und Typinferenz*)
Abgabetermin: Dienstag, 19.1.1999, 12 Uhr; E-mail an mueller@informatik

In der Vorlesung wurde die um Typen erweiterte Sprache TIMP vorgestellt.

- Programmieren Sie den Typüberprüfungsalgorithmus dieser Sprache in der Programmiersprache Prolog. Verwenden Sie für die abstrakte Syntax ausschließlich Präfixkonstrukte analog zur Aufgabe 5 auf Blatt 2.
- Testen Sie Ihr Programm als Typinferenzverfahren, indem Sie für ein geeignetes Beispiel die Typen ausrechnen lassen.

Wir wünschen Ihnen ein frohes Fest und ein gutes neues Jahr!