

# Compilerbau II

## WS 2014/15

### Leitfaden zur Prüfungsvorbereitung

#### Attributierte Grammatiken

Was versteht man unter einer attributierten Grammatik? Stichworte: Attribut, Wertebereich, synthetische und geerbte Attribute, semantische Regeln. Gibt es immer eine Auswertungsreihenfolge für die Attribute? Wie bestimmt man die Auswertungsreihenfolge? Wozu verwendet man attributierte Grammatiken? Was versteht man unter S-attributierten bzw. L-attributierten Grammatiken? Wozu führt man sie ein? Wie sieht bei ihnen die Auswertungsreihenfolge aus? Welche Zusammenhänge bestehen zwischen S- und L-attributierten Grammatiken?

#### Typüberprüfung

Wie sieht das in der Vorlesung definierte Typsystem für die funktionale Sprache aus? Welche Arten von Typurteilen gibt es? Wie sehen die wichtigsten Typregeln aus? Wie erhält man daraus die Implementierung eines Typecheckers? Was hat das mit attributierten Grammatiken zu tun? Warum führt man überhaupt eine Typüberprüfung vor der eigentlichen Übersetzung durch?

#### Umgebungssemantik und Codegenerierung

Wie sind Umgebungen und closures definiert? Wie sehen die wichtigsten big step Regeln der Umgebungssemantik aus? Wie erhält man daraus eine—zunächst naive—Implementierung eines Interpreters? Wie kann man den Interpreter verbessern, um daraus letztendlich einen Codegenerator zu erhalten? Welchen Nettoeffekt sollte der generierte Code haben? Wie sieht (in groben Zügen) die Übersetzung aus? Was kann man an der Übersetzung noch verbessern, um effizienteren Code zu erhalten? Was versteht man unter Unter- bzw. Überversorgung von Funktionsaufrufen. Können sie bei wohlgetypten Programmen überhaupt auftreten?

#### Datenflussanalyse

Wozu dient die liveness-Analyse? Wie wird sie durchgeführt, d.h. wie sieht das Ungleichungssystem aus, das man lösen muss? Wie löst man es?

Wie sieht der allgemeine Ansatz für eine (Rückwärts-)Datenflussanalyse aus? Stichworte: Vollständiger Verband, monotone Funktionen, Kanteneffekte, zugehöriges (Un-)Gleichungssystem, kleinster (Prä-)Fixpunkt. Existiert er immer? Kann man ihn immer berechnen? Welche Verfahren gibt es dazu? Was versteht man unter der MOP-Lösung? Welcher Zusammenhang besteht zwischen Fixpunkt- und MOP-Lösung (Sätze von Kam/Ullman und Kildall)?

Welchen Nachteil hat die bisherige liveness-Analyse? Wie kann man sie noch verbessern?

Wozu dient die Verfügbarkeits-Analyse? Wie unterscheidet sich eine Vorwärts- von einer Rückwärtsanalyse? Wie sehen die Kanteneffekte der Verfügbarkeits-Analyse aus?