Hector Seminar

Bau einer Programmiersprache

2020

Modulausschreibung

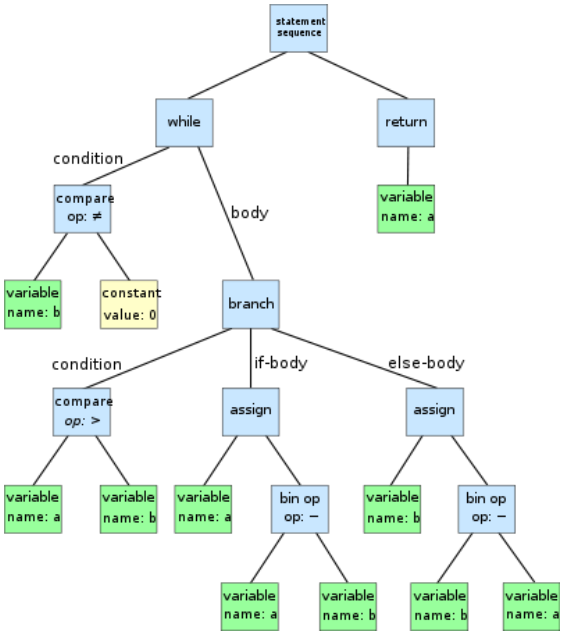
Zeitraum: A

Geübte Programmierer*Innen haben eine ziemlich genaue Vorstellung davon, was die verschiedenen Anweisungen der verwendeten Programmiersprache bedeuten. Diese wird häufig durch Übung erworben. Allerdings gibt es auch in den meisten Programmiersprachen komplizierte Aspekte, bei denen ein intuitives Verständnis der Programmiersprache nicht immer ausreicht (z.B. bei Programmen, die gleichzeitig mehrere Prozessoren nutzen, oder komplizierten Vererbungsstrukturen).
Die Art und Weise, in der man Anweisungen in einer Programmiersprache aufschreiben muss, wird als *Syntax* bezeichnet. Die Bedeutung der Anweisung bezeichnet man als *Semantik*. Zum Beispiel ist „x = y + z“ Syntax um die Semantik „Berechne die Summe der Werte der Variablen y und z und speichere diese in der Variablen x“ auszudrücken. Natürlich wäre für dieselbe Semantik auch anderer Syntax vorstellbar, beispielsweise „y + z → x“.

Ein präzises Verständnis der Semantik einer Programmiersprache ist notwendig, um eine neue Programmiersprache zu entwickeln. Der vom Programmierer geschriebene Quelltext, der in der Regel aus Anweisungen der Programmiersprache besteht, muss auf einem Computer ausgeführt werden. Häufig benutzt man Compiler, die den Quelltext in die Maschinensprache des verwendeten Rechners übersetzen. Dieser kann dann ausgeführt werden. Natürlich muss der Compiler genau die Semantik der Programmiersprache beachten, damit sich das Programm so verhält wie erwartet. Eine meistens einfachere Alternative ist ein Interpreter, der das Programm nicht in einem Stück in Maschinensprache übersetzt, sondern „interaktiv“ selbst ausführt. Von der Funktionsweise von Compilern oder Interpretern merkt man als Programmierer in der Regel nichts, sondern benutzt sie als Black-Box, deren Funktion als Magie erscheint.

Ziel des Moduls ist es zu verstehen wie man formal die Semantik einer Programmiersprache definieren und diese am Computer implementieren kann. Wir wollen dies am Beispiel einer einfachen „selbstgebauten“ Programmiersprache nachvollziehen. Schrittweise werden wir neue Konstrukte in unsere Programmiersprache aufnehmen, deren Semantik festlegen und diese in einem Interpreter auch programmieren. Mit diesem können wir Programme in unserer Programmiersprache dann ausführen. Dabei lernen wir die grundlegenden formalen Techniken und die Funktionsweise eines Interpreters kennen.

Wir beginnen mit der Darstellung des Programm als abstrakter Syntaxbaum (*abstract syntax tree, (AST)*). Ein AST ist eine baumartige Darstellung der syntaktischen Struktur des in der Programmiersprache geschriebenen Quelltextes, die sich gut zum Verarbeiten am Rechner eignet. Damit wollen wir verstehen, wie man (arithmetische bzw. boolsche) Ausdrücke rekursiv auswerten kann und wie man mit verschiedenen Datentypen zur Laufzeit umgeht. Als nächsten Schritt erweitern wir diese Ausdrücke um Variablen, die sowohl in arithmetischen bzw. boolschen Ausdrücken verwendet werden als auch deren Ergebnis speichern können. Damit können wir verschiedene Ausdrücke auch nacheinander auswerten und die Ergebnisse der vorherigen verwenden. Danach werden wir uns mit der Implementierung der üblichen Kontrollstrukturen (if, while, for) sowie Funktionsaufrufen und der Sichtbarkeit von Variablen beschäftigen.



Hector Seminar

Bau einer Programmiersprache

2020

Modulausschreibung

Zeitraum: A

Voraussetzungen:
Interesse am Programmieren und an Mathematik und deren Anwendung in der Informatik.
Es wird eine besonders hohe Bereitschaft erwartet, sich mit den Themen selbstständig auseinander zu setzen. Zwischen den Treffen müssen die besprochenen Inhalte intensiv nachbereitet werden. Eine regelmäßige Teilnahme an angegebenen Terminen ist notwendig.
Die gemeinsame Abschlusspräsentation findet am Samstag, 04. Juli 2020 am International Department in Karlsruhe statt.

Geplante Termine:

- **Do, 06.02.2020, 16^{00}-18^{15} Uhr**
- **Do, 13.02.2020, 16^{00}-18^{15} Uhr**
- **Do, 20.02.2020, 16^{00}-18^{15} Uhr**
- **Do, 05.03.2020, 16^{00}-18^{15} Uhr**
- **Do, 12.03.2020, 16^{00}-18^{15} Uhr**
- **Do, 19.03.2020, 16^{00}-18^{15} Uhr**
- **Do, 26.03.2020, 16^{00}-18^{15} Uhr**
- **Sa, 04.07.2020, Modulfest International Department KIT**

Teilnehmerzahl: höchstens 20

Ort: Bunsengymnasium,
Humboldtstr. 23, Heidelberg

Betreuer: Moritz Lichter, TU Kaiserslautern
Gisela Döbbeling, Kursleiterin Heidelberg, email: doebbeling@hector-seminar.de
Dr. Oliver Rudolph, Kursleiter Heidelberg, email: rudolph@hector-seminar.de