Bedeutung und Korrektheit von C Programmen

Vorlesung vom 07.04.2008: Einführung

Christoph Lüth & Lutz Schröder

SS 08

Organisatorisches

- Veranstalter:

  Christoph Lüth — Cartesium 2.043, Tel. 64223
  Lutz Schröder — Cartesium 2.051, Tel. 64216
  `{cxl,lschrode}@informatik.uni-bremen.de`
  `http://www.informatik.uni-bremen.de/~cxl,~lschrode`
- Termine:

  Vorlesung: Dienstag 10 – 12 MZH 4194
  Mittwoch 13 – 15 MZH 7210
- Website:
  `www.informatik.uni-bremen.de/~cxl/lehre/svc.ss08`
- Wichtig: Vorlesung am 09.04.2008 fällt aus!

Scheinkriterien & Übungsbetrieb

- Übungsbetrieb:
  - Regelmäßige Übungen
  - Vier Übungsblätter
    - Ggf. unterteilt in Abschnitte
- Scheinkriterien:
  - Abgabe aller Übungsblätter & ggf. mündliche Präsentation

Reasons to hate C

- Mangelnde Typsicherheit: `x == 2` vs. `x= 2`
- Mangelnde Datenabstraktion
- Verwirrende Typkonversionen: `printf("%d", sizeof('a'))`
- Felder vs. Zeiger
- Mangelnde Modularisierung (geschachtelte Funktionen, Module)
- Syntaktisches Rauschen: `auto`, `register`, `const`, `static`

Reasons to love C

- Einfach und schnell
- Standardisiert
- Weit verbreitet
- Sehr gute Werkzeugunterstützung

Lernziele

1. Die Sprache C — Was Sie schon immer über C wissen wollten ...
2. Bedeutung — Informell (Sprachspezifikation) und formal (mathematisch)
3. Korrektheit — Beweis von Programmeigenschaften

Gliederung

- Teil I: Die Sprache
  - Eine informelle Einführung in die Tiefen und Untiefen
- Teil II: Bedeutung
  - Formale Bedeutung des C-Standards
- Teil III: Korrektheit
  - Beweis von Programmeigenschaften mit dem Floyd-Hoare-Kalkül

Die Sprache C

- Der Standard (ISO IEC 9899:1999): Aufbau. Inhalt, Nomenklatur
- Das Typsystem: Deklaratoren, *specifier*, *qualifier* und *storage class*, vollständige und unvollständige Typen
- Speichermodell: Zeiger, Felder, mehrdimensionale Felder

Bedeutung

- Einführung in die denotationelle Semantik
- Bedeutung von Anweisungen
- Modellierung des Speichermodells
- Auswertung von Ausdrücken

Beweis von Programmeigenschaften

- Der Floyd-Hoare-Kalkül (Wdh)

$$\{P\}p\{Q\}$$

- Regeln für C, Korrektheit der Regeln
- Berechung von Verifikationsbedinungen

Übungen

- Implementierung eines Parsers
- Implementierung von statischen Analysen
- Korrektheitsbeweise & Verifikationsbedingungen (Papier & Bleistift)
- Programmiersprache nach Wahl
  - Aber: muss uns bekannt sein
  - Wir empfehlen Haskell