

Theoretische Informatik

Einführungsvorlesung

Matthias Heizmann

Software Engineering
Albert-Ludwigs-University Freiburg

17. Oktober 2018

- Motivation / Vorlesungsinhalt
- Organisation
- Erstes Kapitel: Formale Sprachen

Motivation

Richtig oder falsch?

“Computer können alles berechnen, wir brauchen nur genug Rechenleistung und genug Arbeitsspeicher.”



Foto: SuperMUC am Leibniz-Rechenzentrum

Richtig oder falsch?

“Manche Probleme kann man nur mit bestimmten Programmiersprachen lösen.”



Richtig oder falsch?

“Wenn man sich genug anstrengt kann man alle Computerprogramme verstehen.”

```
16 string sInput;  
17 int iLength, iN;  
18 double dblTemp;  
19 bool again = true;  
20  
21 while (again) {  
22     iN = -1;  
23     again = false;  
24     getline(cin, sInput);  
25     system("cls");  
26     stringstream(sInput) >> dblTemp;  
27     iLength = sInput.length();  
28     if (iLength < 4) {  
29         again = true;  
30         continue;  
31     } else if (sInput[iLength - 3] != '.') {  
32         again = true;  
33         continue;  
34     } while (++iN < iLength) {  
35         if (isdigit(sInput[iN])) {  
36             continue;  
37         } else if (iN == (iLength - 3)) {  
38             continue;  
39         }  
40     }  
41 }
```

- Abstrakte Betrachtung von Computern und Berechnungsverfahren

- Abstrakte Betrachtung von Computern und Berechnungsverfahren
- Betrachtung von Computern und Berechnungsverfahren mit mathematischen Formalismen

Warum abstrakte Betrachtung?

- Konkrete Hardware/Software sehr komplex

Warum abstrakte Betrachtung?

- Konkrete Hardware/Software sehr komplex
 - Bsp. CPU: Processor Programming Reference (PPR) for AMD Family 17h Model 01h, Revision B1 Processor hat ca. 250 Seiten.
 - Bsp. Programmiersprache: ISO/IEC 9899:2011 ("C11 Standard") hat ca. 700 Seiten.

Warum abstrakte Betrachtung?

- Konkrete Hardware/Software sehr komplex
 - Bsp. CPU: Processor Programming Reference (PPR) for AMD Family 17h Model 01h, Revision B1 Processor hat ca. 250 Seiten.
 - Bsp. Programmiersprache: ISO/IEC 9899:2011 ("C11 Standard") hat ca. 700 Seiten.
- Wachsende Vielfalt von Hardware und Programmiersprachen

Warum abstrakte Betrachtung?

- Konkrete Hardware/Software sehr komplex
 - Bsp. CPU: Processor Programming Reference (PPR) for AMD Family 17h Model 01h, Revision B1 Processor hat ca. 250 Seiten.
 - Bsp. Programmiersprache: ISO/IEC 9899:2011 ("C11 Standard") hat ca. 700 Seiten.
- Wachsende Vielfalt von Hardware und Programmiersprachen
 - 70er Jahre: Basic, Cobol, Fortran, PL/I, ...
 - Heute: C#, Haskell, Java, Python, ...
 - 2040: ???

Warum mathematische Formalismen?

Missverständnisfreie Kommunikation ermöglichen!

Warum mathematische Formalismen?

Missverständnisfreie Kommunikation ermöglichen!

Implementieren Sie einen Algorithmus der für zwei gegebene Zahlen den größten gemeinsamen Teiler berechnet.

Warum mathematische Formalismen?

Missverständnisfreie Kommunikation ermöglichen!

- 1 Es kann keinen Compiler geben, der immer herausfindet ob das Eingabeprogramm immer terminiert.
- 2 Für jedes $n \in \mathbb{N}$ gilt: Für jedes Computerprogramm, das höchstens n Byte Speicher benötigt, können wir entscheiden ob dieses terminiert.

Widerspruch?

- Was ist ein Programm?
- Was bedeutet "Terminierung"?
- Was ist eine Eingabe?
- Was bedeutet "Speicher benötigen"?
- Was bedeutet "immer herausfinden"?

1 Formale Sprachen und Automatentheorie

- 1 Formale Sprachen und Automatentheorie
“Rechnen mit Zeichenketten und Mengen von Zeichenketten”

- ① Formale Sprachen und Automatentheorie
“Rechnen mit Zeichenketten und Mengen von Zeichenketten”
 - Endliche Automaten
 - Reguläre Ausdrücke
 - Grammatiken
 - Automaten mit Stapelspeicher
 - Turingmaschinen

- ① Formale Sprachen und Automatentheorie
“Rechnen mit Zeichenketten und Mengen von Zeichenketten”
 - Endliche Automaten
 - Reguläre Ausdrücke
 - Grammatiken
 - Automaten mit Stapelspeicher
 - Turingmaschinen
- ② Berechenbarkeitstheorie
 - Nicht berechenbare Funktionen
 - Unentscheidbare Probleme
 - Reduktion

- ① Formale Sprachen und Automatentheorie
“Rechnen mit Zeichenketten und Mengen von Zeichenketten”
 - Endliche Automaten
 - Reguläre Ausdrücke
 - Grammatiken
 - Automaten mit Stapelspeicher
 - Turingmaschinen
- ② Berechenbarkeitstheorie
 - Nicht berechenbare Funktionen
 - Unentscheidbare Probleme
 - Reduktion
- ③ Komplexitätstheorie
 - Berechnungskomplexität auf Turingmaschinen
 - Komplexitätsklassen P und NP

Organisation

siehe Vorlesungswebsite