

Algorithmen

Lukas Abelt

`lukas.abelt@airbus.com`

DHBW Ravensburg
Wirtschaftsinformatik

Ravensburg
27. März 2019

Outline

- 1 Allgemeines
 - Begriffsklärung
 - Ziele des Moduls
- 2 Beschreibung
 - Formale Eigenschaften
 - Darstellungsformen
- 3 Analyse
 - Korrektheit eines Algorithmus
 - Komplexitätsanalyse

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Begriffklärung

Etymologie

- Leitet sich ursprünglich vom persischen Astronomen „Muhammad Ibn-Musa al-Hwarizmi“ ab
 - Schrieb Bücher über das indische Zahlensystem (um 800 n. Chr.)
 - Im 12. Jh übersetzt ins lateinische
 - Dabei wurde der Namensbestandteil „al-Hwarizmi“ in „Algorismi“ lateinisiert
- Durch spätere Überlieferungen wurde der Begriff später als Zusammensetzung betrachtet aus...
 - Dem Namen „Algus-“...
 - und dem aus dem griechisch entlehnten „-rismus“ (Zahl)

Begriffsklärung

Was bedeutet das jetzt

Formale Definition

Eine Berechnungsvorschrift zur Lösung eines Problems heißt genau dann Algorithmus, wenn eine zu dieser Berechnungsvorschrift äquivalente Turingmaschine existiert, die für jede Eingabe, die eine Lösung besitzt, stoppt.

Oder auch

Ein Algorithmus ist eine domänenunabhängige Beschreibung einer Handlungsvorschrift zur Lösung eines Problems. Eine bestimmte Eingabe wird in eine bestimmte Ausgabe überführt.

Begriffsklärung

Also

- Ist also die Beschreibung eines Programmes oder einer Funktion
 - Unabhängig von der verwendeten Programmiersprache!
 - Source Code direkt ist also kein Algorithmus...
 - ...aber aus diesem lässt sich der verwendete Algorithmus ableiten und beschreiben
- Algorithmen können in verschiedenen Formen dargestellt werden (Mehr dazu im nächsten Kapitel)

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Ziele

- ▣ Am Ende des Moduls könnt ihr...
 - ▣ Einen Algorithmus in eine Implementierung umsetzen
 - ▣ Aus einer Implementierung den Algorithmus ableiten
 - ▣ Die formalen Eigenschaften von Algorithmen kennen
 - ▣ Algorithmen anhand der kennengelernten Methoden zu analysieren

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Eigenschaften von Algorithmen

Grundlegendes

- **Finitheit** - Ein Algorithmus lässt sich in endlich vielen Schritten eindeutig beschreiben
- **Ausführbarkeit** - Jeder Einzelschritt muss tatsächlich ausführbar sein
- **Platzkomplexität** - Ein Algorithmus benötigt zu jedem Zeitpunkt nur endlich viel Speicherplatz
- **Terminierung** - Der Algorithmus benötigt eine endliche Anzahl von Schritten zur Ausführung
- **Determiniertheit** - Der Algorithmus muss bei gleichen Rahmenbedingungen das gleiche Ergebnis liefern
- **Determinismus** - Der nächste Schritt des Algorithmus ist zu jedem Zeitpunkt genau definiert

Korrektheit von Algorithmen

- Jeder Algorithmus sollte auch in allen Fällen das korrekte Ergebnis liefern...
- Klingt simpel, aber eindeutiger Beweis für alle Eingaben oft schwierig
- Testen an ausgewählten Beispielen **nicht** ausreichend
 - Jedoch verringern umfangreiche Tests natürlich das Risiko eines unentdeckten Fehler
- Korrektheit lässt sich im Grunde nur durch formalen Beweis zeigen
 - Diese sind häufig sehr umfangreich und komplex...
 - ...und deshalb auch nicht Teil der Vorlesung

Korrektheit von Algorithmen



Quelle:

„Program testing can be used to show the presence of bugs, but never to show their absence!“

Edsger W. Dijkstra

Effizienz von Algorithmen

- Ergibt sich indirekt aus den Grundlegenden Eigenschaften
- Effizienz lässt sich über verschiedene Größen beschreiben:
 - Speicherverbrauch
 - Zeitverbrauch
- Die sind jedoch oft Implementierungs- und Rechnerabhängig
- Deshalb wird mit formalisierten Modellen gearbeitet
- ...Mehr dazu im Kapitel „Analyse“

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Möglichkeiten der Darstellung

- Zur Definition von Algorithmen gibt es verschiedenste Möglichkeiten
- Mit ganz eigenen Vor- und Nachteilen
- Wir betrachten im Rahmen der Vorlesung:
 - Prosatext
 - Pseudocode
 - Struktogramme
 - Programmablaufplan (PAP)

Was beschreiben wir?

Unser Referenzalgorithmus

- Um die verschiedenen Elemente zu vergleichen, wollen wir mit allen den folgenden Algorithmus beschreiben:

Referenz

Für eine Zahl n (Wobei gilt: $n \in \mathbb{N}$), soll die Summe aller geraden Zahlen von 0 bis n berechnet werden.

Darstellung als Prosatext

Der simple Weg

- Simpleste Herangehensweise
- Man beschreibt in eigenen Worten, wie man vorgehen würde um die gegebene Problemstellung zu lösen
- **Achtung:** Unterscheiden zwischen Problemstellung und Lösungsbeschreibung!
- Auch in Prosaform sollten die Einzelschritte eindeutig beschrieben sein
- Nicht standardisiert → Beschreibung von Algorithmen inkonsistent

Prosabeschreibung

Für unseren Algorithmus

Addiere alle geraden Zahlen

Lese die Zahl n ein.

Anschließend setze die Zählvariable i sowie die Ergebnisvariable res auf 0.

Wenn i gerade ist, addiere i auf die Ergebnisvariable. Erhöhe anschließend i um 1. Wiederhole die letzten zwei Schritte bis i größer ist als n .

Gebe res aus

Darstellung als Pseudocode

Der Zwischenweg

- ❑ Mischung aus Prosa und tatsächlichem Code
- ❑ Orientiert sich an den in Programmiersprachen vorhandenen Strukturen (If-then-else, Schleifen...)
- ❑ Nutzt dabei aber leicht verständliche und programmiersprachenunabhängige Begriffe
- ❑ Wie Code in der Regel zeilenweise auf atomare Operationen beschränkt
- ❑ Keine formale Standardisierung, dadurch auch hier Inkonsistenzen möglich → Aber weniger als bei Prosabeschreibung

Pseudocode

Für unser Pseudoproblem

```
1  LESE  n
2  SETZE res=0
3  FÜR i=0 BIS n
4      WENN istGerade(i) DANN
5          res+=i
6      ENDE WENN
7  ENDE FÜR
8  GEBE res AUS
```

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Inhalt

1 Allgemeines

- Begriffsklärung
- Ziele des Moduls

2 Beschreibung

- Formale Eigenschaften
- Darstellungsformen

3 Analyse

- Korrektheit eines Algorithmus
- Komplexitätsanalyse

Kontakt

- E-Mail: `lukas.abelt@airbus.com`
- GitHub: `https://www.github.com/LuAbelt`
- GitLab: `https://www.gitlab.com/LuAbelt`
- Telefon(Firma): 07545 - 8 8895
- Telegram: LuAbelt