Einführung in die Programmierung

Teil 3: Daten und Algorithmen

Prof. Dr. Peer Kröger,
Florian Richter, Michael Fromm

Wintersemester 2018/2019

Übersicht

1. Datendarstellung durch Zeichenreihen
2. Syntaxdefinitionen
3. Eigenschaften von Algorithmen
4. Paradigmen der Algorithmenentwicklung

Roadmap

- Wir nehmen unsere Diskussion über den Algorithmus-Begriff wieder auf und konkretisieren einige wichtige Aspekte.
- Wir beschäftigen uns zunächst mit dem Aspekt der (eindeutigen) Darstellung der zu verarbeitenden Daten
- Anschließend diskutieren wir anhand einem Beispiel wesentliche Eigenschaften von Algorithmen und unterschiedliche Herangensweisen in der Algorithmenentwicklung.

Daten

- Wir betrachten zunächst die Daten (Objekte), die durch Algorithmen verarbeitet werden sollen.
- Wir wollen insbesondere klären, wie diese Daten dargestellt werden.
- Typische Daten sind Zahlen, z.B. die Zahl “drei”, die wie folgt dargestellt werden kann:
  - 3
  - DREI
  - III
  - drei ausgestreckte Finger einer Hand
  - ...

Datendarstellung

- Wir unterscheiden bei einem Objekt
  - die Darstellung, (*Syntax*, “Bezeichnung”),
  - seine Bedeutung, (*Semantik*, “Information”).
- Man kann die Syntax (von etwas) definieren, ohne die entsprechende Semantik zu kennen.
- Die Syntax von Daten (genauso: Programmiersprachen) muss formal (und eindeutig) definiert sein (z.B. mit Hilfe von „Grammatiken“).
- Wir führen hier allerdings eher eine informelle Diskussion und verweisen auf andere Vorlesungen.

Datendarstellung

- Einige der oben genannten Datendarstellungen für die Zahl „drei“ sind für maschinelle Verarbeitung nicht geeignet.
- Alle geeigneten Datendarstellungen beruhen auf dem Grundprinzip der *Zeichenreihe*, die wir im folgenden formal definieren.
- Dazu benötigen wir zunächst eine Menge an „Zeichen“, die zur Bildung von Zeichenreihen zur Verfügung steht.

Definition (Alphabet)

Ein Alphabet $\mathcal{A}$ ist eine endliche Menge, deren Elemente Zeichen genannt werden. Alphabete, die genau zwei Zeichen enthalten heißen binär.

Beispiele

- Menge der Großbuchstaben: {A,B,C, ...,Z}
- Menge der Dezimalziffern: {0,1,2,3, ...,9}
- Menge der Vorzeichen: $\{+, -\}$
- Menge der Richtungszeiger eines Lifts: $\{\uparrow, \downarrow\}$
- Ein wichtiges binäres Alphabet besteht aus den *Binärziffern* (*Bits*) $\{0, 1\}$.

Definition (Zeichenreihe)

Eine Zeichenreihe über einem Alphabet $\mathcal{A}$ ist eine (endliche) Folge von Zeichen aus $\mathcal{A}$.

- Formal ist damit auch die leere Folge eine Zeichenreihe.
- Wir schreiben Zeichenreihen/Folgen $(x_1, x_2, \ldots, x_n)$ auch als $x_1 x_2 \ldots x_n$.

Beispiele

- Sei $\mathcal{A}_1 = \{$A,B,C,...,Z$\}$
  - Die Folge INFORMATIK ist eine Zeichenreihe über $\mathcal{A}_1$.
  - Die Folge (R,I,C,H,T,E,R) ist eine Zeichenreihe über $\mathcal{A}_1$.
  - Die Folge Kröger ist *keine* Zeichenreihe über $\mathcal{A}_1$.
    Warum?
- Sei $\mathcal{A}_2 = \{0, 1\}$
  - Die Folge 0 ist eine Zeichenreihe über $\mathcal{A}_2$.
  - Die Folge 1 ist eine Zeichenreihe über $\mathcal{A}_2$.
  - Die Folge 01 ist eine Zeichenreihe über $\mathcal{A}_2$.
  - Die Folge 10 ist eine Zeichenreihe über $\mathcal{A}_2$.
  - Die Folge 11 ist eine Zeichenreihe über $\mathcal{A}_2$.
  - Die Folge 00 ist eine Zeichenreihe über $\mathcal{A}_2$.
  - ...

Bezeichnung von Daten

- Wir verwenden ausschließlich Zeichenreihen zur Bezeichnung von Daten.
- Im Folgenden betrachten wir als Beispiel die Darstellung von natürlichen Zahlen (Elementen der Menge $\mathbb{N}_0$).
- Die Zahl “Dreizehn” lässt sich u.a. durch folgende Zeichenreihen bezeichnen:

13	$(\mathcal{A} = \{0, 1, 2, \ldots, 9\})$,
DREIZEHN	$(\mathcal{A} = \{\text{A,B,\ldots,Z}\})$,
|||||||||||||	$(\mathcal{A} = \{\|\})$.

- Nicht alle diese Darstellungen sind für den praktischen Gebrauch (z.B. Rechnen) geeignet.

Zifferndarstellung / p-adische Zahlendarstellung

- Am besten für die Darstellung von natürlichen Zahlen eignet sich die *Zifferndarstellung* über einem vorgegebenen Alphabet von Ziffern, d.h. die Zeichen des zugrunde liegenden Alphabets sind Ziffern (oder werden als Ziffern interpretiert).
- Beispiel für eine Zifferndarstellung ist die allgemein gebräuchliche *Dezimaldarstellung* über dem Alphabet $\{0, 1, 2, \ldots, 9\}$.
- Die folgende Definition der *p-adischen Zahlendarstellung* führt allgemein die Zifferndarstellung über ein beliebiges Alphabet von Ziffern ein.

Definition (*p*-adische Zahlendarstellung)

Sei $p \in \mathbb{N}$, $p \geq 2$ *und* $\mathcal{A}_p = \{z_0, z_1 \ldots, z_{p-1}\}$ *ein Alphabet mit p Zeichen (genannt Ziffern)* $z_0, z_1 \ldots, z_{p-1}$.

Die Funktion $Z : \mathcal{A}_p \to \mathbb{N}_0$ *bildet jedes Zeichen aus* $\mathcal{A}_p$ *auf eine natürliche Zahl wie folgt ab:*

$$Z(z_i) = i \qquad \text{für } i = 0, \ldots, p-1.$$

Eine Zeichenreihe $x = x_n x_{n-1} \ldots x_1 x_0$ *(der Länge* $(n+1)$*) über* $\mathcal{A}_p$ *(d.h.* $x_i \in \mathcal{A}_p$ *für* $0 \leq i \leq n$*) bezeichnet die Zahl*

$$\mathcal{Z}(x) = p^n \cdot Z(x_n) + p^{n-1} \cdot Z(x_{n-1}) + \ldots + p \cdot Z(x_1) + Z(x_0).$$

Zur Verdeutlichung schreiben wir auch x_p *statt* x. x_p *heißt p-adische Zahlendarstellung der Zahl* $\mathcal{Z}(x_p) \in \mathbb{N}_0$.

Nochmal: $\mathcal{Z}(x) = p^n \cdot Z(x_n) + p^{n-1} \cdot Z(x_{n-1}) + \ldots + p \cdot Z(x_1) + Z(x_0)$.

Beispiel

- Mit $p = 10$ und $\mathcal{A}_{10} = \{z_0, z_1 \ldots, z_9\}$ erhält man die Dezimaldarstellung wenn man statt den Ziffern z_i des Alphabets gleich die Zahl $Z(z_i)$ schreibt (also z.B. statt z_3 schreibe $Z(z_3) = 3$ bzw. statt $z_9\, z_8\, z_3$ direkt 983):

  $\mathcal{Z}(983_{10}) = 10^2 \cdot 9 + 10^1 \cdot 8 + 10^0 \cdot 3 =$ "neunhundertdreiundachtzig".

  (Wir schreiben direkt $\mathcal{A}_{10} = \{0, 1 \ldots, 9\}$ anstelle von $\{z_0, z_1 \ldots, z_9\}$)

Beispiele

- $p = 2$ und $\mathcal{A}_2 = \{0, 1\}$ (*Binärdarstellung*):
  $\mathcal{Z}(1111010111_2) =$
  $2^9 \cdot 1 + 2^8 \cdot 1 + 2^7 \cdot 1 + 2^6 \cdot 1 + 2^5 \cdot 0 + 2^4 \cdot 1 + 2^3 \cdot 0 + 2^2 \cdot 1 + 2 \cdot 1 + 1 =$
  "neunhundertdreiundachtzig".
- $p = 8$ und $\mathcal{A}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ (*Oktaldarstellung*):
  $\mathcal{Z}(1727_8) = 8^3 \cdot 1 + 8^2 \cdot 7 + 8 \cdot 2 + 7 =$
  "neunhundertdreiundachtzig".
- $p = 16$ und $\mathcal{A}_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
  (*Hexadezimaldarstellung*):
  $\mathcal{Z}(3D7_{16}) = 16^2 \cdot 3 + 16 \cdot 13 + 7 =$
  "neunhundertdreiundachtzig".

Zifferndarstellung / *p*-adische Zahlendarstellung

- Führende Nullen sind in der Definition der *p*-adischen Zahldarstellung zugelassen, z.B. ist die Zeichenreihe

  000983

  eine zulässige Dezimaldarstellung und bezeichnet die gleiche Zahl wie die Zeichenreihe 983.
- Offensichtlich können führende Nullen immer weggelassen werden, außer bei der Bezeichnung “0” für die Zahl “null”.
- Für jede Zahl aus $\mathbb{N}_0$ gibt es für beliebiges $p \geq 2$ eine *p*-adische Darstellung.
- Betrachtet man nur Darstellungen ohne führende Nullen, so ist (zu festem $p \geq 2$) die *p*-adische Zahldarstellung eindeutig.

Menschengerechte Darstellung von Daten

- Zur Entwicklung von Algorithmen (z.B. mittels Programmiersprachen) werden wir typischerweise die Dezimaldarstellung der natürlichen Zahlen verwenden.
- Allgemein gibt es für die meisten Daten eine *Standarddarstellung* bei denen die Lesbarkeit der Darstellung für den menschlichen Benutzer im Vordergrund steht.
- Eine Maschinen-nahe Darstellung ist die Binärdarstellung, aber auch die Oktal- und Hexadezimaldarstellung, die für uns Menschen meist etwas ungewöhnlicher sind.

- Wir verwenden hier folgende Standardbezeichnungen wie sie in den üblichen höheren Programmiersprachen gebräuchlich sind:

Natürliche Zahlen	$\mathbb{N}_0$	Dezimaldarstellung (ohne führende Nullen)
Ganze Zahlen	$\mathbb{Z}$	wie natürliche Zahlen, ggf. mit Vorzeichen "-", z.B. 3,-3.
Reelle Zahlen	$\mathbb{R}$	*Gleitpunktdarstellung*, siehe später.
Wahrheitswerte	$\mathbb{B}$	*TRUE* und *FALSE* für "wahr" bzw. "falsch".

Darstellung von Zeichen und Texten

- Eine Dezimaldarstellung (z.B. 983) stellt eine natürliche Zahl dar, die verarbeitet werden kann.
- Die Zeichenreihe selbst (und nicht die dargestellte Zahl) könnte aber auch Gegenstand der Verarbeitung sein.
- Zeichenreihen können also nicht nur Darstellungen von Objekten sein, sondern auch selbst Objekte, die dargestellt werden müssen.
- Zeichenreihen heißen in diesem Zusammenhang *Texte*.

- In der Praxis wird zur Bildung von Texten häufig das sog. *ASCII-Alphabet* benutzt.
- Das ASCII-Alphabet repräsentiert eine Menge von Zeichen, die wir im folgenden als *CHAR* bezeichnen.
- Die Elemente von *CHAR* finden Sie in allen gängigen Lehrbüchern.

Maschinengerechte Darstellung von Daten

- Wie bereits erwähnt wird auf Rechenanlagen meist eine andere Darstellung der Daten gewählt (typischerweise Zeichenreihen über den oben benannten Alphabeten $\mathcal{A}_2$, $\mathcal{A}_8$ und $\mathcal{A}_{16}$).
- Aus technischen Gründen kann die kleinste Speichereinheit eines Computers (das *Bit*) nur zwei Zustände speichern:
  - Zustand 1: es liegt (elektr.) Spannung an.
  - Zustand 0: es liegt keine Spannung an.
- Daher werden Werte (Daten/Objekte) als Bitmuster (Zeichenreihe über dem Alphabet $\mathcal{A}_2 = \{0, 1\}$) codiert gespeichert (auch Darstellungen über $\mathcal{A}_8$ und $\mathcal{A}_{16}$ sind mit diesen technischen Gegebenheiten gut vereinbar).

Maschinengerechte Darstellung von Daten

- Intern kann der Computer also z.B. die natürlichen Zahlen in Binärcodierung repräsentieren.
- Ganze Zahlen können intern ebenfalls leicht als Zeichenkette über dem Alphabet $\mathcal{A}_2 = \{0, 1\}$ codiert werden (z.B. mit einem führenden Bit für das Vorzeichen; Genaueres darüber werden Sie in der Vorlesung “Rechnerarchitektur” lernen).
- Zeichen aus dem ASCII-Alphabet werden intern meist mit einer natürlichen Zahl repräsentiert, also auch hier gibt es eine binäre Darstellung.

- Die Binärdarstellung der reellen Zahlen ist etwas komplizierter.
- Die *Gleitpunktdarstellung* einer Zahl z ist

$$z = m \cdot 2^e,$$

  wobei sowohl m (*Mantisse*) als auch e (*Exponent*) binär repräsentiert werden (können).
- In vielen Programmiersprachen (z.B. Java) wird eine reelle Zahl dargestellt durch $z = m \cdot 10^e$, z.B. 3.14, -7.45, $1.33E - 2$ (für $1.33 \cdot 10^{-2}$).

Maschinengerechte Darstellung von Daten

- Eine genaue Spezifikation lernen wir im nächsten Abschnitt kennen.
- Achtung: Für viele reelle Zahlen gibt es gar keine derartige Darstellung (z.B. für $\sqrt{2}$). Die darstellbaren Zahlen heißen auch *Gleitpunktzahlen* und sind eine (echte) Teilmenge von $\mathbb{R}$.
- Dieser Aspekt der maschinengerechten Darstellung von Daten ist bei der Entwicklung von Algorithmen möglicherweise wichtig! Warum?

Maschinengerechte Darstellung von Daten

- Ein weiterer Aspekt der maschinengerechten Darstellung ist, dass die Ressourcen (Speicherzellen) einer Rechenanlage begrenzt sind.
- Es stehen daher für die Darstellung von Daten immer nur endlich viele Bits zur Verfügung.
- D.h. zur Darstellung einer beliebigen natürlichen Zahl stehen nur Zeichenketten mit fixer *Länge* zur Verfügung (notfalls mit führenden Nullen).

- Achtung:
  Werte, deren Darstellung mehr Zeichen (Bits) benötigen würden, können somit nicht dargestellt werden.
- Die Länge der zur Verfügung stehenden Zeichenketten hat damit offenbar Einfluss auf den Wertebereich der darstellbaren Objekte.
- Dies ist bei der Entwicklung von Algorithmen möglicherweise ebenfalls wichtig! Warum?

Maschinengerechte Darstellung von Daten

- Die Elemente des ASCII-Alphabets (zur Darstellung von Zeichen und Texten) werden intern durch natürliche Zahlen codiert.
- Diese Zahlen werden im Rechner entsprechend binär dargestellt (urspr. als 7-Bit Zeichenreihe, später als 8-Bit z.B. als UTF-8).
- Beispielsweise ist dem Symbol “a” (Element des ASCII-Alphabets) der Wert 97_{10} bzw. 01100001_2 zugeordnet, dem Symbol “4” (als Zeichen, d.h. Element des ASCII-Alphabets) der Wert 52_{10} bzw. 00110100_2.
- Achtung: Einheitliche Zeichencodierung ist eines der “größten ungelösten” Problem der Informatik ...

Zusammenfassung

- Daten werden typischerweise durch Zeichenreihen dargestellt.
- Die *p*-adische Darstellung definiert die Darstellung von natürlichen Zahlen über ein Alphabet mit *p* Zeichen.
- Die meisten Programmiersprachen verwenden die *menschengerechte* Dezimaldarstellung ($p = 10$) als Standard-Darstellung.
- Auf Basis der *p*-adischen Darstellung lassen sich auch ganze Zahlen und reelle Zahlen (Gleitpunktdarstellung) darstellen.
- Zeichen als Daten werden typw. über das ASCII Alphabet definiert/durch dessen Elemente repräsentiert.

Zusammenfassung

- Die *maschinengerechte* Darstellung von natürlichen Zahlen ist die Binärdarstellung ($p = 2$).
- Auf Basis der Binärdarstellung lassen sich auch ganze Zahlen und reelle Zahlen darstellen.
- Die Binärdarstellung von reellen Zahlen ist allerdings "unvollständig".
- Außerdem ist zu beachten, dass für die Zeichenreihen, die Zahlen repräsentieren, nur eine begrenzte (meist vorher definierte) Anzahl von Zeichen (Bits) zur Verfügung steht.
- Die Elemente des ASCII-Alphabets werden intern wiederum binär codiert (z.B. UTF-8).