Algorithmische Methoden 1

Wolfgang Windsteiger

RISC Institut

Das Thema der heutigen Vorlesung

Zahlen am Computer

Tupel

$$A^n := \underbrace{A \times \cdots \times A}_{n \text{ mal}} := \{\underbrace{(a_1, \ldots, a_n)}_{n\text{-Tupel}} \mid a_i \in A \text{ für } 1 \leq i \leq n\},$$

Grundoperationen für $t \in A^n$:

Länge von t: Anzahl der Komponenten, $|t|$.

Einzel-Zugriff: Zugriff auf einzelne Komponenten, t_i.

Mehrfach-Zugriff: Zugriff auf mehrere Komponenten, $t_{i:j} = (t_i, \ldots, t_j)$.

Leeres Tupel: $()$.

Realisierung am Computer: in beinahe jeder Programmiersprache einfach.

Basisdarstellung

Problemstellung (Basisdarstellung).

$$\begin{aligned} \text{Gegeben:} &\quad a, B \in \mathbb{N} \\ \text{mit:} &\quad B \geq 2. \\ \text{Gesucht:} &\quad L \in \mathbb{N},\ z \in \{0, \ldots, B-1\}^L \\ \text{mit:} &\quad a = \sum_{i=1}^{L} z_i B^{L-i} \text{ und } z_1 \neq 0. \end{aligned}$$

Vom Beweis zum Algorithmus

Satz (Basisdarstellung natürlicher Zahlen)

Sei $B \in \mathbb{N}$ mit $B \geq 2$ und $a \in \mathbb{N}$. Dann existieren eindeutig bestimmte $L \in \mathbb{N}$ und $z_i \in \{0, \ldots, B-1\}$ für alle $1 \leq i \leq L$, sodass

$$a = \sum_{i=1}^{L} z_i B^{L-i} \tag{26}$$

mit $z_1 \neq 0$. Man spricht dabei von der B-adischen Darstellung von a.

Beweis mit Induktion nach a. Konstruktiv.

Vom Beweis zum Algorithmus

Algorithmus *BasisdarstellungN*: Basisdarstellung in ℕ

$q \leftarrow a \operatorname{div} B$, $r \leftarrow a \bmod B$
if $q = 0$
  $L \leftarrow 1$, $z \leftarrow (r)$
else
  $(\eta, \zeta) \leftarrow \mathit{BasisdarstellungN}(q, B)$
  $z \leftarrow \mathit{EinfügenEnde}(\zeta, r)$
  $L \leftarrow \eta + 1$
return (L, z)

Aufruf: $\mathit{BasisdarstellungN}(a, B)$
Eingabe: $a, B \in \mathbb{N}$
mit: $B \geq 2$
Ausgabe: $L \in \mathbb{N}$,
$z \in \{0, \ldots, B-1\}^L$
mit: $a = \sum_{i=1}^{L} z_i B^{L-i}$ und
$z_1 \neq 0$.