Einführung in die Computerlinguistik – Berechenbarkeit, Entscheidbarkeit, Halteproblem

Dozentin: Wiebke Petersen

14.1.2009

Hinweis zu den Folien

Der Text dieser Folien ist größtenteils dem Kapitel “Eine Reise zum Rand der Welt” der Website `http://www.gk-informatik.de/` von Roland Mechling entnommen.

Was genau ist ein Algorithmus

Definition 1

Ein Algorithmus stellt die Berechnung einer partiellen Funktion $f : D_f \rightarrow \mathbf{N}$ dar, wobei $Df \in \mathbf{N}$ die Definitionsmenge von f ist. Dabei gilt für jedes $n \in \mathbf{N}$:

- wenn $n \in D_f$, dann liefert der Algorithmus $f(n)$;
- wenn $n \notin D_f$, dann terminiert der Algorithmus nicht.

Was genau ist ein Algorithmus

Definition 1

Ein Algorithmus stellt die Berechnung einer partiellen Funktion $f : D_f \rightarrow \mathbf{N}$ dar, wobei $Df \in \mathbf{N}$ die Definitionsmenge von f ist. Dabei gilt für jedes $n \in \mathbf{N}$:

- wenn $n \in D_f$, dann liefert der Algorithmus $f(n)$;
- wenn $n \notin D_f$, dann terminiert der Algorithmus nicht.

Zu jedem Algorithmus gibt es eine "passende" partielle Funktion!

Was genau ist ein Algorithmus

Definition 1

Ein Algorithmus stellt die Berechnung einer partiellen Funktion $f : D_f \rightarrow \mathbf{N}$ dar, wobei $Df \in \mathbf{N}$ die Definitionsmenge von f ist. Dabei gilt für jedes $n \in \mathbf{N}$:

- wenn $n \in D_f$, dann liefert der Algorithmus $f(n)$;
- wenn $n \notin D_f$, dann terminiert der Algorithmus nicht.

Zu jedem Algorithmus gibt es eine "passende" partielle Funktion!
Aber: ist auch jede partielle Funktion berechenbar?

Wieviele Algorithmen gibt es?

- Jeder Algorithmus kann in einem Programm realisiert werden.

Wieviele Algorithmen gibt es?

- Jeder Algorithmus kann in einem Programm realisiert werden.
- Eine Programmiersprache heißt “universell”, wenn sich mit ihr alle (aus)denkbaren Algorithmen in entsprechende Programme umsetzen lassen.

Wieviele Algorithmen gibt es?

- Jeder Algorithmus kann in einem Programm realisiert werden.
- Eine Programmiersprache heißt “universell”, wenn sich mit ihr alle (aus)denkbaren Algorithmen in entsprechende Programme umsetzen lassen.
- Der Quelltext eines jeden Programms entspricht einer binären Zahl.

Wieviele Algorithmen gibt es?

- Jeder Algorithmus kann in einem Programm realisiert werden.
- Eine Programmiersprache heißt “universell”, wenn sich mit ihr alle (aus)denkbaren Algorithmen in entsprechende Programme umsetzen lassen.
- Der Quelltext eines jeden Programms entspricht einer binären Zahl.
- $\Rightarrow$ es kann höchstens so viele Algorithmen wie natürliche Zahlen geben.

Wieviele Algorithmen gibt es?

- Jeder Algorithmus kann in einem Programm realisiert werden.
- Eine Programmiersprache heißt "universell", wenn sich mit ihr alle (aus)denkbaren Algorithmen in entsprechende Programme umsetzen lassen.
- Der Quelltext eines jeden Programms entspricht einer binären Zahl.
- $\Rightarrow$ es kann höchstens so viele Algorithmen wie natürliche Zahlen geben.

Theorem 2

Es kann höchstens abzählbar viele Algorithmen geben.

Wieviele partielle Funktionen gibt es?

- Zu jeder Teilmenge $T \subseteq \mathbf{N}$ der natürlichen Zahlen gibt es eine (partielle) Funktion χ (charakteristische Funktion):

$$\chi(n) = \begin{cases} 1 & \text{wenn } n \in T \\ 0 & \text{wenn } n \notin T \end{cases}$$

Wieviele partielle Funktionen gibt es?

- Zu jeder Teilmenge $T \subseteq \mathbf{N}$ der natürlichen Zahlen gibt es eine (partielle) Funktion χ (charakteristische Funktion):

$$\chi(n) = \begin{cases} 1 & \text{wenn } n \in T \\ 0 & \text{wenn } n \notin T \end{cases}$$

- Es gibt überabzählbar viele Teilmengen der natürlichen Zahlen (Tafelexkurs).

Wieviele partielle Funktionen gibt es?

- Zu jeder Teilmenge $T \subseteq \mathbf{N}$ der natürlichen Zahlen gibt es eine (partielle) Funktion χ (charakteristische Funktion):

$$\chi(n) = \begin{cases} 1 & \text{wenn } n \in T \\ 0 & \text{wenn } n \notin T \end{cases}$$

- Es gibt überabzählbar viele Teilmengen der natürlichen Zahlen (Tafelexkurs).

Theorem 3

Nicht alle partiellen Funktionen sind berechenbar!

Gibt es einen Halte-Tester?

Definition 4

Unter einem universellen Halte-Tester versteht man einen Algorithmus, der zu jedem beliebigen Programm *XYZ* und jedem beliebigen Datensatz *DAT* nach endlich vielen Schritten entscheidet, ob *XYZ* bei Eingabe von *DAT* anhält oder nicht.

Gibt es einen Halte-Tester?

Definition 4

Unter einem universellen Halte-Tester versteht man einen Algorithmus, der zu jedem beliebigen Programm *XYZ* und jedem beliebigen Datensatz *DAT* nach endlich vielen Schritten entscheidet, ob *XYZ* bei Eingabe von *DAT* anhält oder nicht.

Theorem 5

Das Halte-Problem ist nicht entscheidbar, d.h. es gibt keinen Algorithmus, der nach endlich vielen Schritten entscheiden kann, ob ein beliebiger anderer Algorithmus für beliebige gegebenene Eingabedaten halten wird oder nicht.

Beweis der Nichtexistenz eines Halte-Testers

- Angenommen es gebe ein solches Programm `Halte-Tester`.

Beweis der Nichtexistenz eines Halte-Testers

- Angenommen es gebe ein solches Programm `Halte-Tester`.
- `Meta` sei ein Programm, dass ein Programm `P` als Input nimmt und
  - terminiert, wenn `Halte-Tester(P,P)=nein`
  - und sonst in eine Endlosschleife gerät.

Beweis der Nichtexistenz eines Halte-Testers

- Angenommen es gebe ein solches Programm `Halte-Tester`.
- `Meta` sei ein Programm, dass ein Programm `P` als Input nimmt und
  - terminiert, wenn `Halte-Tester(P,P)=nein`
  - und sonst in eine Endlosschleife gerät.
- Frage: Terminiert `Meta(Meta)`?

Beweis der Nichtexistenz eines Halte-Testers

- Angenommen es gebe ein solches Programm `Halte-Tester`.
- `Meta` sei ein Programm, dass ein Programm `P` als Input nimmt und
  - terminiert, wenn `Halte-Tester(P,P)=nein`
  - und sonst in eine Endlosschleife gerät.
- Frage: Terminiert `Meta(Meta)`?
- Die Frage, ob `Meta(Meta)` terminiert führt zu einem Widerspruch.

Beweis der Nichtexistenz eines Halte-Testers

- Angenommen es gebe ein solches Programm `Halte-Tester`.
- `Meta` sei ein Programm, dass ein Programm `P` als Input nimmt und
  - terminiert, wenn `Halte-Tester(P,P)=nein`
  - und sonst in eine Endlosschleife gerät.
- Frage: Terminiert `Meta(Meta)`?
- Die Frage, ob `Meta(Meta)` terminiert führt zu einem Widerspruch.
- $\Rightarrow$ das Programm `Halte-Tester` kann nicht existieren.