Lernzettel

Halteproblem, Unentscheidbarkeit und Beweistechniken

Universität: Technische Universität Berlin
Kurs/Modul: Theoretische Grundlagen der Informatik
Erstellungsdatum: September 19, 2025

Lernzettel: Halteproblem, Unentscheidbarkeit und Beweistechniken

(1) Halteproblem. Das Halteproblem ist definiert als

$$H = \{\, \langle M, w \rangle \mid M \text{ ist eine Turingmaschine},\ w \in \Sigma^*,\ M \text{ hält auf } w \,\}.$$

Hier bedeutet „hält auf w“, dass M auf der Eingabe w in einer endlichen Berechnung terminiert.

(2) Unentscheidbarkeit und Entscheidbarkeit. Ein Problem L ist entscheidbar, wenn es eine Turingmaschine C gibt, die für jede Eingabe x korrekt entscheidet, d. h. C(x)=Ja oder Nein liefert. Ein Problem, bei dem dies nicht möglich ist, heißt unentscheidbar. Es gibt außerdem sprachen, die rekursiv aufzählbar, aber nicht entscheidbar sind (semi-entscheidbar), z. B. A_{TM} .

(3) Beweistechniken in der Informatik. Beweise verwenden oft folgende Techniken:

- Beweis durch Widerspruch (Reduktio ad absurdum)
- Beweis durch Konstruktion (direkter Beweis)
- Reduktion von A nach B (Many-one-Reduktion)
- Diagonalisierung
- Beweis durch Simulation/Konstruktion von Maschinen zur Reduktion

(4) Halteproblem – Beweis der Unentscheidbarkeit (Skizze). Angenommen, es gäbe eine Entscheidungsmaschine H, die für alle $\langle M, w \rangle$ entscheidet, ob M auf w hält.

Konstruktion einer Maschine D wie folgt:

$$\mathrm{D}(y): \quad \begin{cases} \text{Simuliere } H \text{ auf } \langle D, y \rangle, \\ \text{Falls } H(\langle D, y \rangle) = \text{„hält“},\ \text{dann } D \text{ läuft eine Endlosschleife;} \\ \text{Falls } H(\langle D, y \rangle) = \text{„hält nicht“},\ \text{dann } D \text{ hält.} \end{cases}$$

Betrachte D_D. Wenn H$\langle D, D \rangle$ =„hält“, dann D(D) muss enden, aber nach der Definition von D würde D(D) durch die Schleife weiterlaufen. Umgekehrt, wenn H$\langle D, D \rangle$ =„hält nicht“, würde D(D) halten, was wiederum gegen die Bedingung von D verstößt. Wiederspruch. Daher existiert kein Entscheidungsverfahren für HALT.

(5) Reduktion von $\mathbf{A}_{\mathrm{TM}}$ auf HALT (Beweisidee). Ziel: Zeigen, dass HALT unentscheidbar ist, indem man A_{TM} auf HALT reduziert.

Gegeben sei eine Eingabe $\langle M, w \rangle$ für A_{TM}, also ob M w akzeptiert. Konstruiere eine TM M' aus M und w wie folgt, die unabhängig von ihrer Eingabe x arbeitet:

$$\mathrm{M'}(x): \quad \text{Simuliere } M \text{ auf } w. \begin{cases} \text{Falls } M \text{ akzeptiert } w,\ \text{dann halte;} \\ \text{ansonsten läuft M' weiter.} \end{cases}$$

Dann gilt: HALT entscheidet $\langle M', x \rangle$ genau dann, wenn M w akzeptiert. Falls HALT existierte, könnte man A_{TM} entscheiden, was bekanntlich unmöglich ist. Daher HALT unentscheidbar.

(6) Beweise und Diagonalisierung. Diagonalisierungsargumente liefern klassische Beweise für Unentscheidbarkeit durch Konstruktion eines Widerspruchs gegenüber einer hypothetisch vollständigen Aufzählung oder Entscheidungsprozedur. Typische Anwendungen finden sich in HALT, A_{TM} und verwandten Problemen.

(7) Wichtige Folgerungen. - HALT ist unentscheidbar. - A_{TM} ist unentscheidbar. - Aus Unentscheidbarkeit von A_{TM} folgt oft Unentscheidbarkeit weiterer verwandter Probleme durch Reduktion. - Beweistechniken wie Reduktion, Diagonalisierung und konstruktive Maschinenbildung sind zentrale Werkzeuge der theoretischen Informatik.

(8) Übungs-ähnliche Hinweise. - Formuliere Reduktionsregeln sauber als Abbildungen $\leq_m$. - Verwende klare Maschinenbeschreibungen (Pseudo-Code ist erlaubt, solange er gut lesbar bleibt). - Achte darauf, dass alle Beweise logisch konsistent und Schritt-für-Schritt nachvollziehbar sind.