

Unentscheidbarkeit des Halteproblems: Unterprogrammtechnik

Prof. Dr. Berthold Vöcking
Lehrstuhl Informatik 1
Algorithmen und Komplexität
RWTH Aachen

Oktober 2011

Die Diagonalsprache:

$$D = \{ w \in \{0, 1\}^* \mid w = w_i \text{ und } M_i \text{ akzeptiert } w \text{ nicht} \} .$$

Satz:

Die Diagonalsprache D ist nicht rekursiv.

Beweisansatz: Diagonalisierung

Das Komplement zur Diagonalsprache ist

$$\bar{D} = \{ w \in \{0, 1\}^* \mid w = w_i \text{ und } M_i \text{ akzeptiert } w \}$$

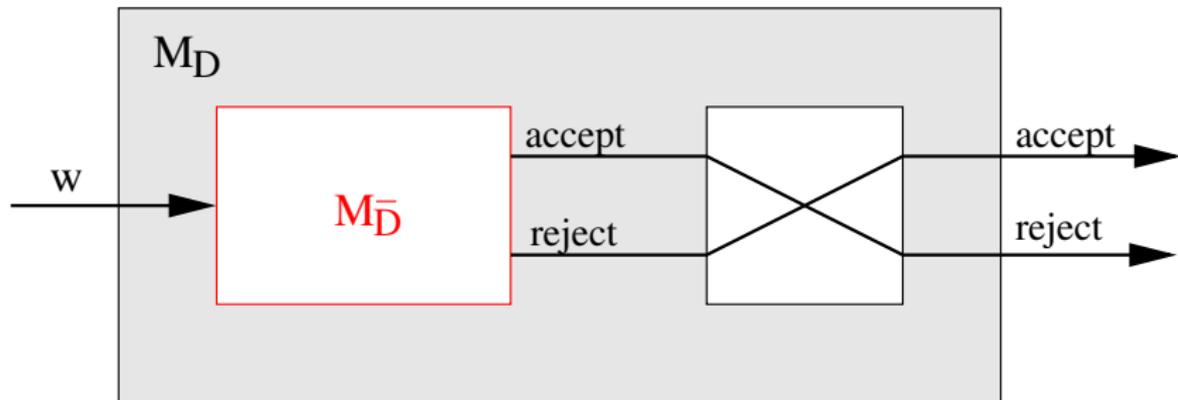
Satz:

Das Komplement \bar{D} der Diagonalsprache ist nicht rekursiv.

Beweis:

- Zum Widerspruch nehmen wir an, es gibt eine TM $M_{\bar{D}}$, die die Sprache \bar{D} entscheidet.
- Gemäß der Def *rekursiver Sprachen* hält $M_{\bar{D}}$ auf jeder Eingabe w und akzeptiert genau dann, wenn $w \in \bar{D}$.
- Wir konstruieren nun eine TM M , die $M_{\bar{D}}$ als Unterprogramm verwendet: M startet $M_{\bar{D}}$ auf der vorliegenden Eingabe und negiert anschließend die Ausgabe von $M_{\bar{D}}$.
- Die TM M entscheidet nun offensichtlich D . Ein Widerspruch zur Unentscheidbarkeit von D . □

Illustration: Aus $M_{\bar{D}}$ konstruieren wir M_D .



Aber die Existenz von M_D steht im Widerspruch zur Unentscheidbarkeit von D . Damit kann es $M_{\bar{D}}$ nicht geben, und \bar{D} ist nicht entscheidbar.

Die Beweistechnik aus diesem Satz lässt sich allgemein wie folgt zusammenfassen:

Unterprogrammtechnik zum Nachweis von Unentscheidbarkeit

Um nachzuweisen, dass eine Sprache L nicht rekursiv ist, genügt es zu zeigen, dass man durch Unterprogrammaufruf einer TM M_L , die L entscheidet, ein anderes Problem L' entscheiden kann, das bereits als nicht rekursiv bekannt ist.

Im Folgenden üben wir die Unterprogrammtechnik an einigen Beispielsprachen, die auch das Halteproblem umfassen.

Das Halteproblem ist wie folgt definiert

$$H = \{ \langle M \rangle w \mid M \text{ hält auf } w \} .$$

Satz:

Das Halteproblem H ist nicht rekursiv.

Beweis:

Wir nutzen die Unterprogrammtechnik:

- Sei M_H eine TM die H entscheidet, also eine TM, die auf jeder Eingabe hält, und nur Eingaben der Form $\langle M \rangle w$ akzeptiert, bei denen M auf w hält.
- Wir konstruieren eine TM $M_{\bar{D}}$ mit M_H als Unterprogramm, die \bar{D} entscheidet, was im Widerspruch zur Nicht-Berechenbarkeit von \bar{D} steht.

Aus diesem Widerspruch ergibt sich die Unmöglichkeit der TM M_H .

Algorithmus der TM $M_{\bar{D}}$ mit Unterprogramm M_H :

- 1) Auf Eingabe w , berechne i , so dass gilt $w = w_i$.
- 2) Berechne nun die Gödelnummer der i -ten TM, also $\langle M_i \rangle$.
- 3) Jetzt starte M_H als Unterprogramm mit Eingabe $\langle M_i \rangle w$.
 - 3.1) Falls M_H akzeptiert, so simuliere das Verhalten von M_i auf w (genau wie die universelle TM U dies tun würde).
 - 3.2) Falls M_H verwirft, so verwirf die Eingabe.