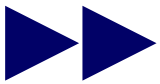
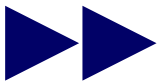
Theoretische Informatik
Kap 2: Berechnungstheorie



Gliederung der Vorlesung

0. Grundbegriffe
1. Formale Sprachen/Automatentheorie
    - 1.1. Grammatiken
    - 1.2. Reguläre Sprachen
    - 1.3. Kontextfreie Sprachen
2. Berechnungstheorie
    - 2.1. Berechenbarkeitsmodelle
    - 2.2. Die Churchsche These
    - **2.3. Unentscheidbarkeit**
3. Komplexitätstheorie
    - 3.1. Nicht-deterministische Turing Maschinen
    - 3.1 Komplexitätsmaße
    - 3.2. Das P=NP? Problem

Theoretische Informatik
Kap 2: Berechnungstheorie



Unentscheidbarkeit

Beispiele für Entscheidungsprobleme

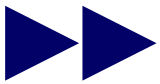
Zusammengesetzte Zahl

- Gegeben: eine natürliche Zahl k
- Frage: Gibt es natürliche Zahlen a>1 und b>1 mit a*b = k?

Erfüllbarkeit

- Gegeben: eine aussagenlogische Formel F
- Frage: Ist F erfüllbar?

Theoretische Informatik
Kap 2: Berechnungstheorie



Unentscheidbarkeit

Beispiele für Entscheidungsprobleme

10. Hilbertsches Problem

- Gegeben: eine Diophantische Gleichung $D(x_1,...,x_n) = 0$
- Frage: Gibt es ganze Zahlen a_1, ..., a_n, so daß $D(a_1,...,a_n) = 0$?

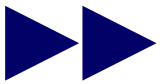
Totalitätsproblem für C++-Funktionen

- Gegeben: eine Funktion f in C++
- Frage: Berechnet diese Funktion für jede Eingabe ein Ergebnis?

Äquivalenzproblem für C++-Funktionen

- Gegeben: Zwei Definitionen von Funktionen f1 und f2 in C++.
- Frage: Leisten die Funktionen f1 und f2 dasselbe?

Theoretische Informatik
Kap 2: Berechnungstheorie



Unentscheidbarkeit

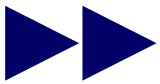
- ein Entscheidungsproblem Π

  ... ist eine allgemeine Ja/Nein-Frage mit einer Reihe von unspezifizierten Parametern

- eine Instanz eines Entscheidungsproblems Π

  ... entsteht, indem die Parameter spezifiziert werden

Das Entscheidungsproblem Π ist lösbar (unlösbar), wenn es einen (keinen) Algorithmus gibt, der für jede Instanz des Entscheidungs-problems Π die zugehörige Ja/Nein-Frage korrekt beantwortet.



Unentscheidbarkeit

Abstraktion

es sei Π ein Entscheidungsproblem

es sei cod ein geeignetes Kodierungsschema, mit dem man jeder Instanz I des Entscheidungsproblems Π eindeutig eine natürliche Zahl n = cod(I) zuordnen kann

Die zum Entscheidungsproblem Π gehörige Sprache $L(\Pi) \subseteq N$ enthält alle natürliche Zahlen n, so daß gilt:

- für die Instanz I des Entscheidungsproblems Π mit cod(I) = n ist die zugehörige Ja/Nein-Frage mit Ja zu beantworten

Theoretische Informatik
Kap 2: Berechnungstheorie

Unentscheidbarkeit

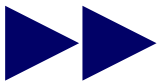
zentraler Begriff

es sei $L \subseteq N$

Die Sprache L heißt entscheidbar, falls es einen Algorithmus A gibt, der für jedes $x \in N$ folgendes leistet:

- A rechnet bei Eingabe von x nur endlich viele Schritte
- falls $x \in L$ gilt, so gibt A eine 1 aus
- falls $x \notin L$ gilt, so gibt A eine 0 aus

Theoretische Informatik
Kap 2: Berechnungstheorie



Unentscheidbarkeit

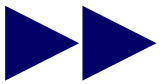
zentraler Begriff (/* unter Verwendung der Churchschen These */)

es sei $L \subseteq N$

Die Sprache L heißt entscheidbar, falls es eine Turing-Maschine M gibt, die für jedes $x \in N$ folgendes leistet:

- $f_M(x)$ ist definiert, d.h. M überführt die Anfangskonfiguration $z_0 bin(x) B$ in eine Endkonfiguration $uz_e v$
- falls $x \in L$ gilt, so ist $f_M(x) = 1$, d.h. $v = 1B$
- falls $x \notin L$ gilt, so ist $f_M(x) = 0$, d.h. $v = 0B$

Hinweis: f_M bezeichnet wie üblich die von M berechnete einstellige Funktion



Unentscheidbarkeit

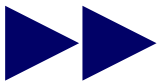
Konsequenz

es sei Π ein Entscheidungsproblem

es sei cod ein geeignetes Kodierungsschema, mit dem man jeder Instanz I des Entscheidungsproblems Π eindeutig eine natürliche Zahl n = cod(I) zuordnen kann

Das Entscheidungsproblem Π ist lösbar genau dann, wenn die zugehörige Sprache L(Π) entscheidbar ist.

Theoretische Informatik
Kap 2: Berechnungstheorie



Unentscheidbarkeit

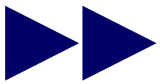
wir wissen bereits ...

es gibt unentscheidbare Sprachen $L \subseteq N$

... Diagonalisierung

uns interessiert ...

Gibt es interessante Entscheidungsprobleme Π, so daß die zugehörigen Sprachen $L(\Pi)$ unentscheidbar sind?



Unentscheidbarkeit

Beispiele für unlösbare Entscheidungsprobleme

Halteproblem

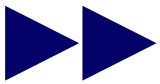
- Gegeben: eine Turing-Maschine M und eine Zahl x
- Frage: Stoppt M auf der Eingabe bin(x)? m.a.W.: Ist $f_M(x)$ definiert?

Totalitätsproblem

- Gegeben: eine Turing-Maschine M
- Frage: Stoppt M auf jeder Eingabe bin(x)? m.a.W.: Ist $f_M(x)$ für alle $x \in N$ definiert?

Äquivalenzproblem

- Gegeben: zwei Turing-Maschinen M und M'
- Frage: Berechnen M und M' dieselbe einstellige Funktion über den natürlichen Zahlen? m.a.W: Gilt für alle $x \in N$: $f_M(x) = f_{M'}(x)$?



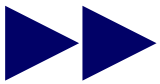
Unentscheidbarkeit

Fahrplan

- zeigen für ausgewählte Entscheidungsprobleme Π, daß sie unlösbar sind

  ... d.h. zeigen, daß die zugehörigen Sprachen L(Π) unentscheidbar sind

Theoretische Informatik
Kap 2: Berechnungstheorie



Unentscheidbarkeit

Halteproblem (HP)

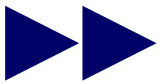
- Gegeben: eine Turing-Maschine M und eine Zahl x
- Frage: Stoppt M auf der Eingabe bin(x)? m.a.W.: Ist $f_M(x)$ definiert?

es sei M_0, M_1, M_2, ... eine effektive Aufzählung aller Programme
für Turing-Maschinen

dann ist die zum Halteproblem gehörige Sprache L(HP) wie folgt definiert:

$$L(HP) = \{ (j,x) \mid f_{M_j}(x) \text{ ist definiert } \}$$

Behauptung: die Sprache L(HP) ist unentscheidbar



Unentscheidbarkeit

Beweisidee (/* Diagonalisierung */):

1. wir definieren eine einstellige Funktion f*
2. wir zeigen, daß f* nicht berechenbar ist
3. wir zeigen folgende Implikation:

   wenn die Sprache L(HP) entscheidbar ist, so ist die Funktion f* berechenbar

Unentscheidbarkeit

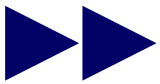
Schritt 1:

es sei f* wie folgt definiert

für alle $z \in N$ gilt:

$$f^*(z) = \begin{cases} f_{M_z}(z) + 1, & \text{falls } f_{M_z}(z) = y \text{ für ein } y \in N \\ 0, & \text{falls } f_{M_z}(z) \text{ undefiniert ist} \end{cases}$$

Anmerkung 1: offenbar gilt nach Definition $f^*(z) \neq f_{M_z}(z)$ für alle $z \in N$



Unentscheidbarkeit

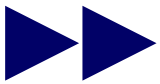
Schritt 2: wir nehmen an, daß f* berechenbar ist

dann gibt es eine TM M, so daß gilt: $f_M(z) = f^*(z)$ für alle $z \in N$

dann gibt es $j \in N$, so daß für die TM M_j gilt: $f_{M_j}(z) = f_M(z) = f^*(z)$ für alle $z \in N$

betrachte z = j

- dann muß insbesondere $f_{M_j}(j) = f^*(j)$ gelten
- das kann aber nicht der Fall sein (/* siehe Anmerkung (1) auf der letzten Folie */)



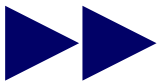
Unentscheidbarkeit

Schritt 3: wir nehmen an, daß die Menge L(HP) entscheidbar ist

d.h. es gibt eine TM M, die für alle j, x $\in$ N folgendes leistet

- $f_M(j,x)$ ist definiert
- falls (j,x) $\in$ L(HP), so ist $f_M(j,x) = 1$
- falls (j,x) $\notin$ L(HP), so ist $f_M(j,x) = 0$

wir definieren nun eine TM M*, die die Funktion f* berechnet

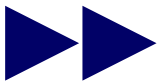


Unentscheidbarkeit

Schritt 3: Definition von M* (/* prinzipielle Idee */)

1. M* startet die TM M auf der Eingabe Bbin(j)Bbin(j)B
2. nachdem M den Endzustand erreicht hat, prüft M*, ob das Ergebnis 0 oder 1 auf dem Band steht
   a. falls eine 0 auf dem Band steht, so stoppt M*
   b. falls eine 1 auf dem Band steht, so
      - startet M* die TM M_j auf der Eingabe Bbin(j)B
      - nachdem M_j den Endzustand erreicht hat, startet M eine TM N, die die Funktion f(x) = x+1 berechnet
      - sobald N den Endzustand erreicht hat, stoppt M*

Hinweis: es sei bin(j) die aktuelle Eingabe von M*, M* muß sicherstellen, daß auf dem Band stets die Eingabe bin(j) „verfügbar“ ist



Unentscheidbarkeit

Zusammenfassung des Beweises

- haben nachgewiesen, daß die Funktion f* nicht berechenbar ist
- haben nachgewiesen, daß unter der Annahme, daß die Sprache L(HP) entscheidbar ist, die Funktion f* berechenbar wäre

Konsequenz

- die Sprache L(HP) ist nicht entscheidbar

mit anderen Worten:

das Halteproblem ist unlösbar !!!

Unentscheidbarkeit

Äquivalenzproblem

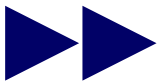
- Gegeben: zwei Turing-Maschinen M und M'
- Frage: Berechnen M und M' dieselbe einstellige Funktion über den natürlichen Zahlen? m.a.W: Gilt für alle $x \in N$: $f_M(x) = f_{M'}(x)$?

es sei M_0, M_1, M_2, ... eine effektive Aufzählung aller Turing-Maschinen

dann ist die zum Äquivalenzproblem gehörige Sprache L(ÄP) wie folgt definiert:

$$L(ÄP) = \{ (k,l) \mid f_{M_k}(x) = f_{M_l}(x) \text{ für alle } x \in N \}$$

Behauptung: die Sprache L(ÄP) ist unentscheidbar



Unentscheidbarkeit

Beweisidee (/* Reduktion */):

wir zeigen, daß man zu jeder Instanz I des Halteproblems eine Instanz t(I) des Äquivalenzproblems konstruieren kann, so daß folgender Zusammenhang gilt:

- wenn für die Instanz I des HP die zugehörige Ja/Nein-Frage mit Ja (/* Nein */) zu beantworten ist, so ist die zugehörige Ja/Nein-Frage für die Instanz t(I) des ÄP auch mit Ja (/* Nein */) zu beantworten

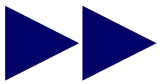
Unentscheidbarkeit

$L(HP) = \{ (j,x) \mid f_{M_j}(x) \text{ ist definiert } \}$
$L(ÄP) = \{ (k,l) \mid f_{M_k}(x) = f_{M_l}(x) \text{ für alle } x \in N \}$

es seien $j, x \in N$ /* als Instanz des HP */
bilde (j,x) ab auf bestimmte k, l /* als Instanz des ÄP */
/* $k=t(j,x,1)$, $l=t(j,x,2)$ */

wir definieren dazu zwei TM und $M_{t(j,x,1)}$ und $M_{t(j,x,2)}$, so daß gilt:

- $M_{t(j,x,1)}$ und $M_{t(j,x,2)}$ berechnen genau dann dieselbe einstellige Funktion, wenn die von der TM M_j berechnete Funktion an der Stelle x definiert ist

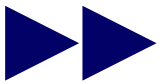


Unentscheidbarkeit

Definition von $M_{t(j,x,1)}$

1. $M_{t(j,x,1)}$ löscht die aktuelle Eingabe vom Band
2. $M_{t(j,x,1)}$ schreibt eine 0 aufs Band und stoppt

offenbar berechnet die TM $M_{t(j,x,1)}$ die einstellige Funktion f mit $f(z) = 0$ für alle $z \in N$



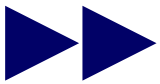
Unentscheidbarkeit

Definition von $M_{t(j,x,2)}$

1. $M_{t(j,x,2)}$ löscht die aktuelle Eingabe vom Band
2. $M_{t(j,x,2)}$ schreibt bin(x) aufs Band und startet die TM M_j
3. falls M_j den Endzustand erreicht, so löscht $M_{t(j,x,2)}$ das Ergebnis, schreibt eine 0 aufs Band und stoppt

offenbar berechnet die TM $M'_{t(j,x,2)}$ die folgende einstellige Funktion f:

- f(z) = 0 für alle $z \in N$ (/* falls $f_{M_j}(x)$ definiert ist */)
- f(z) ist undefiniert für alle $z \in N$ (/* falls $f_{M_j}(x)$ undefiniert ist */)



Unentscheidbarkeit

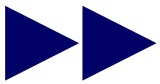
zentrale Beobachtung

Es gibt einen Algorithmus A, der folgendes leistet:

- A akzeptiert als Eingabe ein Programm für eine TM M_j und eine Zahl x
- A produziert als Ausgabe ein Programm für die TM $M_{t(j,x,1)}$ und ein Programm für die TM $M_{t(j,x,2)}$

... damit wird zu einer Instanz I = (j,x) des Halteproblems eine Instanz t(I) = (t(j,x,1),t(j,x,2)) des Äquivalenzproblems berechnet

... I ist eine „Ja“-Instanz des Halteproblems genau dann, wenn t(I) eine „Ja“-Instanz des Äquivalenzproblems ist



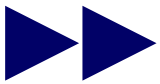
Unentscheidbarkeit

Konsequenz

wenn die Sprache L(ÄP) entscheidbar ist, so ist auch die Sprache L(HP) entscheidbar

algorithmische Idee

1. bestimme unter Verwendung von Algorithmus A die Programme der TM $M_{t(j,x,1)}$ und TM $M_{t(j,x,2)}$
2. bestimme die Indizes k, l $\in$ N der Programme der TM $M_{t(j,x,1)}$ und TM $M_{t(j,x,2)}$ in der effektiven Aufzählung M_0, M_1, M_2, ... aller TM
3. starte die TM, die die Sprache L(ÄP) entscheidet, für die Eingabe bin(k)Bbin(l)



Unentscheidbarkeit

Zusammenfassung des Beweises

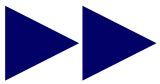
- haben nachgewiesen, daß unter der Annahme, daß die Sprache L(ÄP) entscheidbar ist, auch die Sprache L(HP) entscheidbar ist
- wir wissen bereits, daß die Sprache L(HP) nicht entscheidbar ist

Konsequenz

- die Sprache L(ÄP) ist nicht entscheidbar

mit anderen Worten:

das Äquivalenzproblem ist unlösbar !!!



Unentscheidbarkeit

Hinweis

- der Beweis, daß die Sprache L(ÄP) unentscheidbar ist, basiert auf einer sehr allgemeinen Idee

Zielstellung

... man möchte bereits vorhandenes Wissen über die Lösbarkeit bzw. Unlösbarkeit von Entscheidungsproblemen verwenden, um herauszubekommen, ob andere interessierende Entscheidungsprobleme lösbar bzw. unlösbar sind

d.h. man möchte bereits vorhandenes Wissen über die Entscheidbarkeit bzw. Unentscheidbarkeit der zugehörigen Sprachen verwenden, um herauszubekommen, ob ...

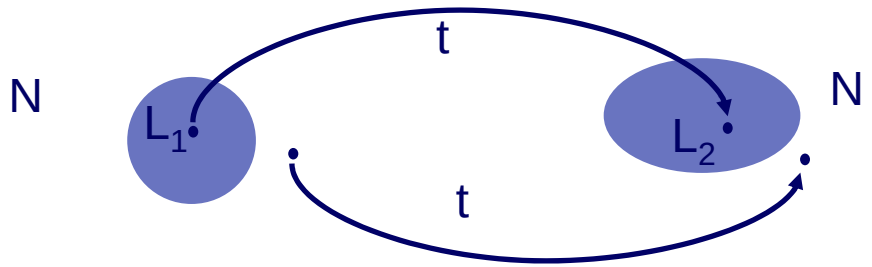
Theoretische Informatik
Kap 2: Berechnungstheorie

Unentscheidbarkeit

Es seien $L_1, L_2 \subseteq N$. Die Sprache L_1 ist auf die Sprache L_2 reduzierbar, falls es eine einstellige berechenbare Funktion $t: N \rightarrow N$ gibt, so daß für alle $x \in N$ gilt:

- t(x) ist definiert
- $x \in L_1$ gdw. $t(x) \in L_2$

Bezeichnung: $L_1 \leq_T L_2$



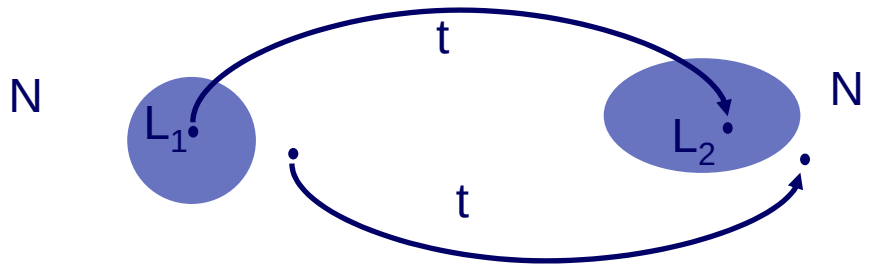
Theoretische Informatik
Kap 2: Berechnungstheorie

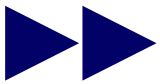
Unentscheidbarkeit

Es seien $L_1, L_2 \subseteq N$. Die Sprache L_1 ist auf die Sprache L_2 reduzierbar, falls es eine von einer TM berechenbare Funktion t: N → N gibt, so daß für alle $x \in N$ gilt:

- t(x) ist definiert
- $x \in L_1$ gdw. $t(x) \in L_2$

Bezeichnung: $L_1 \leq_T L_2$





Unentscheidbarkeit

Konsequenzen

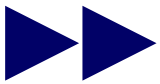
Es seien $L_1, L_2 \subseteq N$ mit $L_1 \leq_T L_2$. Dann gilt:

(1) Wenn L_2 entscheidbar ist, so ist auch L_1 entscheidbar.
(2) Wenn L_1 unentscheidbar ist, so ist auch L_2 unentscheidbar.

Nachweis von (1): „Komposition“
Nachweis von (2): folgt unmittelbar aus (1)

Interpretation

$L_1 \leq_T L_2$ bedeutet ... das Problem, die Sprache L_2 zu entscheiden, ist mindestens so „schwierig“, wie das Problem, die Sprache L_1 zu entscheiden



Unentscheidbarkeit

Verwendung

es sei Π_2 das interessierende Entscheidungsproblem, von dem man vermutet, daß es unlösbar ist

Schritt 1:

wähle ein „geeignetes“ Entscheidungsproblem Π_1, von dem bekannt ist, daß es unlösbar ist

Schritt 2:

zeige, daß für die zugehörigen Sprachen $L(\Pi_1)$ und $L(\Pi_2)$ gilt:

$L(\Pi_1) \leq_T L(\Pi_2)$