

Formale Grundlagen der Informatik 1

Kapitel 11

Aufzählbarkeit und (Un-)Entscheidbarkeit (Teil 2)

Frank Heitmann

heitmann@informatik.uni-hamburg.de

12. Mai 2015

Entscheidbarkeit vs. Aufzählbarkeit

Entscheidbarkeit von M :

- Es gibt eine TM A mit $L(A) = M$.
- **und** A hält auf jeder Eingabe (in einem Endzustand, wenn das vorgelegte Wort in M ist, sonst in einem Nicht-Endzustand).

Aufzählbarkeit von M :

- Es gibt eine TM A mit $L(A) = M$.

Definition

Die von Turing-Maschinen akzeptierten Sprachen bilden die Sprachfamilie RE der **aufzählbaren Sprachen**. Die Sprachen, die von TMs akzeptiert werden, so dass die TM zusätzlich auf jeder Eingabe anhält, bilden die Sprachfamilie REC der **entscheidbaren Sprachen**.

Entscheidbare Sprachen

Satz

Folgende Sprachen sind entscheidbar:

$$DFA_{acc} := \{ \langle A, w \rangle \mid A \text{ ist ein DFA und akzeptiert } w \}$$

$$NFA_{acc} := \{ \langle A, w \rangle \mid A \text{ ist ein NFA und akzeptiert } w \}$$

$$DFA_{\emptyset} := \{ \langle A \rangle \mid A \text{ ist ein DFA und } L(A) = \emptyset \}$$

$$DFA_{=} := \{ \langle A, B \rangle \mid A \text{ und } B \text{ sind DFAs mit } L(A) = L(B) \}$$

$$CFG_{acc} := \{ \langle G, w \rangle \mid G \text{ ist eine CFG und generiert } w \}$$

$$L_{+1} := \{ \langle x, y \rangle \mid y = x + 1 \}$$

$$L_{+} := \{ \langle x, y, z \rangle \mid x + y = z \}$$

$$L := \{ \langle G, v \rangle \mid G \text{ ist ein Graph und } v \text{ ein erreichbarer Knoten} \}$$

*Ferner ist jede reguläre und jede kontextfreie Sprache entscheidbar.
(Und daneben noch viele, viele mehr ...)*

Das Akzeptanz-/Halteproblem

Satz

Die folgende Sprache ist aufzählbar

$$TM_{acc} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und akzeptiert } w \}$$

Satz

Die Sprache TM_{acc} von oben ist nicht entscheidbar.

Die Sprache L_d

Man kann die Wörter eines Alphabets aufzählen, z.B. indem man sie erst nach Länge und dann lexikalisch sortiert:

$a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \dots$

So kann man sowohl die Kodierungen aller Turing-Maschinen aufzählen als auch die Wörter, die sie lesen. Wir können dann von den Turing-Maschinen M_1, M_2, M_3, \dots und den Wörtern w_1, w_2, w_3, \dots sprechen.

Anmerkung

Genauer kann man hier die lexikalische Ordnung und Gödelisierungen einführen. Siehe Skript.

Die Sprache L_d

Man kann die Wörter eines Alphabets aufzählen, z.B. indem man sie erst nach Länge und dann lexikalisch sortiert:

$a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \dots$

So kann man sowohl die Kodierungen aller Turing-Maschinen aufzählen als auch die Wörter, die sie lesen. Wir können dann von den Turing-Maschinen M_1, M_2, M_3, \dots und den Wörtern w_1, w_2, w_3, \dots sprechen.

Anmerkung

Genauer kann man hier die lexikalische Ordnung und Gödelisierungen einführen. Siehe Skript.

Die Sprache L_d

Man kann die Wörter eines Alphabets aufzählen, z.B. indem man sie erst nach Länge und dann lexikalisch sortiert:

$a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \dots$

So kann man sowohl die Kodierungen aller Turing-Maschinen aufzählen als auch die Wörter, die sie lesen. Wir können dann von den Turing-Maschinen M_1, M_2, M_3, \dots und den Wörtern w_1, w_2, w_3, \dots sprechen.

Anmerkung

Genauer kann man hier die lexikalische Ordnung und Gödelisierungen einführen. Siehe Skript.

Die Sprache L_d

Satz

$L_d := \{w_i \mid w_i \notin L(A_i)\}$ ist nicht aufzählbar.

Die Sprache L_d

Satz

$L_d := \{w_i \mid w_i \notin L(A_i)\}$ ist nicht aufzählbar.

Beweis

Man kann die Wörter und die TMs in einer Matrix anordnen:

	M_1	M_2	M_3	M_4	\dots	L_d
w_1	1	1	0	0		–
w_2	1	0	0	0	\dots	Ja
w_3	0	1	1	1	\dots	–
w_4	0	0	1	0		Ja
.

L_d entspricht gerade der Diagonalen, von der nur die Einträge mit 0 in L_d aufgenommen werden.

Die Sprache L_d

$$L_d := \{w_i \mid w_i \notin L(A_i)\} \notin RE$$

Das Problem ist, dass eine TM, die L_d akzeptiert, ein M_j in der Matrix sein müsste und dann wird es einen Widerspruch mit w_j geben (akzeptieren/nicht akzeptieren).

Genauer: Angenommen L_d wäre aufzählbar. Dann gibt es eine TM A_j aus der Aufzählung mit $L(A_j) = L_d$. Wir betrachten w_j . Es muss entweder $w_j \in L_d$ oder $w_j \notin L_d$ sein. Beide Fälle führen zu einem Widerspruch:

- $w_j \in L_d$, dann $w_j \notin L(A_j)$ (nach Def. von L_d), dann $w_j \notin L_d$ (nach Annahme $L(A_j) = L_d$).
- $w_j \notin L_d$, dann $w_j \notin L(A_j)$ (nach Annahme), dann $w_j \in L_d$ (Def. von L_d).

Also ist L_d nicht aufzählbar! □

Die Sprache L_d

$$L_d := \{w_j \mid w_j \notin L(A_j)\} \notin RE$$

Das Problem ist, dass eine TM, die L_d akzeptiert, ein M_j in der Matrix sein müsste und dann wird es einen Widerspruch mit w_j geben (akzeptieren/nicht akzeptieren).

Genauer: Angenommen L_d wäre aufzählbar. Dann gibt es eine TM A_j aus der Aufzählung mit $L(A_j) = L_d$. Wir betrachten w_j . Es muss entweder $w_j \in L_d$ oder $w_j \notin L_d$ sein. Beide Fälle führen zu einem Widerspruch:

- $w_j \in L_d$, dann $w_j \notin L(A_j)$ (nach Def. von L_d), dann $w_j \notin L_d$ (nach Annahme $L(A_j) = L_d$).
- $w_j \notin L_d$, dann $w_j \notin L(A_j)$ (nach Annahme), dann $w_j \in L_d$ (Def. von L_d).

Also ist L_d nicht aufzählbar! □

Die Sprache L_d

$$L_d := \{w_j \mid w_j \notin L(A_j)\} \notin RE$$

Das Problem ist, dass eine TM, die L_d akzeptiert, ein M_j in der Matrix sein müsste und dann wird es einen Widerspruch mit w_j geben (akzeptieren/nicht akzeptieren).

Genauer: Angenommen L_d wäre aufzählbar. Dann gibt es eine TM A_j aus der Aufzählung mit $L(A_j) = L_d$. Wir betrachten w_j . Es muss entweder $w_j \in L_d$ oder $w_j \notin L_d$ sein. Beide Fälle führen zu einem Widerspruch:

- $w_j \in L_d$, dann $w_j \notin L(A_j)$ (nach Def. von L_d), dann $w_j \notin L_d$ (nach Annahme $L(A_j) = L_d$).
- $w_j \notin L_d$, dann $w_j \notin L(A_j)$ (nach Annahme), dann $w_j \in L_d$ (Def. von L_d).

Also ist L_d nicht aufzählbar!



Die Sprache L_d

$$L_d := \{w_j \mid w_j \notin L(A_j)\} \notin RE$$

Das Problem ist, dass eine TM, die L_d akzeptiert, ein M_j in der Matrix sein müsste und dann wird es einen Widerspruch mit w_j geben (akzeptieren/nicht akzeptieren).

Genauer: Angenommen L_d wäre aufzählbar. Dann gibt es eine TM A_j aus der Aufzählung mit $L(A_j) = L_d$. Wir betrachten w_j . Es muss entweder $w_j \in L_d$ oder $w_j \notin L_d$ sein. Beide Fälle führen zu einem Widerspruch:

- $w_j \in L_d$, dann $w_j \notin L(A_j)$ (nach Def. von L_d), dann $w_j \notin L_d$ (nach Annahme $L(A_j) = L_d$).
- $w_j \notin L_d$, dann $w_j \notin L(A_j)$ (nach Annahme), dann $w_j \in L_d$ (Def. von L_d).

Also ist L_d nicht aufzählbar!



Die Sprache L_d

$$L_d := \{w_j \mid w_j \notin L(A_j)\} \notin RE$$

Das Problem ist, dass eine TM, die L_d akzeptiert, ein M_j in der Matrix sein müsste und dann wird es einen Widerspruch mit w_j geben (akzeptieren/nicht akzeptieren).

Genauer: Angenommen L_d wäre aufzählbar. Dann gibt es eine TM A_j aus der Aufzählung mit $L(A_j) = L_d$. Wir betrachten w_j . Es muss entweder $w_j \in L_d$ oder $w_j \notin L_d$ sein. Beide Fälle führen zu einem Widerspruch:

- $w_j \in L_d$, dann $w_j \notin L(A_j)$ (nach Def. von L_d), dann $w_j \notin L_d$ (nach Annahme $L(A_j) = L_d$).
- $w_j \notin L_d$, dann $w_j \notin L(A_j)$ (nach Annahme), dann $w_j \in L_d$ (Def. von L_d).

Also ist L_d nicht aufzählbar!



Die Sprache L_d

$$L_d := \{w_j \mid w_j \notin L(A_j)\} \notin RE$$

Das Problem ist, dass eine TM, die L_d akzeptiert, ein M_j in der Matrix sein müsste und dann wird es einen Widerspruch mit w_j geben (akzeptieren/nicht akzeptieren).

Genauer: Angenommen L_d wäre aufzählbar. Dann gibt es eine TM A_j aus der Aufzählung mit $L(A_j) = L_d$. Wir betrachten w_j . Es muss entweder $w_j \in L_d$ oder $w_j \notin L_d$ sein. Beide Fälle führen zu einem Widerspruch:

- $w_j \in L_d$, dann $w_j \notin L(A_j)$ (nach Def. von L_d), dann $w_j \notin L_d$ (nach Annahme $L(A_j) = L_d$).
- $w_j \notin L_d$, dann $w_j \notin L(A_j)$ (nach Annahme), dann $w_j \in L_d$ (Def. von L_d).

Also ist L_d nicht aufzählbar!



Die Sprache L_d und darüber hinaus...

- L_d ist also nicht einmal aufzählbar!
- Sie ist aber *abzählbar*!
- Eine Menge, die dann nicht einmal mehr abzählbar ist, ist z.B. die Menge der reellen Zahlen.

Bemerkung

Eine Menge ist abzählbar, wenn sie endlich ist oder eine Bijektion von den natürlichen Zahlen angegeben werden kann.

Die Sprache L_d und darüber hinaus...

- L_d ist also nicht einmal aufzählbar!
- Sie ist aber *ab*zählbar!
- Eine Menge, die dann nicht einmal mehr abzählbar ist, ist z.B. die Menge der reellen Zahlen.

Bemerkung

Eine Menge ist abzählbar, wenn sie endlich ist oder eine Bijektion von den natürlichen Zahlen angegeben werden kann.

Zusammenfassung

Wir haben bisher kennengelernt/gesehen:

- REC - entscheidbare Sprachen
- RE - aufzählbare Sprachen
- mit TM_{acc} eine Sprache, die aufzählbar, aber nicht mehr entscheidbar ist (unentscheidbar)
- mit L_d eine Sprache, die nicht mehr aufzählbar ist

Damit:

$$REG \subsetneq CF \subsetneq REC \subsetneq RE$$

Zusammenfassung

Wir haben bisher kennengelernt/gesehen:

- REC - entscheidbare Sprachen
- RE - aufzählbare Sprachen
- mit TM_{acc} eine Sprache, die aufzählbar, aber nicht mehr entscheidbar ist (unentscheidbar)
- mit L_d eine Sprache, die nicht mehr aufzählbar ist

Damit:

$$REG \subsetneq CF \subsetneq REC \subsetneq RE$$

Zusammenfassung

Wir haben bisher kennengelernt/gesehen:

- REC - entscheidbare Sprachen
- RE - aufzählbare Sprachen
- mit TM_{acc} eine Sprache, die aufzählbar, aber nicht mehr entscheidbar ist (unentscheidbar)
- mit L_d eine Sprache, die nicht mehr aufzählbar ist

Damit:

$$REG \subsetneq CF \subsetneq REC \subsetneq RE$$

Zusammenfassung

Wir haben bisher kennengelernt/gesehen:

- REC - entscheidbare Sprachen
- RE - aufzählbare Sprachen
- mit TM_{acc} eine Sprache, die aufzählbar, aber nicht mehr entscheidbar ist (unentscheidbar)
- mit L_d eine Sprache, die nicht mehr aufzählbar ist

Damit:

$$REG \subsetneq CF \subsetneq REC \subsetneq RE$$

Zusammenfassung

Wir haben bisher kennengelernt/gesehen:

- REC - entscheidbare Sprachen
- RE - aufzählbare Sprachen
- mit TM_{acc} eine Sprache, die aufzählbar, aber nicht mehr entscheidbar ist (unentscheidbar)
- mit L_d eine Sprache, die nicht mehr aufzählbar ist

Damit:

$$REG \subsetneq CF \subsetneq REC \subsetneq RE$$

Zusammenfassung

Wir haben bisher kennengelernt/gesehen:

- REC - entscheidbare Sprachen
- RE - aufzählbare Sprachen
- mit TM_{acc} eine Sprache, die aufzählbar, aber nicht mehr entscheidbar ist (unentscheidbar)
- mit L_d eine Sprache, die nicht mehr aufzählbar ist

Damit:

$$REG \subsetneq CF \subsetneq REC \subsetneq RE$$

Reduktion

Das Ziel

Wir wollen nun weitere Probleme als unentscheidbar nachweisen

Will man eine Sprache L als unentscheidbar nachweisen, so ist das übliche Vorgehen:

- 1 Wir nehmen an L wäre entscheidbar. Sei M eine TM, die L entscheidet.
- 2 Unter Nutzung von M nun eine TM konstruieren, die ein schon als unentscheidbar nachgewiesenes Problem entscheidet.
- 3 Das ist dann ein Widerspruch, also kann M nicht existieren.

Dies nennt man eine *Reduktion*, und sagt, dass das unentscheidbare Problem auf das neue Problem reduziert wurde.

Reduktion

Das Ziel

Wir wollen nun weitere Probleme als unentscheidbar nachweisen

Will man eine Sprache L als unentscheidbar nachweisen, so ist das übliche Vorgehen:

- 1 Wir nehmen an L wäre entscheidbar. Sei M eine TM, die L entscheidet.
- 2 Unter Nutzung von M nun eine TM konstruieren, die ein schon als unentscheidbar nachgewiesenes Problem entscheidet.
- 3 Das ist dann ein Widerspruch, also kann M nicht existieren.

Dies nennt man eine *Reduktion*, und sagt, dass das unentscheidbare Problem auf das neue Problem reduziert wurde.

Reduktion

Das Ziel

Wir wollen nun weitere Probleme als unentscheidbar nachweisen

Will man eine Sprache L als unentscheidbar nachweisen, so ist das übliche Vorgehen:

- 1 Wir nehmen an L wäre entscheidbar. Sei M eine TM, die L entscheidet.
- 2 Unter Nutzung von M nun eine TM konstruieren, die ein schon als unentscheidbar nachgewiesenes Problem entscheidet.
- 3 Das ist dann ein Widerspruch, also kann M nicht existieren.

Dies nennt man eine *Reduktion*, und sagt, dass das unentscheidbare Problem auf das neue Problem reduziert wurde.

Reduktion

Das Ziel

Wir wollen nun weitere Probleme als unentscheidbar nachweisen

Will man eine Sprache L als unentscheidbar nachweisen, so ist das übliche Vorgehen:

- 1 Wir nehmen an L wäre entscheidbar. Sei M eine TM, die L entscheidet.
- 2 Unter Nutzung von M nun eine TM konstruieren, die ein schon als unentscheidbar nachgewiesenes Problem entscheidet.
- 3 Das ist dann ein Widerspruch, also kann M nicht existieren.

Dies nennt man eine *Reduktion*, und sagt, dass das unentscheidbare Problem auf das neue Problem reduziert wurde.

Reduktion

Das Ziel

Wir wollen nun weitere Probleme als unentscheidbar nachweisen

Will man eine Sprache L als unentscheidbar nachweisen, so ist das übliche Vorgehen:

- 1 Wir nehmen an L wäre entscheidbar. Sei M eine TM, die L entscheidet.
- 2 Unter Nutzung von M nun eine TM konstruieren, die ein schon als unentscheidbar nachgewiesenes Problem entscheidet.
- 3 Das ist dann ein Widerspruch, also kann M nicht existieren.

Dies nennt man eine *Reduktion*, und sagt, dass das unentscheidbare Problem auf das neue Problem reduziert wurde.

Reduktion

Das Ziel

Wir wollen nun weitere Probleme als unentscheidbar nachweisen

Will man eine Sprache L als unentscheidbar nachweisen, so ist das übliche Vorgehen:

- 1 Wir nehmen an L wäre entscheidbar. Sei M eine TM, die L entscheidet.
- 2 Unter Nutzung von M nun eine TM konstruieren, die ein schon als unentscheidbar nachgewiesenes Problem entscheidet.
- 3 Das ist dann ein Widerspruch, also kann M nicht existieren.

Dies nennt man eine *Reduktion*, und sagt, dass das unentscheidbare Problem auf das neue Problem reduziert wurde.

Weitere unentscheidbare Probleme

Satz

$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$

ist nicht entscheidbar.

Anmerkung

TM_{halt} ist das **Halteproblem**. Dies wird oft in Unentscheidbarkeitsbeweisen benutzt.

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis

Angenommen TM_{halt} w\u00e4re entscheidbar. Sei H eine TM, die TM_{halt} entscheidet. Wir konstruieren nun eine TM S , die TM_{acc} (unter Nutzung von H) entscheidet. Da aber TM_{acc} nicht entscheidbar ist, muss die Annahme falsch sein und daher ist auch TM_{halt} nicht entscheidbar.

Anmerkung

TM_{halt} ist das **Halteproblem**. Dies wird oft in Unentscheidbarkeitsbeweisen benutzt.

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und hält auf } w \}$$

ist nicht entscheidbar.

Beweis

Angenommen TM_{halt} wäre entscheidbar. Sei H eine TM, die TM_{halt} entscheidet. Wir konstruieren nun eine TM S , die TM_{acc} (unter Nutzung von H) entscheidet. Da aber TM_{acc} nicht entscheidbar ist, muss die Annahme falsch sein und daher ist auch TM_{halt} nicht entscheidbar.

Anmerkung

TM_{halt} ist das **Halteproblem**. Dies wird oft in Unentscheidbarkeitsbeweisen benutzt.

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis.

H entscheidet nach Annahme TM_{halt} . S arbeitet nun bei Eingabe $\langle M, w \rangle$ wie folgt:

- 1 Starte H mit Eingabe $\langle M, w \rangle$
- 2 Lehnt H ab, lehne ab.
- 3 Akzeptiert H , simuliere M auf w (M terminiert, da H akzeptiert hat!)
- 4 Wenn M akzeptiert, akzeptiere; sonst lehne ab.

Wie eben beschrieben, kann TM_{halt} nicht entscheidbar sein. \square

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis.

H entscheidet nach Annahme TM_{halt} . S arbeitet nun bei Eingabe $\langle M, w \rangle$ wie folgt:

- 1 Starte H mit Eingabe $\langle M, w \rangle$
- 2 Lehnt H ab, lehne ab.
- 3 Akzeptiert H , simuliere M auf w (M terminiert, da H akzeptiert hat!)
- 4 Wenn M akzeptiert, akzeptiere; sonst lehne ab.

Wie eben beschrieben, kann TM_{halt} nicht entscheidbar sein. \square

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis.

H entscheidet nach Annahme TM_{halt} . S arbeitet nun bei Eingabe $\langle M, w \rangle$ wie folgt:

- ① Starte H mit Eingabe $\langle M, w \rangle$
- ② Lehnt H ab, lehne ab.
- ③ Akzeptiert H , simuliere M auf w (M terminiert, da H akzeptiert hat!)
- ④ Wenn M akzeptiert, akzeptiere; sonst lehne ab.

Wie eben beschrieben, kann TM_{halt} nicht entscheidbar sein. □

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis.

H entscheidet nach Annahme TM_{halt} . S arbeitet nun bei Eingabe $\langle M, w \rangle$ wie folgt:

- 1 Starte H mit Eingabe $\langle M, w \rangle$
- 2 Lehnt H ab, lehne ab.
- 3 Akzeptiert H , simuliere M auf w (M terminiert, da H akzeptiert hat!)
- 4 Wenn M akzeptiert, akzeptiere; sonst lehne ab.

Wie eben beschrieben, kann TM_{halt} nicht entscheidbar sein. \square

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis.

H entscheidet nach Annahme TM_{halt} . S arbeitet nun bei Eingabe $\langle M, w \rangle$ wie folgt:

- ① Starte H mit Eingabe $\langle M, w \rangle$
- ② Lehnt H ab, lehne ab.
- ③ Akzeptiert H , simuliere M auf w (M terminiert, da H akzeptiert hat!)
- ④ Wenn M akzeptiert, akzeptiere; sonst lehne ab.

Wie eben beschrieben, kann TM_{halt} nicht entscheidbar sein. □

Weitere unentscheidbare Probleme

Satz

$$TM_{halt} := \{ \langle M, w \rangle \mid M \text{ ist eine TM und h\u00e4lt auf } w \}$$

ist nicht entscheidbar.

Beweis.

H entscheidet nach Annahme TM_{halt} . S arbeitet nun bei Eingabe $\langle M, w \rangle$ wie folgt:

- ① Starte H mit Eingabe $\langle M, w \rangle$
- ② Lehnt H ab, lehne ab.
- ③ Akzeptiert H , simuliere M auf w (M terminiert, da H akzeptiert hat!)
- ④ Wenn M akzeptiert, akzeptiere; sonst lehne ab.

Wie eben beschrieben, kann TM_{halt} nicht entscheidbar sein. □

Weitere unentscheidbare Probleme

Satz

$TM_{print0} := \{ \langle M \rangle \mid \text{gestartet auf } \lambda \text{ gibt } M \text{ irgendwann } 0 \text{ aus} \}$

ist nicht entscheidbar.

Weitere unentscheidbare Probleme

Satz

$TM_{print0} := \{ \langle M \rangle \mid \text{gestartet auf } \lambda \text{ gibt } M \text{ irgendwann } 0 \text{ aus} \}$

ist nicht entscheidbar.

Beweis.

In den Präsenzaufgaben!



Weitere unentscheidbare Probleme

Definition

Ein *nutzloser Zustand* in einer TM ist ein Zustand, der bei keinem Eingabewort jemals betreten wird. Sei

$$\text{UselessState} = \{ \langle A, q \rangle \mid A \text{ ist eine TM und} \\ q \text{ ein nutzloser Zustand von } A \}$$

Bemerkung

Dieses Problem ist eng verwandt mit der Frage, ob eine bestimmte Stelle bspw. in einem Java-Programm jemals erreicht wird.

Weitere unentscheidbare Probleme

Satz

$UselessState = \{ \langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A \}$

ist unentscheidbar.

Beweis

Angenommen $UselessState$ wäre entscheidbar und A_{US} eine TM, die das Problem entscheidet. Wir wollen damit nun das Halteproblem TM_{halt} entscheiden.

Die Idee

- Zu einer gegebenen TM M mit Zustandsmenge Z und Bandalphabet Γ konstruieren wir eine TM M' mit:
 - einem neuen Zustand z_{neu}
 - für jeden Zustand $z \in Z$ von M und jedes Symbol $x \in \Gamma$, für das es keinen Übergang aus z in M gab (d.h. es gab keine Kante (z, x, X, Y, z') in M mit X, Y, z' beliebig) wird eine neue Kante (z, x, x, R, z_{neu}) hinzugefügt

Hält M , so kann M' noch einen Übergang nach z_{neu} machen. Dort hält M' . *M hält also genau dann in irgendeinem Zustand, wenn M' in z_{neu} hält.*
- Zu $\langle M, w \rangle$ kann außerdem eine TM M'' konstruiert werden, die bei jeder Eingabe das Band löscht und dann w auf M' startet, wobei M' wie eben beschreiben aus M hervorgeht.

Die Idee

- Zu einer gegebenen TM M mit Zustandsmenge Z und Bandalphabet Γ konstruieren wir eine TM M' mit:
 - einem neuen Zustand z_{neu}
 - für jeden Zustand $z \in Z$ von M und jedes Symbol $x \in \Gamma$, für das es keinen Übergang aus z in M gab (d.h. es gab keine Kante (z, x, X, Y, z') in M mit X, Y, z' beliebig) wird eine neue Kante (z, x, x, R, z_{neu}) hinzugefügtHält M , so kann M' noch einen Übergang nach z_{neu} machen. Dort hält M' . *M hält also genau dann in irgendeinem Zustand, wenn M' in z_{neu} hält.*
- Zu $\langle M, w \rangle$ kann außerdem eine TM M'' konstruiert werden, die bei jeder Eingabe das Band löscht und dann w auf M' startet, wobei M' wie eben beschreiben aus M hervorgeht.

Die Idee

- Zu einer gegebenen TM M mit Zustandsmenge Z und Bandalphabet Γ konstruieren wir eine TM M' mit:
 - einem neuen Zustand z_{neu}
 - für jeden Zustand $z \in Z$ von M und jedes Symbol $x \in \Gamma$, für das es keinen Übergang aus z in M gab (d.h. es gab keine Kante (z, x, X, Y, z') in M mit X, Y, z' beliebig) wird eine neue Kante (z, x, x, R, z_{neu}) hinzugefügt

Hält M , so kann M' noch einen Übergang nach z_{neu} machen. Dort hält M' . *M hält also genau dann in irgendeinem Zustand, wenn M' in z_{neu} hält.*

- Zu $\langle M, w \rangle$ kann außerdem eine TM M'' konstruiert werden, die bei jeder Eingabe das Band löscht und dann w auf M' startet, wobei M' wie eben beschreiben aus M hervorgeht.

Die Idee

- Zu einer gegebenen TM M mit Zustandsmenge Z und Bandalphabet Γ konstruieren wir eine TM M' mit:
 - einem neuen Zustand z_{neu}
 - für jeden Zustand $z \in Z$ von M und jedes Symbol $x \in \Gamma$, für das es keinen Übergang aus z in M gab (d.h. es gab keine Kante (z, x, X, Y, z') in M mit X, Y, z' beliebig) wird eine neue Kante (z, x, x, R, z_{neu}) hinzugefügt

Hält M , so kann M' noch einen Übergang nach z_{neu} machen. Dort hält M' . *M hält also genau dann in irgendeinem Zustand, wenn M' in z_{neu} hält.*
- Zu $\langle M, w \rangle$ kann außerdem eine TM M'' konstruiert werden, die bei jeder Eingabe das Band löscht und dann w auf M' startet, wobei M' wie eben beschreiben aus M hervorgeht.

Die Idee

- Zu einer gegebenen TM M mit Zustandsmenge Z und Bandalphabet Γ konstruieren wir eine TM M' mit:
 - einem neuen Zustand z_{neu}
 - für jeden Zustand $z \in Z$ von M und jedes Symbol $x \in \Gamma$, für das es keinen Übergang aus z in M gab (d.h. es gab keine Kante (z, x, X, Y, z') in M mit X, Y, z' beliebig) wird eine neue Kante (z, x, x, R, z_{neu}) hinzugefügtHält M , so kann M' noch einen Übergang nach z_{neu} machen. Dort hält M' . *M hält also genau dann in irgendeinem Zustand, wenn M' in z_{neu} hält.*
- Zu $\langle M, w \rangle$ kann außerdem eine TM M'' konstruiert werden, die bei jeder Eingabe das Band löscht und dann w auf M' startet, wobei M' wie eben beschreiben aus M hervorgeht.

Die Idee

Zwischenstand

- Wenn M auf w hält, dann hält M'' bei jeder Eingabe in Z_{neu} .
- Wenn M nicht auf w hält, dann besucht M'' Z_{neu} niemals.

Abschluss

UselessState = $\{\langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A\}$

Bei Eingabe $\langle M, w \rangle$

- 1 Konstruiere aus $\langle M, w \rangle$ die TM M'' .
- 2 Starte A_{US} auf $\langle M'', z_{neu} \rangle$.
- 3 Akzeptiert A_{US} , dann ist z_{neu} ein nutzloser Zustand und nach obigem hält dann M nicht auf w und wir lehnen ab.
- 4 Lehnt A_{US} ab, so ist z_{neu} nicht nutzlos, wird also besucht. Dann aber hält M auf w an und wir akzeptieren.

Dies entscheidet das Halteproblem, was nicht sein kann! A_{US} existiert also nicht! □

Abschluss

$UselessState = \{ \langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A \}$

Bei Eingabe $\langle M, w \rangle$

- 1 Konstruiere aus $\langle M, w \rangle$ die TM M'' .
- 2 Starte A_{US} auf $\langle M'', z_{neu} \rangle$.
- 3 Akzeptiert A_{US} , dann ist z_{neu} ein nutzloser Zustand und nach obigem hält dann M nicht auf w und wir lehnen ab.
- 4 Lehnt A_{US} ab, so ist z_{neu} nicht nutzlos, wird also besucht. Dann aber hält M auf w an und wir akzeptieren.

Dies entscheidet das Halteproblem, was nicht sein kann! A_{US} existiert also nicht! □

Abschluss

UselessState = $\{\langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A\}$

Bei Eingabe $\langle M, w \rangle$

- 1 Konstruiere aus $\langle M, w \rangle$ die TM M'' .
- 2 Starte A_{US} auf $\langle M'', z_{neu} \rangle$.
- 3 Akzeptiert A_{US} , dann ist z_{neu} ein nutzloser Zustand und nach obigem hält dann M nicht auf w und wir lehnen ab.
- 4 Lehnt A_{US} ab, so ist z_{neu} nicht nutzlos, wird also besucht. Dann aber hält M auf w an und wir akzeptieren.

Dies entscheidet das Halteproblem, was nicht sein kann! A_{US} existiert also nicht! □

Abschluss

$UselessState = \{ \langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A \}$

Bei Eingabe $\langle M, w \rangle$

- 1 Konstruiere aus $\langle M, w \rangle$ die TM M'' .
- 2 Starte A_{US} auf $\langle M'', z_{neu} \rangle$.
- 3 Akzeptiert A_{US} , dann ist z_{neu} ein nutzloser Zustand und nach obigem hält dann M nicht auf w und wir lehnen ab.
- 4 Lehnt A_{US} ab, so ist z_{neu} nicht nutzlos, wird also besucht. Dann aber hält M auf w an und wir akzeptieren.

Dies entscheidet das Halteproblem, was nicht sein kann! A_{US} existiert also nicht! □

Abschluss

UselessState = $\{\langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A\}$

Bei Eingabe $\langle M, w \rangle$

- 1 Konstruiere aus $\langle M, w \rangle$ die TM M'' .
- 2 Starte A_{US} auf $\langle M'', z_{neu} \rangle$.
- 3 Akzeptiert A_{US} , dann ist z_{neu} ein nutzloser Zustand und nach obigem hält dann M nicht auf w und wir lehnen ab.
- 4 Lehnt A_{US} ab, so ist z_{neu} nicht nutzlos, wird also besucht. Dann aber hält M auf w an und wir akzeptieren.

Dies entscheidet das Halteproblem, was nicht sein kann! A_{US} existiert also nicht! □

Abschluss

$UselessState = \{ \langle A, q \rangle \mid A \text{ TM, } q \text{ nutzloser Zustand von } A \}$

Bei Eingabe $\langle M, w \rangle$

- 1 Konstruiere aus $\langle M, w \rangle$ die TM M'' .
- 2 Starte A_{US} auf $\langle M'', z_{neu} \rangle$.
- 3 Akzeptiert A_{US} , dann ist z_{neu} ein nutzloser Zustand und nach obigem hält dann M nicht auf w und wir lehnen ab.
- 4 Lehnt A_{US} ab, so ist z_{neu} nicht nutzlos, wird also besucht. Dann aber hält M auf w an und wir akzeptieren.

Dies entscheidet das Halteproblem, was nicht sein kann! A_{US} existiert also nicht! □

Zusammenfassung

Wir haben gestern und heute

- entscheidbare und
- aufzählbare

Sprachen kennengelernt und gesehen, dass

- $REG \subsetneq CF \subsetneq REC \subsetneq RE$

gilt!

Ferner haben wir das Halteproblem und weitere **nicht entscheidbare** Probleme kennengelernt und eine Technik (Reduktion), wie man neue Probleme als unentscheidbar nachweisen kann.

Ausblick

Wir wissen nun, dass es viele entscheidbare Probleme und leider auch wichtige unentscheidbare Probleme gibt. Für die Praxis genügt es aber nicht, dass ein Problem entscheidbar ist! Die "Kosten" sind auch wichtig!

- Wie lange brauchen wir um das Problem zu lösen?
- Wie viel Platz/Speicher benötigen wir?

Damit beschäftigen wir uns nächste Woche...

Ausblick

Wir wissen nun, dass es viele entscheidbare Probleme und leider auch wichtige unentscheidbare Probleme gibt. Für die Praxis genügt es aber nicht, dass ein Problem entscheidbar ist! Die “Kosten” sind auch wichtig!

- Wie lange brauchen wir um das Problem zu lösen?
- Wie viel Platz/Speicher benötigen wir?

Damit beschäftigen wir uns nächste Woche...

Wiederholung

Wiederholung
(der Vorlesungen 1-11)

Überblick

Wir hatten die Sprachfamilien

- der regulären Sprachen (REG),
- der kontextfreien Sprachen (CF),
- der entscheidbaren Sprachen (REC) und
- der aufzählbaren Sprachen (RE).

Dabei gilt

- $REG \subsetneq CF \subsetneq REC \subsetneq RE$

Reguläre Sprachen - Modelle

Die Sprachfamilie der regulären Sprachen wird erfasst von:

- deterministischen, endlichen Automaten (DFAs)
- nichtdeterministischen, endlichen Automaten (NFAs)
 - mit/ohne λ -Kanten
- reguläre/rationale Ausdrücke
- (rechts-)lineare Grammatiken

Reguläre Sprachen - Begriffe

Begriffe:

- Zustände, Eingabesymbole, Alphabet, Überföhrungsfunktion, Übergangsrelation, Startzustände, Endzustände,
- vollständig, initial zusammenhängend,
- Konfiguration, Rechnung, akzeptierte Sprache,
- deterministisch, nichtdeterministisch,
- Abschlusseigenschaften

Reguläre Sprachen - Verfahren

Techniken/Verfahren:

- Beweis, dass ein Automat eine Sprache akzeptiert (zwei Richtungen zu zeigen!)
- Techniken zum Konstruieren eines Automaten
- Potenzautomatenkonstruktion
- Verschiedene Konstruktionen, um Abschlusseigenschaften zu zeigen
- Pumping Lemma der regulären Sprachen

Kontextfreie Sprachen - Modelle

Die Sprachfamilie der kontextfreien Sprachen wird erfasst von:

- nichtdeterministische Kellerautomaten
 - Akzeptanz mit leerem Keller
 - Akzeptanz mit Endzustand
- kontextfreien Grammatiken

Kontextfreie Sprachen - Begriffe

Begriffe zum PDA

- wie beim DFA/NFA

Begriffe bei Grammatiken:

- Nonterminale, Terminale, Produktionen, Startsymbol,
- Ableitung, erzeugte/generierte Sprache
- λ -Produktion

Kontextfreie Sprachen - Verfahren

Techniken/Verfahren:

- Beweis, dass eine Grammatik eine Sprache generiert (zwei Richtungen zu zeigen!)
- Beweis, dass ein Automat eine Sprache akzeptiert (zwei Richtungen zu zeigen!)
- Techniken zum Konstruieren einer Grammatik
- Techniken zum Konstruieren eines Automaten
- Herstellung der Chomsky-Normalform
 - Technik zum Reduzieren (produktiv, erreichbar)
- Verschiedene Konstruktionen, um Abschlusseigenschaften zu zeigen
- Pumping Lemma der kontextfreien Sprachen

Entscheidbare und aufzählbare Sprachen - Modelle

Die Sprachfamilie der aufzählbaren Sprachen wird erfasst von:

- Turing-Maschinen
 - deterministische (DTMs)
 - nichtdeterministische (NTMs)
 - einseitig/beidseitig unendliches Band
 - mehrere Bänder
 - ...
- jedes Modell, das die intuitiv berechenbaren Funktionen erfasst (\Rightarrow Church-Turing-These)

Entscheidbare und aufzählbare Sprachen - Begriffe

Begriffe:

- Zustände, Eingabesymbole, Bandsymbole, Überföhrungsfunktion, Überföhrungsrelation, Startzustand, Endzustände, Symbol für das leere Feld,
- Konfiguration
- Schrittrelation, Rechnung, Erfolgsrechnung, akzeptierte Sprache
- (Turing-)berechenbar
- aufzählbar, entscheidbar

Entscheidbare und aufzählbare Sprachen - Verfahren

Techniken/Verfahren:

- TM konstruieren, die eine Sprache akzeptiert
- TM konstruieren, die eine Funktion berechnet
- TM konstruieren, die eine Sprache entscheidet
- Beweisen, dass ein Problem unentscheidbar ist

Was wir ausgelassen haben...

Ausgelassen haben wir...

- deterministische Kellerautomaten (DPDAs)
 - Akzeptanz mit leerem Keller
 - Akzeptanz mit Endzustand
- linear beschränkte Automaten (LBAs)
- kontextsensitive Grammatiken (Typ-1)