1

DTM
Deterministische Turing-Maschine

2

NTM
Nichtdeterministische Turing-Maschine

3

Entscheidungsproblem

4

(Un-)Entscheidbarkeit

5

Semi-Entscheidbarkeit

6

Äquivalente Aussagen

7

Co-Semi-Entscheidbarkeit

8

Aufzählbarkeit

2 *Antwort*

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$
Q...Zustandsmenge
Σ...Eingabealphabet
Γ...Bandalphabet mit $\Gamma \subseteq \Sigma \cup \{\text{␣}\}$
δ...Übergangsfkt. $Q \times \Gamma \to 2\hat{}(Q \times \Gamma \times \{L, R, N\})$
q_0...Startzustand $q_0 \in Q$
F...akzeptierende Endzustände $F \subseteq Q$

1 *Antwort*

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$
Q...Zustandsmenge
Σ...Eingabealphabet
Γ...Bandalphabet mit $\Gamma \subseteq \Sigma \cup \{\text{␣}\}$
δ...Übergangsfkt. $Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$
q_0...Startzustand $q_0 \in Q$
F...akzeptierende Endzustände $F \subseteq Q$

4 *Antwort*

Eine Eigenschaft auf einer Menge heißt entscheidbar (auch rekursiv, rekursiv ableitbar), wenn es ein Entscheidungsverfahren für sie gibt. Ein Entscheidungsverfahren ist ein Algorithmus, der für jedes Element der Menge beantworten kann, ob es die Eigenschaft hat oder nicht. Wenn es kein solches Entscheidungsverfahren gibt, dann nennt man die Eigenschaft unentscheidbar.

3 *Antwort*

Als Entscheidungsproblem bezeichnet man die Frage, ob und wie für eine gegebene Eigenschaft ein Entscheidungsverfahren formuliert werden kann.

6 *Antwort*

- M ist rekursiv aufzählbar.
- M ist semi-entscheidbar.
- M ist vom Chomsky-Typ 0.
- M ist die Menge aller Berechnungsergebnisse einer Turing-Maschine.
- M ist Definitionsbereich einer berechenbaren Funktion.
- M ist Wertebereich einer berechenbaren Funktion.
- M ist endlich oder Wertebereich einer injektiven berechenbaren Funktion.
- M liegt in der Klasse Σ_1^0 der arithmetischen Hierarchie.
- M lässt sich many-one auf das Halteproblem reduzieren.

5 *Antwort*

Eine Menge M heiße semi-entscheidbar, wenn die partielle charakteristische Funktion $x_M : \mathbb{N} \to \{1\}$ von M berechenbar ist.

8 *Antwort*

Eigenschaft einer Menge, dass es eine "Generatorfunktion" gibt, die alle Elemente aufzählt

7 *Antwort*

Ob den Elementen einer Menge, die die Eigenschaft nicht haben, das Gegenteil der Eigenschaft eindeutig nachgewiesen werden kann.

9

Abzählbarkeit

10

Überabzählbarkeit

11

Halteproblem

12

Cantor-Funktion

13

Cantor-Diagonalisierung

14

Cantors erstes Diagonalargument

15

Cantors zweites Diagonalargument

16

Cantorsche Paarungsfunktion

10 *Antwort*

Eigenschaft einer Menge, nicht abzählbar zu sein (keine Bijektion auf $\mathbb{N}$)

9 *Antwort*

Menge, die die gleiche Mächtigkeit wie $\mathbb{N}$ hat (eindimensional unendlich bzw abzählbar unendlich)

12 *Antwort*

Die Verteilungsfunktion der Cantorverteilung

11 *Antwort*

Frage, ob eine Maschine (zB eine TM) auf einer bestimmten Eingabe hält (oder in eine Endlosschleife geht). Ist unentscheidbar (semi-, nicht co-semi-), NP-hart

14 *Antwort*

Die Mächtigkeit zweier Mengen A und B ist genau gleich, wenn eine Bijektion zwischen A und B gibt

13 *Antwort*

Bezeichung der von Cantor entwickelten Diagonalverfahren

16 *Antwort*

Basiert auf dem Diagonalargument von Cantor ($\mathbb{N}\times\mathbb{N} \to \mathbb{N}$)

15 *Antwort*

sei r_i: $r_1 = 0, b_{11}b_{12}b_{13}...$
$r_1 = 0, b_{21}b_{22}b_{23}...$
$r_1 = 0, b_{31}b_{32}b_{33}...$
$\bar{r} = 0, \bar{r}_{11}\bar{r}_{22}\bar{r}_{33}...$
$\bar{r}$ ist dann nicht in der Menge von r_i

17

Ackermannfunktion

18

Topologie

19

Gödelsche Unvollständigkeitssätze

20

LOOP-Programm: Definition

21

LOOP-Programm: ADD-Funktion

22

LOOP-Programm: SUB-Funktion

23

LOOP-Programm: MUL-Funktion

24

LOOP-Programm: POT-Funktion

18 *Antwort*

tbd

17 *Antwort*

Funktion der Form: $\varphi(a,b,0) = a+b$
$\varphi(a,0,n+1) = \alpha(a,n)$
$\varphi(a,b+1,n+1) = \varphi(a,\varphi(a,b,n+1),n)$ oder ähnlich mit extrem schnellem Wachstum

20 *Antwort*

P ist LOOP Programm, wenn von der Form:
$x_i := x_j + n$,
$x_i := x_j - n$,
$LOOP x_i DO P_j END$,
$p_i; p_j$

19 *Antwort*

Die Gödelschen Unvollständigkeitssätze weisen nach das es in hinreichend starken Systemen, Aussagen geben muss die man weder formal beweisen noch widerlegen kann. Es gibt den ersten und den 2. Unvollständigkeitssatz

22 *Antwort*

SUB x_1 x_2:
x_0 := x_1 + 0;
LOOP x_2 DO x_0 := x_0 − 1 END

21 *Antwort*

ADD x_1 x_2:
x_0 := x_1 + 0;
LOOP x_2 DO x_0 := x_0 + 1 END

24 *Antwort*

POT x_1 x_2:
x_0 := x_1 + 0;
LOOP x_2 DO MUL x_0 x_1 END

23 *Antwort*

MUL x_1 x_2:
x_0 := x_1 + 0;
LOOP x_2 DO ADD x_0 x_1 END

25

LOOP-Programm: DIV-Funktion

26

LOOP-Programm: MAX-Funktion

27

LOOP-Programm: MIN-Funktion

28

LOOP-Programm: MOD-Funktion

29

LOOP-Programm: KGV-Funktion

30

LOOP-Programm: GGT-Funktion

31

LOOP-Programm: GGT-Funktion
(in Abhängigkeit von KGV)

32

LOOP-Programm: KGV-Funktion
(in Abhängigkeit von GGT)

26 *Antwort*

MAX x_1 x_2:
x_0 := x_1 + 0;
SUB x_0 x_2;
ADD x_0 x_2

25 *Antwort*

DIV x_1 x_2:
x_0 := x_1;
LOOP x_1 DO:
  x_3 := x_0 + 0;
  MUL x_3 x_2;
  SUB x_3 x_1;
  IF x_1 != 0 THEN x_0 := x_0 − 1 END
END

28 *Antwort*

MOD x_1 x_2:
x_1 := x_1 + 1;
LOOP x_2 DO:
  LOOP x_1 DO x_0 := x_1 + 0 END;
  SUB x_1 x_2
END;
x_1 := x_1 − 1

27 *Antwort*

MIN x_1 x_2:
x_0 := x_1 + 0;
MAX x_1 x_2;
ADD x_0 x_2;
SUB x_0 x_1

(beide aufaddieren, davon das Maximum abziehen)

30 *Antwort*

GGT x_1 x_2:
x_0 := x_1 − 1;
x_3 := x_1 + 0;
MUL x_3 x_2;
LOOP x_1 DO:
  x_4 := x_0 + 0;
  MOD x_4 x_1;
  IF x_4 != 0 THEN x_0 := x_0 − 1 END
  x_4 := x_0 + 0;
  MOD x_4 x_2;
  IF x_4 != 0 THEN x_0 := x_0 − 1 END
END

29 *Antwort*

KGV x_1 x_2:
x_0 := x_1 + 1;
x_3 := x_1 + 0;
MUL x_3 x_2;
LOOP x_3 DO:
  x_4 := x_0 + 0;
  MOD x_4 x_2;
  IF x_4 != 0 THEN ADD x_0 x_1 END
END

32 *Antwort*

KGV x_1 x_2:
x_0 := x_1 + 0;
MUL x_0 x_2;
x_3 := x_1 + 0;
GGT x_3 x_2;
DIV x_0 x_3

31 *Antwort*

GGT x_1 x_2:
x_0 := x_1 + 0;
MUL x_0 x_2;
x_3 := x_1 + 0;
KGV x_3 x_2;
DIV x_0 x_3

33

LOOP-Programm: Fallunterscheidung (IF)

34

WHILE-Programm

35

Kolmogorov-Komplexität

36

Many-One-Reduktion

37

Schubfachprinzip

38

Satz von Rice

39

PKP oder PCP
Postsches Korrespondenzproblem

40

Äquivalenzproblem

34 *Antwort*

P ::= x_i := x_j + c
P ::= x_i := x_j − c
P ::= P; P
P ::= LOOP x_i DO P END
P ::= WHILE $x_i \neq 0$ DO P END

33 *Antwort*

IF x_0 != 0 THEN P END:
LOOP x_0 DO x_1 := 1 END;
LOOP x_1 DO P END

36 *Antwort*

Problem A ist auf B many-one-reduzierbar ($A \leq_m B$), falls es eine berechenbare Funktion $f : A \to B$ gibt.

35 *Antwort*

Maß für die Strukturiertheit einer Zeichenkette. Gegeben durch die Länge des kürzesten Programms, das diese Zeichenkette erzeugt.

38 *Antwort*

Es ist unmöglich, eine beliebige, nicht-triviale Eigenschaft der erzeugten Funktion einer Turing-Maschine algorithmisch zu entscheiden.
Trivial wäre "immer akzeptieren" oder "immer verwerfen".

37 *Antwort*

Falls man n Objekte auf m Mengen ($n, m > 0$) verteilt und $n > m$ gilt, gibt es mindestens eine Menge, die mehr als 1 Objekt enthält. Auch: Taubenschlagprinzip, Dirichlet-Prinzip.

40 *Antwort*

Das Problem, zu entscheiden, ob zwei formale Definitionen von zwei Sprachen L_1 und L_2 äquivalent sind, also $L_1 = L_2$ gilt.
Die Sprachen können durch Grammatiken, Automaten oder ganz anders definiert sein.

39 *Antwort*

Beispiel für ein unentscheidbares Problem.

41

Komplexitätsklassen und deren Beziehungen

42

P, NP, coNP

43

PSPACE, EXPTIME, NEXPTIME

44

EXPSPACE, NEXPSPACE

45

(P,NP,PSPACE)-hart

46

(P,NP,PSPACE)-vollständig

47

Wortproblem Deterministischer Endlicher Automaten

48

SAT
Erfüllbarkeitsproblem (der Aussagenlogik)

42 *Antwort*

- P, NP, coNP sind Komplexitätsklassen.
- P (auch PTIME) enthält die Entscheidungsprobleme, die in Polynomialzeit durch DTM lösbar sind. (Klasse der "praktisch lösbaren" Probleme)
- NP enthält die Entscheidungsprobleme (die von einer NTM in Polynomialzeit gelöst werden können), bei denen es für positive Antworten Beweise (Zertifikate) gibt, die in Polynomialzeit verifiziert werden können.

41 *Antwort*

- P, NP, coNP, PSPACE, EXPTIME, NEXPTIME, EXPSPACE, NEXPSPACE sind Komplexitätsklassen.
- Beziehungen:
  - P $\subseteq$ NP $\subseteq$ PSPACE $\subseteq$ EXPTIME
  - EXPTIME $\subseteq$ NEXPTIME $\subseteq$ EXPSPACE
  - EXPSPACE = NEXPSPACE
  - P $\subset$ EXPTIME
  - P = coP dh P = NP $\Leftrightarrow$ NP = coNP

44 *Antwort*

- EXPSPACE, NEXPSPACE sind Komplexitätsklassen.
- EXPSPACE enthält die Entscheidungsprobleme, die von einer DTM mit durch $O(2^{p(n)})$ beschränktem Platz (in beliebig langer Zeit) entschieden werden können.
- NEXPSPACE enthält die Entscheidungsprobleme, die von einer NTM mit durch $O(2^{p(n)})$ beschränktem Platz (in beliebig langer Zeit) entschieden werden können. Es gilt NEXPSPACE = EXPSPACE.

43 *Antwort*

- PSPACE, EXPTIME, NEXPTIME sind Komplexitätsklassen.
- PSPACE enthält die Entscheidungsprobleme, die von einer DTM mit polynomiellem Platz (in beliebig langer Zeit) entschieden werden können.
- EXPTIME enthält die Entscheidungsprobleme, die von einer DTM mit durch $O(2^{p(n)})$ beschränkter Zeit entschieden werden können.
- NEXPTIME enthält die Entscheidungsprobleme, die von einer NTM mit durch $O(2^{p(n)})$ beschränkter Zeit entschieden werden können.

46 *Antwort*

P ist NP-vollständig, wenn es in NP liegt und NP-hart ist. Dh alle Probleme in NP sind polynomiell reduzierbar auf P und P ist polynomiell reduzierbar auf alle anderen NP-harten Probleme.

45 *Antwort*

Auch als (P,NP,PSPACE)-Schwere bezeichnet.
Die formale Sprache $L' \subseteq \Sigma*$ ist NP-hart, wenn gilt:
$\forall L \in \text{NP} : L \leq_p L'$
(alle L aus NP sind polynomiell reduzierbar auf L')

48 *Antwort*

Entscheidungsproblem, ob eine aussagenlogische Formel erfüllbar ist. SAT ist NP-vollständig.

47 *Antwort*

Das Wortproblem ist das Entscheidungsproblem, ob ein gegebenes Wort zu einer Sprache gehört, oder nicht. Ist (semi-)Entscheidbar, wenn die Sprache dies auch ist.

49

Kleene-Stern

50

Liste einiger P-vollständigen Probleme

51

Liste einiger NP-vollständigen Probleme

52

Liste einiger NP-vollständigen Probleme

53

Liste einiger PSPACE-vollständigen Probleme

54

Liste einiger EXPTIME-vollständigen Probleme

55

Formalisieren (Ablauf)

56

Cliquenproblem (CLIQUE)

\# 50 *Antwort*

tbd

\# 49 *Antwort*

Die kleenesche Hülle (auch endlicher Abschluss, Kleene-*-Abschluss, Verkettungshülle oder Sternhülle genannt) eines Alphabets Σ oder einer formalen Sprache L ist die Menge aller Wörter, die durch beliebige Konkatenation (Verknüpfung) von Symbolen des Alphabets Σ bzw. von Wörtern der Sprache L gebildet werden können, wobei das leere Wort ε inbegriffen ist.

\# 52 *Antwort*

- 3-SAT
- Kantenfärbungsproblem (CHROMATIC NUMBER)
- Problem der exakten Überdeckung (EXACT COVER)
- Steinerbaumproblem (STEINER TREE)
- Hitting-Set-Problem (HITTING SET)
- Rucksackproblem (KNAPSACK)
- Partitionsproblem (PARTITION)
- Maximaler Schnitt (MAX CUT)

\# 51 *Antwort*

- Erfüllbarkeitsproblem (SAT)
- Cliquenproblem (CLIQUE)
- Mengenpackungsproblem (SET PACKING)
- Knotenüberdeckungsproblem (VERTEX COVER)
- Mengenüberdeckungsproblem (SET COVERING)
- (un)gerichtetes Hamiltonkreisproblem ((UN)DIRECTED HAMILTONIAN CIRCUIT)

\# 54 *Antwort*

- Dame
- Go
- Schach
- Shogi

\# 53 *Antwort*

- hex
- Go-Moku
- Reversi

\# 56 *Antwort*

Entscheidungsproblem der Graphentheorie.
Problem: Gibt es zu einem Graphen G und einer Zahl n eine Clique der Mindestgröße n in G?
Eine Clique ist ein Teilgraph, dessen Knoten alle direkt miteinander verbunden sind.

\# 55 *Antwort*

tbd

57

Mengenpackungsproblem (SET PACKING)

58

Knotenüberdeckungsproblem (VERTEX COVER)

59

Mengenüberdeckungsproblem (SET COVER)

60

3SAT

61

QBF

62

LBA
Linear Bounded Automaton
Linear beschränkte Turingmaschine

63

Pränexform

64

Skolemform

58 *Antwort*

Entscheidungsproblem der Graphentheorie.
Problem: Für einen einfachen Graphen G (ungerichtet, ohne Mehrfachkanten oder Schleifen) und eine Zahl $k \in \mathbb{N}$ prüfe, ob es eine Teilmenge U der Knoten mit $|U| \leq k$ gibt, sodass jede Kante von G mit mindestens einem Knoten aus U verbunden ist.

57 *Antwort*

Entscheidungsproblem der Kombinatorik.
Problem: Für eine endliche Menge U und n Teilmengen S_j $(0 \leq j \leq n)$ von U: existieren mindestens $k \leq n$ paarweise disjunkte Teilmengen S_j $(0 \leq j \leq k)$ von U?
Visualisierung: Es existieren n Werkzeugkästen mit unterschiedlichem Inhalt. Es ist eine Teilmenge der Werkzeugkästen gesucht, in der jedes Werkzeug exakt einmal vorhanden ist.

60 *Antwort*

Variante von SAT (Erfüllbarkeitsproblem der Aussagenlogik).
Problem: Ist eine aussagenlogische Formel F, die in konjunktiver Normalform mit höchstens 3 Literalen pro Klausel gegeben ist, erfüllbar?
z.B.: $F = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2)$

59 *Antwort*

Entscheidungsproblem der Kombinatorik.
Problem: Für eine Menge U und n Teilmengen S_j $(0 \leq j \leq n)$ von U und einer Zahl $k \leq n$ mit $k, n \in \mathbb{N}$ prüfe, ob eine Vereinigung von k oder weniger Teilmengen S_j $(0 \leq j \leq k)$ existiert, die der Menge U entspricht.

62 *Antwort*

TM, die den Bereich des Bandes, auf dem die Eingabe steht, nicht verlässt (bzw verlassen darf).
Um ein größeres Band zu simulieren, kann das Bandalphabet mehr Elemente (zB Tupel) erhalten.

61 *Antwort*

tbd

64 *Antwort*

Formel, die in der Normalform nach Alber Thoralf Skolem ist.
Hilfreich zur Prüfung der Erfüllbarkeit. Eine Formelmenge ist genau dann erfüllbar, wenn ihre Skolemform erfüllbar ist. In anderen Aspekten stimmt eine Formel nicht zwangsweise mit ihrer Skolemform überein.

63 *Antwort*

Mögliche Normalform, in der Aussagen der Prädikatenlogik dargestellt werden können.
Benötigt als Vorstufe zur Skolemform.
Pränexform ist genau dann erfüllt, wenn alle Quantoren außerhalb bzw. vor der eigentlichen Formel stehen.
Umformung in Pränexform: bereinigen (dabei werden ggf wegen Negationsoperationen Quantoren negiert), Quantoren an den Anfang schieben.

65

Klauselform
Klauselnormalform

66

$\models$
Schlussfolgerung, Inferenz

67

Resolutionsverfahren
(Aussagenlogik)

68

Resolutionsverfahren
(Prädikatenlogik)

69

Resolvent(e)

70

Unifikator

71

Allgemeinster Unifikator

72

Herbrand-Universum

65 *Antwort*

Formel in konjunktiver Normalform, bei der die Konjunktionen in Mengenschreibweise zusammengefasst werden.
Eine Formal in Klauselform ist eine logische Verknüpfung von Literalen, die als disjunkte Normalform oder konjunktive Normalform notiert ist.

66 *Antwort*

tbd

67 *Antwort*

tbd

68 *Antwort*

tbd

69 *Antwort*

tbd

70 *Antwort*

Eine Substitution S heißt eine Vereinheitlichung (Unifikator) der Literale $L_1, L_2, ..., L_m$, wenn durch die Anwendung von S die Argumente aller Literale zur Übereinstimmung gebracht werden, d.h. wenn $S(L_1) = S(L_2) = \cdots = S(L_m)$

71 *Antwort*

tbd

72 *Antwort*

tbd

73

Herbrand-Modell

74

Herbrand-Expansion

74 *Antwort*

tbd

73 *Antwort*

tbd