Lösungen zur Aufgabensammlung

Dieses Dokument enthält (1) alle relevanten Ergänzungsaufgaben der letzten zwei Jahre, (2) Aufgaben der letzten drei Prüfungen und (3) einige frisch ausgedachten Aufgaben.

Die Aufgaben sind nach Themem sortiert und wurden mit Musterlösungen versehen. Bei den Lösungen zu den Aufgaben aus (1) wird auf die entsprechenden Ergänzungen verwiesen, damit man sich bei Bedarf Aufzeichnungen dazu anschauen kann. Diese werden bis zur Prüfung online bleiben.

Alte Prüfungen sind unter

`www.fmi.uni-stuttgart.de/ti/teaching/exams`

zu finden.

Mehr Aufgaben stehen auf den Vorlesungswebseiten der letzten Jahre zur Verfügung. Die Hausaufgaben vom Sommersemester 2018 wurden in den Ergänzungen 4, 6, 8, 10, 12, 14 und 15 (ebenfalls vom Sommersemester 2018) diskutiert.

Diese Datei wird, wie alle anderen auf der Ergänzungswebseite, laufend verbessert. Es werden jedoch keine Aufgaben hinzugefügt oder gestrichen, sodass die Nummerierung unverändert bleiben wird.

Viel Erfolg bei der Prüfungsvorbereitung!

Fehlermeldungen bitte an `camino@fmi.uni-stuttgart.de`. Vielen Dank!

Inhaltsverzeichnis

Turingmaschinen

(Auffrischung aus Theoretische Informatik I)

Aufgabe 1

Seien $m \in \mathbb{N} \setminus \{0\}$ eine natürliche Zahl, $\Sigma = \{a_1, \ldots, a_m\}$ ein m-elementiges Alphabet und L folgende Sprache über Σ:

$$L = \{w \in \Sigma^* \mid \text{in } w \text{ kommt jeder Buchstabe aus } \Sigma \text{ vor}\}.$$

Geben Sie eine DTM M für L an.

1. M darf den Leseschreibkopf nicht nach links bewegen.
2. M darf höchstens $2m$ Zustände haben.
3. M darf höchstens $m+2$ Zustände und höchstens $m+2$ Bandsymbole haben.

Lösung

1. $M = (\mathcal{P}(\Sigma), \Sigma, \Sigma \cup \{\square\}, \delta, \emptyset, \square, \{\Sigma\})$ mit

$$\delta(q, x) = \begin{cases} (q, \square, N) & \text{für } x = \square \\ (q \cup \{x\}, x, R) & \text{für } x \neq \square. \end{cases}$$

2. $M = (\{s_1, \ldots, s_m, f_1, \ldots, f_m\}, \Sigma, \Sigma \cup \{\Box\}, \delta, s_1, \Box, \{f_m\})$ mit

$$\delta(s_i, a_j) = \begin{cases} (f_i, a_j, N) & \text{für } i = j \\ (s_i, a_j, R) & \text{für } i \neq j \end{cases} \qquad \delta(s_i, \Box) = (s_i, \Box, N)$$

$$\delta(f_i, a_j) = (f_i, a_j, L) \qquad \delta(f_i, \Box) = \begin{cases} (f_i, \Box, N) & \text{für } i = m \\ (s_{i+1}, \Box, R) & \text{für } i \neq m \end{cases}$$

   für alle $i, j \in \{1, \ldots, m\}$.

3. (Siehe Ergänzungsblatt 1, Aufgabe 1.)

Aufgabe 2

Geben Sie für jede der folgenden Sprachen L eine DTM an.

1. $L = \{a^k \mid k \text{ ist eine Zweierpotenz}\}$
2. $L = \{a^k \mid k \text{ ist eine Quadratzahl}\}$

Lösung

Bitte Lösungen mit einem Turingmaschinensimulator selber testen!

Aufgabe 3

Seien $a_1, \ldots, a_n$ beliebige Buchstaben aus einem Alphabet Σ. Geben Sie in Abhängigkeit der a_i eine DTM M an, die nur das Wort $w = a_1 \ldots a_n$ akzeptiert, d. h. : $T(M) = \{w\}$.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 2)

Aussagen- und Prädikatenlogik

(Vorlesungseinheiten 1-4)

Aufgabe 4

Eine aussagenlogische Formel ist in *kanonischer DNF* (bzw. *kanonischer KNF*), wenn sie in DNF (bzw. KNF) ist und jede Klausel jede Variable aus der Formel genau einmal enthält.

Sei $F = ((A \to B) \leftrightarrow (\neg B \vee A)) \wedge C$.

1. Geben Sie eine zu F äquivalente Formel in kanonischer DNF an.
2. Geben Sie eine zu F äquivalente Formel in kanonischer KNF an.

Lösung

Bis auf Umordnung der Klauseln und Literalen sind die Antworten eindeutig.

1. $(\neg A \wedge \neg B \wedge C) \vee (A \wedge B \wedge C)$.
2. $(A \vee B \vee C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee C)$.

Aufgabe 5

Gegeben seien folgende prädikatenlogische Formeln:

$$\begin{array}{lll} F_1 = \forall x \exists y P(x,y), & F_4 = \forall x \neg P(x, f(x)), & F_7 = \forall x (\exists y P(y,x) \rightarrow Q(x)), \\ F_2 = \forall x \neg P(x,x), & F_5 = \forall x \forall y (P(y, f(x)) \rightarrow P(x,y)), & F_8 = \neg Q(f(f(x))) \\ F_3 = \exists y \forall x \neg P(x,y), & F_6 = P(a, f(f(a))), & F_9 = P(f(y), y). \end{array}$$

Geben Sie ein Modell mit möglichst kleinem Universum für $F = \bigwedge_{i=1}^{9} F_i$ an.

Lösung

(Siehe Ergänzungsblatt 1, Aufgabe 2.)

Aufgabe 6

Entscheiden Sie für jede der folgenden Formeln, ob diese erfüllbar ist oder nicht. Geben Sie ein Modell mit möglichst kleinem Universum an oder begründen Sie, warum sie kein Modell besitzt.

1. $F = P(a,b) \wedge$
   $\forall x \Big(\forall y \forall z \big(\big(P(x,y) \wedge P(y,z)\big) \rightarrow P(x,z) \big) \wedge \exists y P(x,y) \wedge \forall y \big(P(x,y) \rightarrow \neg P(y,x)\big) \Big)$
2. $F = \forall x \Big(\big(P(x) \rightarrow \forall y Q(x,y)\big) \wedge \neg Q(x,x) \Big) \wedge \exists x P(x)$
3. $F = P\big(f(a), a\big) \wedge \forall x \Big(\neg P(x,x) \wedge P\big(f(x), x\big) \wedge \forall y \forall z \Big(\big(P(x,y) \wedge P(y,z)\big) \rightarrow P(x,z) \Big) \Big)$

Lösung

1. (Siehe Ergänzungsblatt 1, Aufgabe 3.1.)
2. (Siehe Ergänzungsblatt 1, Aufgabe 3.2.)
3. F ist erfüllbar, besitzt aber kein endliches Modell. Ein abzählbar unendliches Modell ist $\mathcal{A} = (\mathbb{N}, I)$ mit $I(a) = 0$, $I(P) = \{(m,n) \in \mathbb{N}^2 \mid m > n\}$ und $I(f)\colon \mathbb{N} \rightarrow \mathbb{N}$, $n \mapsto n+1$.

Turing-, LOOP-, GOTO- und WHILE-Berechenbarkeit

(Vorlesungseinheiten 5-9)

Aufgabe 7

Welche der folgenden Gleichungen gelten für beliebige $x, y, z \in \mathbb{N}$? Beweisen Sie Ihre Antworten.

1. $x \dot{-} (y \dot{-} z) = x \dot{-} y + z$
2. $(x \dot{-} y) \dot{-} z = x \dot{-} (y + z)$
3. $x + y \dot{-} z = x + (y \dot{-} z)$
4. $x(y \dot{-} z) = xy \dot{-} xz$
5. $(x \dot{-} y)^2 = x^2 \dot{-} 2xy + y^2$
6. $(x + y)(x \dot{-} y) = x^2 \dot{-} y^2$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 4, Aufgabe 2)

Aufgabe 8

Was muss für $x, y, z \in \mathbb{N}$ gelten, damit die Gleichung

$$x + y \dot{-} z = x + (y \dot{-} z)$$

erfüllt ist?

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 4, Aufgabe 3)

Aufgabe 9

Geben Sie eine DTM M an, die die Nachfolgerfunktion

$$s \colon \mathbb{N} \to \mathbb{N}, n \mapsto n + 1$$

berechnet.

Lösung

(Siehe Ergänzungsblatt 2, Aufgabe 1.)

Aufgabe 10

Welche der folgenden Aussagen gelten für beliebige Funktionen $f, g \colon \mathbb{N} \to \mathbb{N}$?

1. Wenn f und g Turing-berechenbar sind, dann ist es auch $f + g$.
2. Wenn f Turing-berechenbar ist und g nicht, dann ist $f+g$ nicht Turing-berechenbar.
3. Wenn f und g nicht Turing-berechenbar sind, dann ist es auch $f + g$ nicht.

Hinweis: $f + g$ ist ebenfalls eine Funktion mit $f + g \colon \mathbb{N} \to \mathbb{N}$, $n \mapsto f(n) + g(n)$.

Lösung

1. Die Aussage ist richtig. Seien M_f und M_g DTMs, die f und g berechnen. Dann ist es einfach eine 2-Band-DTM zu konstruieren, die die Eingabe zuerst auf das zweite Band kopiert, sich danach auf Band 1 wie M_f und auf Band 2 wie M_g verhält und am Ende die Binärzahlen auf beiden Bändern zusammenaddiert und als Ergebnis ausgibt.
2. Die Aussage ist richtig. Wäre $f + g$ berechenbar, dann wäre g mittels $g(n) = (f + g)(n) - f(n)$ ebenfalls berechenbar.
3. Sei $h\colon \mathbb{N} \to \{0, 1\}$ eine beliebige nicht Turing-berechenbare Funktion (diese existiert aus Abzählbarkeitsgründen). Dann sind f, g mit $f(n) = h(n)$ und $g(n) = 1 - h(n)$ beide nicht Turing-berechenbar, aber $f + g$ wegen $(f + g)(n) = 1$ schon.

Aufgabe 11

Welche der folgenden Aussagen gelten für beliebige Funktionen $f, g\colon \mathbb{N} \to \mathbb{N}$?

1. Wenn f und g Turing-berechenbar sind, dann ist auch $f \circ g$ Turing-berechenbar.
2. Wenn f und $f \circ g$ Turing-berechenbar sind, dann ist auch g Turing-berechenbar.
3. Wenn g und $f \circ g$ Turing-berechenbar sind, dann ist auch f Turing-berechenbar.

Hierbei bezeichne $\circ$ die Komposition von Funktionen.

Lösung

(Siehe Ergänzungsblatt 2, Aufgabe 6.)

Aufgabe 12

Geben Sie für jede der folgenden Funktionen ein LOOP-Programm an, das die jeweilige Funktion berechnet.

1. $\max\colon \mathbb{N}^2 \to \mathbb{N},\ (m, n) \mapsto \begin{cases} n, & \text{falls } m \leq n \\ m & \text{sonst} \end{cases}$
2. $\mathrm{square}\colon \mathbb{N} \to \mathbb{N},\ n \mapsto n^2$
3. $\mathrm{twopow}\colon \mathbb{N} \to \mathbb{N},\ n \mapsto 2^n$

Lösung

1. (Siehe Ergänzungsblatt 2, Aufgabe 2.)
2. `LOOP` x_1 `DO`
    `LOOP` x_1 `DO`
        $x_0 := x_0 + 1$
    `END`
`END`
3. $x_0 := x_0 + 1$;
`LOOP` x_1 `DO`

```
    LOOP $x_0$ DO
      $x_0 := x_0 + 1$
    END
  END
```

Aufgabe 13

Zeigen Sie, dass folgende Anweisungen durch ein LOOP-Programm P simuliert werden können.

1. $x_i := x_j * x_k$
2. IF $x_i = 0$ THEN P END

Dabei sind x_i, x_j und x_k beliebige Variablen und P ein beliebiges LOOP-Programm.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 2, Aufgabe 2)

Aufgabe 14

Zeigen Sie, dass folgende Funktionen LOOP-berechenbar sind.

1. $\mathrm{sqrt}\colon \mathbb{N} \to \mathbb{N},\ n \mapsto \lfloor\sqrt{n}\rfloor$
2. $d\colon \mathbb{N} \to \mathbb{N},\ n \mapsto |\,\{m \leq n \,|\, m \text{ teilt } n\}\,|$

Hinweis: Für $m, n \in \mathbb{N}$ gilt: m teilt n, falls ein $k \in \mathbb{N}$ existiert mit $n = km$.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 2, Aufgabe 3)

Aufgabe 15

Wir erweitern LOOP-Programme um Anweisungen `HALT` und `ERROR`, die die Ausführung des Programms sofort abbrechen. Bei `HALT` ist der Rückgabewert der aktuelle Wert der Variable x_0. Bei `ERROR` ist das Resultat der Berechnung undefiniert.

1. Gibt es für jedes LOOP-Programm mit `HALT`-Anweisungen ein äquivalentes LOOP-Programm ohne `HALT`-Anweisungen?
2. Gibt es für jedes LOOP-Programm mit `ERROR`-Anweisungen ein äquivalentes LOOP-Programm ohne `ERROR`-Anweisungen?

Lösung

(Siehe Ergänzungsblatt 2, Aufgabe 4.)

Aufgabe 16

Welche der folgenden Aussagen gelten für beliebige Funktionen $f, g\colon \mathbb{N} \to \mathbb{N}$ mit $g(n) = f(n) \bmod 2$? Begründen Sie Ihre Antworten.

1. Wenn f Turing-berechenbar ist, dann ist es g auch.
2. Wenn f nicht Turing-berechenbar ist, dann ist es g auch nicht.
3. g ist Turing-berechenbar, egal ob f es ist oder nicht.

Lösung

Aus Abzählbarkeitsgründen wissen wir, dass eine nicht Turing-berechenbare Funktion $h\colon \mathbb{N} \to \{0, 1\}$ existiert.

1. Die Aussage ist richtig. Für eine gegebene DTM M, die f berechnet kann man eine DTM konstruieren, die zuerst M auf die Eingabe ausführt und danach alle Zeichen bis auf das rechteste löscht. Diese berechnet g.
2. Die Aussage ist falsch. Für $f\colon \mathbb{N} \to \mathbb{N}$ mit $f(n) = 2h(n)$ ist $g(n) = 0$ Turing-berechenbar, aber f nicht (sonst könnte man h mittels $h(n) = f(n)/2$ berechnen).
3. Die Aussage ist falsch. Für $f = h$ gilt $g = h$.

Aufgabe 17

Wir betrachten die ganzzahlige Division

$$\operatorname{div}\colon \mathbb{N}^2 \to \mathbb{N}, (m, n) \mapsto \left\lfloor \frac{m}{n} \right\rfloor,$$

wobei $\operatorname{div}(m, n)$ genau dann für $(m, n) \in \mathbb{N}^2$ definiert ist, wenn $n \neq 0$ gilt.

1. Geben Sie ein WHILE-Programm P an, das div berechnet.
2. Geben Sie ein GOTO-Programm P an, das div berechnet.
3. Ist div
   (a) Turing-berechenbar?
   (b) LOOP-berechenbar?
4. Sei $f\colon \mathbb{N} \to \mathbb{N}$ eine Funktion, die nicht Turing-berechenbar ist. Für welche $c \in \mathbb{N}$ ist $cf\colon \mathbb{N} \to \mathbb{N}, n \mapsto cn$ Turing-berechenbar?

Lösung

(Siehe Ergänzungsblatt 2, Aufgabe 3.)

Aufgabe 18

Wir erweitern GOTO-Programme um variable Sprünge. Beim Erreichen einer Anweisung der Form `GOTO` x_i, wobei x_i eine Programmvariable ist, soll zur Marke M_{x_i} gesprungen werden. Falls die adressierte Marke nicht existiert, ist das Resultat der Berechnung nicht definiert.

Gibt es für jedes GOTO-Programm mit variablen Sprüngen ein äquivalentes GOTO-Programm ohne variable Sprünge?

Lösung

(Siehe Ergänzungsblatt 2, Aufgabe 5.)

Primitive Rekursion

(Vorlesungseinheiten 10-11)

Aufgabe 19

Gegeben sei die Funktion

$$p = \mathrm{rec}(c_1^1, \mathrm{mult}(\pi_1^3, \pi_3^3))(\pi_2^2, \pi_1^2).$$

Geben Sie eine möglichst einfache Zuordnungsvorschfrift für p an.

Lösung

(Siehe Ergänzungsblatt 3, Aufgabe 1.)

Aufgabe 20

Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind, indem Sie für jede Funktion einen definierenden, primitiv rekursiven Ausdruck angeben.

1. $\mathrm{zero}\colon \mathbb{N} \to \mathbb{N},\ x \mapsto \begin{cases} 1, & \text{falls } x = 0 \\ 0 & \text{sonst} \end{cases}$
2. $\mathrm{leq}\colon \mathbb{N}^2 \to \mathbb{N},\ (x, y) \mapsto \begin{cases} 1, & \text{falls } x \leq y \\ 0 & \text{sonst} \end{cases}$
3. $\mathrm{eq}\colon \mathbb{N}^2 \to \mathbb{N},\ (x, y) \mapsto \begin{cases} 1, & \text{falls } x = y \\ 0 & \text{sonst} \end{cases}$
4. $\mathrm{ifzero}\colon \mathbb{N}^3 \to \mathbb{N},\ (x, y, z) \mapsto \begin{cases} y, & \text{falls } x = 0 \\ z & \text{sonst} \end{cases}$
5. $\max\colon \mathbb{N}^2 \to \mathbb{N},\ (x, y) \mapsto \begin{cases} y, & \text{falls } x \leq y \\ x & \text{sonst} \end{cases}$
6. $\Delta\colon \mathbb{N} \to \mathbb{N},\ x \mapsto \sum_{i=0}^{x} i$

Lösung

(Siehe Ergänzungsblatt 3, Aufgabe 2.)

Aufgabe 21

Sei $P\colon \mathbb{N}^3 \to \mathbb{N}$ eine primitiv rekursive Funktion mit Wertebereich $\{0,1\}$. Zeigen Sie, dass der (dreistellige) *beschränkte Existenzoperator* $\exists^3_P\colon \mathbb{N} \to \mathbb{N}$ mit

$$\exists^3_P(x,y,z) = \begin{cases} 1, & \text{falls } \exists \ell \leq x\colon P(\ell,y,z) = 1 \\ 0 & \text{sonst.} \end{cases}$$

primitiv rekursiv ist, indem Sie einen definierenden, primitiv rekursiven Ausdruck angeben.

Hinweis: $\exists^3_P$ ist eine Verallgemeinerungen des beschränkten Existenzoperators Q aus Vorlesungsfolie 10.6.

Lösung

(Siehe Ergänzungsblatt 3, Aufgabe 3.)

Aufgabe 22

Sei $P\colon \mathbb{N}^2 \to \mathbb{N}$ eine primitiv rekursive Funktion mit Wertebereich $\{0,1\}$. Zeigen Sie, dass der (zweistellige) *beschränkte Maximumoperator* $\max^2_P\colon \mathbb{N}^2 \to \mathbb{N}$ mit

$$\max^2_P(x,y) = \begin{cases} \max\{k \leq x \mid P(k,y) = 1\}, & \text{falls ein } k \leq x \text{ existiert mit } P(k,y) = 1 \\ 0 & \text{sonst.} \end{cases}$$

primitiv rekursiv ist, indem Sie für einen definierenden, primitiv rekursiven Ausdruck angeben.

Hinweis: $\max^2_P$ ist eine Verallgemeinerung des beschränkten Maximumoperators q aus Vorlesungsfolie 10.6.

Lösung

(Siehe Ergänzungsblatt 3, Aufgabe 4.)

Aufgabe 23

Seien $k \geq 1$ eine Konstante und $P\colon \mathbb{N}^k \to \mathbb{N}$ eine primitiv rekursive Funktion mit Wertebereich $\{0,1\}$. Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind.

Hinweis: Sie müssen keinen definierenden, primitiv rekursiven Ausdruck angeben.

1. Der k-stellige *beschränkte Existenzoperator* $\exists^k_P\colon \mathbb{N}^k \to \mathbb{N}$ mit

$$\exists^k_P(n_1,n_2,\ldots,n_k) = \begin{cases} 1, & \text{falls } \exists x \leq n_1\colon P(x,n_2,\ldots,n_k) = 1 \\ 0 & \text{sonst.} \end{cases}$$

2. Der k-stellige *beschränkte Alloperator* $\forall_P^k \colon \mathbb{N}^k \to \mathbb{N}$ mit

$$\forall_P^k(n_1, n_2, \ldots, n_k) = \begin{cases} 1, & \text{falls } \forall x \leq n_1 \colon P(x, n_2, \ldots, n_k) = 1 \\ 0 & \text{sonst.} \end{cases}$$

3. Der k-stellige *beschränkte Minimumoperator* $\min_P^k \colon \mathbb{N}^k \to \mathbb{N}$ mit

$$\min_P^k(n_1, n_2, \ldots, n_k) = \min\{x \leq n_1 \mid P(x, n_2, \ldots, n_k) = 1\},$$

falls ein $x \leq n_1$ existiert mit $P(x, n_2, \ldots, n_k) = 1$ und $\min_P^k(n_1, n_2, \ldots, n_k) = n_1 + 1$ sonst.

4. Der k-stellige *beschränkte Maximumoperator* $\max_P^k \colon \mathbb{N}^k \to \mathbb{N}$ mit

$$\max_P^k(n_1, n_2, \ldots, n_k) = \max\{x \leq n_1 \mid P(x, n_2, \ldots, n_k) = 1\},$$

falls ein $x \leq n_1$ existiert mit $P(x, n_2, \ldots, n_k) = 1$ und $\max_P^k(n_1, n_2, \ldots, n_k) = 0$ sonst.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 4, Aufgabe 4)

Fehlerhinweis: Bei Teilaufgabe 3 sollte $\min_P^k(0, n_1, \ldots, n_k) = 1 - P(0, n_1, \ldots, n_k)$ und <u>nicht</u> $\min_P^k(0, n_1, \ldots, n_k) = 0$ stehen.

Aufgabe 24

Zeigen Sie, dass folgende Funktionen primitiv rekursiv sind.

1. divides: $\mathbb{N}^2 \to \mathbb{N},\ (m, n) \mapsto \begin{cases} 1, & \text{falls } n \text{ von } m \text{ geteilt wird} \\ 0 & \text{sonst} \end{cases}$
2. prime: $\mathbb{N} \to \mathbb{N},\ n \mapsto \begin{cases} 1, & \text{falls } n \text{ prim ist} \\ 0 & \text{sonst} \end{cases}$
3. $d \colon \mathbb{N} \to \mathbb{N},\ n \mapsto |\{m \leq n \mid m \text{ teilt } n\}|$
4. $\pi \colon \mathbb{N} \to \mathbb{N},\ n \mapsto |\{p \leq n \mid p \text{ prim}\}|$

Hinweis: d heißt *Teileranzahlfunktion* und π *Primzahlfunktion.*

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 4, Aufgabe 5)

Aufgabe 25

Welche der folgenden Aussagen sind für beliebige Funktionen $a, b, f, g\colon \mathbb{N} \to \mathbb{N}$ mit

$$f(x) = \sum_{i=0}^{x} a(i) \qquad \text{und} \qquad g(x) = \sum_{i=0}^{b(x)} a(i)$$

richtig? Beweisen Sie Ihre Antworten.

1. Wenn a primitiv rekursiv ist, dann ist auch f primitiv rekursiv.
2. Wenn f primitiv rekursiv ist, dann ist auch a primitiv rekursiv.
3. Wenn a und b primitiv rekursiv sind, dann ist auch g primitiv rekursiv.
4. Wenn g primitiv rekursiv ist, dann sind auch a und b primitiv rekursiv.

Lösung

1. Richtig.

   Wenn a primitiv rekursiv ist, dann ist es auch $f = \mathrm{rec}(c_0^0, \mathrm{add}(\pi_1^2, a(s(\pi_2^2))))$.

2. Richtig.

   Wenn f primitiv rekursiv ist, dann ist es auch $a = \mathrm{rec}(f(c_0^0), \mathrm{sub}(f(s(\pi_2^2)), f(\pi_2^2)))$.

3. Richtig.

   Wenn a und b primitiv rekursiv sind, dann sind es auch f (nach Teil 1) und $g = f(b)$.

4. Falsch.

   Für $a = c_0^1$ und b eine beliebige nicht primitiv rekursive Funktion ist $g = a = c_0^1$ primitiv rekursiv, obwohl b das nicht ist.

   Denkaufgabe: Ein Gegenbeispiel finden bei dem g primitiv rekursiv ist und sowohl a als auch b nicht.

Aufgabe 26

Welche der folgenden Aussagen gelten für beliebige Funktionen $f, g\colon \mathbb{N} \to \mathbb{N}$? Begründen Sie Ihre Antworten.

1. Wenn f und g primitiv rekursiv sind, dann ist auch $f \circ g$ primitiv rekursiv.
2. Wenn f und $f \circ g$ primitiv rekursiv sind, dann ist auch g primitiv rekursiv.
3. Wenn g und $f \circ g$ primitiv rekursiv sind, dann ist auch f primitiv rekursiv.

Hierbei bezeichne $\circ$ die Komposition von Funktionen.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 5, Aufgabe 2)

Aufgabe 27

Sei $c\colon \mathbb{N}^2 \to \mathbb{N}$ die *Cantorsche Paarungsfunktion* aus Vorlesungseinheit 10. Zeigen Sie, dass c bijektiv ist und dass sowohl c als auch die Funktionen $e, f\colon \mathbb{N} \to \mathbb{N}$ mit $e(n) = \pi_1^2(c^{-1}(n))$ und $f(n) = \pi_2^2(c^{-1}(n))$ primitiv rekursiv sind.

Hinweis: Beim Beweis der Injektivität von c ist folgende Monotonieeigenschaft hilfreich:

$$\forall x, y, x', y' \in \mathbb{N}\colon x + y > x' + y' \implies c(x, y) > c(x', y').$$

Lösung

(Siehe Ergänzungsblatt 4, Aufgabe 1.)

Partielle Rekursion

(Vorlesungseinheit 12)

Aufgabe 28

Gegeben sei die Funktion

$$f\colon \mathbb{N}^3 \to \mathbb{N},\ (w, x, y) \mapsto \left\lfloor \frac{x \dot{-} w}{y \dot{-} w} \right\rfloor.$$

Geben Sie eine möglichst einfache Beschreibung von μf an.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 3, Aufgabe 3)

Aufgabe 29

Wir betrachten die Funktion

$$f\colon \mathbb{N}^3 \to \mathbb{N},\ (x, y, z) \mapsto \lceil \log_2(y \dot{-} xz) \rceil,$$

die genau dann für $(x, y, z) \in \mathbb{N}^3$ definiert ist, wenn $xz < y$ gilt. Geben Sie eine möglichst einfache Beschreibung von μf an.

Lösung

(Siehe Ergänzungsblatt 4, Aufgabe 2.)

Aufgabe 30

Geben Sie für jede der folgenden totalen Funktionen f eine möglichst einfache Beschreibung von μf an.

1. $f\colon \mathbb{N}^2 \to \mathbb{N}, (w,x) \mapsto x \dot{-} 2w$
2. $f\colon \mathbb{N}^3 \to \mathbb{N}, (w,x,y) \mapsto (x \dot{-} w) + (y \dot{-} w)$
3. $f\colon \mathbb{N} \to \mathbb{N}, w \mapsto w^2 - 6w + 9$
4. $f\colon \mathbb{N}^4 \to \mathbb{N}, (w,x,y,z) \mapsto (x^2 \dot{-} w) + (z \dot{-} y) + (y \dot{-} 2z)$
5. $f\colon \mathbb{N}^2 \to \mathbb{N}, (w,x) \mapsto 5x \dot{-} w$
6. $f\colon \mathbb{N}^3 \to \mathbb{N}, (w,x,y) \mapsto x + (y \dot{-} w)$
7. $f\colon \mathbb{N} \to \mathbb{N}, w \mapsto \sum_{k=0}^{w} k(k+1)$
8. $f\colon \mathbb{N}^4 \to \mathbb{N}, (w,x,y,z) \mapsto (y+z) \dot{-} (w+x)$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 3, Aufgabe 4)

Aufgabe 31

Zeigen Sie, dass jede der folgenden Funktionen μ-rekursiv ist, indem Sie eine μ-rekursive Funktion f angeben, sodass μf die gegebene Funktion ist.

Welche der Funktionen sind zudem primitiv rekursiv?

1. $\mathrm{div}\colon \mathbb{N}^2 \to \mathbb{N}, (x,y) \mapsto \lfloor \frac{x}{y} \rfloor$
2. $\mathrm{mod}\colon \mathbb{N}^2 \to \mathbb{N}, (x,y) \mapsto x \bmod y$
3. $\mathrm{sqrt}\colon \mathbb{N} \to \mathbb{N}, x \mapsto \lfloor \sqrt{x} \rfloor$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 4, Aufgabe 6)

Die Ackermannfunktion

(Vorlesungseinheit 13)

Aufgabe 32

Beantworten Sie folgende Fragen und begründen Sie Ihre Antworten kurz.

1. Gibt es eine primitiv rekursive Funktion, die nicht Turing-berechenbar ist?
2. Ist die Funktion $f\colon \mathbb{N}^2 \to \mathbb{N}, (m, n) \mapsto \log_n(m)$ primitiv rekursiv?
3. Ist jede totale Turing-berechenbare Funktion primitiv rekursiv?
4. Sei a die Ackermann-Funktion. Ist $f\colon \mathbb{N} \to \mathbb{N}, n \mapsto 1 \dot{-} a(n, n)$ primitiv rekursiv?
5. Ist die nirgends definierte Funktion $\Omega\colon \mathbb{N} \to \mathbb{N}$ μ-rekursiv?

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 5, Aufgabe 1)

Entscheidbarkeit

(Vorlesungseinheit 14)

Aufgabe 33

Zeigen Sie, dass die Klasse der entscheidbaren Sprachen unter folgenden Operationen abgeschlossen ist:

1. Komplement
2. Schnitt
3. Vereinigung
4. Differenz
5. Konkatenation
6. Kleene-Stern

Lösung

1. Sei A entscheidbar, also χ_A berechenbar. Dann ist auch $\chi_{\overline{A}}(w) = 1 - \chi_A(w)$ berechenbar und somit $\overline{A}$ entscheidbar.
2. Seien A und B entscheidbar, also χ_A und χ_B berechenbar. Dann ist auch $\chi_{A \cap B}(w) = \chi_A(w) \cdot \chi_B(w)$ berechenbar und somit $A \cap B$ entscheidbar.
3. Seien A und B entscheidbar, also χ_A und χ_B berechenbar. Dann ist auch $\chi_{A \cup B}(w) = \max\{\chi_A(w), \chi_B(w)\}$ berechenbar und somit $A \cup B$ entscheidbar.
4. Seien A und B entscheidbar, also χ_A und χ_B berechenbar. Dann ist auch $\chi_{A \setminus B}(w) = \chi_A(w) \cdot (1 - \chi_B(w))$ berechenbar und somit $A \setminus B$ entscheidbar.
5. Seien A und B entscheidbar, also χ_A und χ_B berechenbar. Dann kann χ_{AB} mit folgendem Algorithmus berechnet werden:

Eingabe:	w

```
for all u,v satisfying w = uv
    if χ_A(u) = 1 ∧ χ_B(v) = 1
        return 1
return 0
```

6. Sei A entscheidbar, also χ_A berechenbar. Dann kann χ_{A^*} mit folgendem Algorithmus berechnet werden:

Eingabe:	w

```
if w = ε
    return 1
else
    for all u_1,...,u_n satisfying w = u_1...u_n
        if ∀i ∈ [n]: χ_A(u_i) = 1
            return 1
    return 0
```

Aufgabe 34

Welche der folgenden Aussagen sind für eine beliebige Sprache A richtig? Beweisen Sie Ihre Antworten.

1. Wenn A entscheidbar ist, dann ist auch $\overline{A}$ entscheidbar.
2. Wenn A entscheidbar ist, dann ist auch A^* entscheidbar.
3. Wenn A^* entscheidbar ist, dann ist auch A entscheidbar.
4. Wenn A kontextsensitiv (d. h. vom Typ 1) ist, dann ist A entscheidbar.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 5, Aufgabe 3)

Aufgabe 35

Welche der folgenden Aussagen sind für beliebige Sprachen A und B richtig? Beweisen Sie Ihre Antworten.

1. Wenn A und B entscheidbar sind, ist auch AB entscheidbar.
2. Wenn A und AB entscheidbar sind, ist auch B entscheidbar.
3. Wenn A und B entscheidbar sind, ist auch $A \cap B$ entscheidbar.
4. Wenn A und B unentscheidbar sind, ist auch $A \cap B$ unentscheidbar.
5. Wenn A und $A \cap B$ entscheidbar sind, ist auch B entscheidbar.
6. Wenn A und B entscheidbar sind, ist auch $A \cup B$ entscheidbar.

7. Wenn A und B unentscheidbar sind, ist auch $A \cup B$ unentscheidbar.
8. Wenn A und $A \cup B$ entscheidbar sind, ist auch B entscheidbar.
9. Wenn $A \cap B$ und $A \cup B$ entscheidbar sind, sind A und B entscheidbar.
10. Wenn $A \subseteq B$ und B entscheidbar ist, ist auch A entscheidbar.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 5, Aufgabe 4)

Fehlerhinweis: Die Lösung zu Teilaufgabe 2 ist falsch. Ein richtiges Gegenbeispiel ist das folgende: $\{0,1\}^*$ und $\{0,1\}^*(K \cup \{\varepsilon\}) = \{0,1\}^*$ sind entscheidbar, aber $K \cup \{\varepsilon\}$ nicht, da die Vereinigung einer unentscheidbaren und einer endlichen Sprache immer unentscheidbar ist. (Verständnisfrage: Warum ist das so?)

Aufgabe 36

Sei Σ ein Alphabet. Zeigen oder widerlegen Sie für jede der folgenden Sprachen L, dass diese für beliebige rekursiv aufzählbaren Sprachen $A, B, A_0, A_1, A_2, \ldots$ über Σ selbst rekursiv aufzählbar sind.

1. $L = A \cup B$
2. $L = AB$
3. $L = A \cap B$
4. $L = \bigcup_{i \in \mathbb{N}} A_i$
5. $L = \bigcap_{i \in \mathbb{N}} A_i$
6. $L = A^*$

Lösung

(Siehe Ergänzungsblatt 5, Aufgabe 1.)

Reduktionen und der Satz von Rice

(Vorlesungseinheiten 15-16)

Aufgabe 37

Sei L die Menge aller Gödelisierungen von Turingmaschinen, die eine Funktion mit unendlicher Bildmenge berechnen, d. h. :

$$L = \{w \,|\, \text{die Bildmenge der von } M_w \text{ berechneten Funktion ist unendlich}\}.$$

Zeigen Sie, dass L unentscheidbar ist, indem Sie

1. den Satz von Rice anwenden.
2. $H_0 \leq L$ zeigen.

Lösung

(Siehe Ergänzungsblatt 5, Aufgabe 2.)

Aufgabe 38

Sei L die Menge aller Gödelisierungen von Turingmaschinen, die eine totale und streng monoton steigende Funktion berechnen, d. h. :

$$L = \{w \mid \text{die von } M_w \text{ berechnete Funktion ist total und streng monoton steigend}\}.$$

Zeigen Sie, dass L unentscheidbar ist, indem Sie

1. $H_0 \leq L$ zeigen.
2. den Satz von Rice anwenden.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 5, Aufgabe 5)

Aufgabe 39

Gegeben seien die Sprachen

$$A = \{u\#v \mid T(M_u) \subseteq T(M_v)\} \quad \text{und} \quad B = \{u\#v \mid T(M_u) = T(M_v)\}.$$

1. Zeigen Sie: $H_0 \leq A$.
2. Zeigen Sie: $A \leq B$.
3. Zeigen Sie: $\overline{H_0} \leq B$.
4. Zeigen Sie: $B \leq A$.
5. Was folgt daraus für die semi- und die co-semi-Entscheidbarkeit von A bzw. B?

Hinweis: Wegen

$$A \leq B \iff \overline{A} \leq \overline{B}$$

gilt für beliebige Sprachen A und B: Falls $A \leq B$ und B co-semi-entscheidbar ist, dann ist auch A co-semi-entscheidbar.

Lösung

1. Wähle $f(x) = u\#v$ mit u ein Gödelindex einer Turingmaschine mit $T(M_u) = \{\varepsilon\}$ und v ein Gödelindex einer Turingmaschine, die den Bandinhalt löscht und sich dann wie M_x verhält.

   Dann ist f total und berechenbar und für alle $x \in \{0,1\}^*$ gilt

$$\begin{aligned} x \in H_0 &\implies M_x \text{ hält auf } \varepsilon \\ &\implies M_v \text{ hält auf jede Eingabe} \\ &\implies T(M_v) = \Sigma_v^* \\ &\implies T(M_u) \subseteq T(M_v) \\ &\implies f(x) \in A \end{aligned}$$

und

$$\begin{aligned} x \notin H_0 &\implies M_x \text{ hält nicht auf } \varepsilon \\ &\implies M_v \text{ hält auf keine Eingabe} \\ &\implies T(M_v) = \emptyset \\ &\implies T(M_u) \nsubseteq T(M_v) \\ &\implies f(x) \notin A, \end{aligned}$$

also $x \in H_0 \iff f(x) \in A$.

2. Falls $x = u\#v$ für $u, v \in \{0,1\}^*$, wähle $f(x) = u'\#v'$ mit $T(M_{u'}) = T(M_u)$ und $T(M_{v'}) = T(M_u) \cap T(M_v)$.

   Dann gilt für alle $x \in \{0,1\}^*\{\#\}\{0,1\}^*$:

$$\begin{aligned} x \in A &\implies T(M_u) \subseteq T(M_v) \\ &\implies T(M_u) = T(M_u) \cap T(M_v) \\ &\implies T(M_{u'}) = T(M_{v'}) \\ &\implies f(x) \in B \end{aligned}$$

   und

$$\begin{aligned} x \notin A &\implies T(M_u) \nsubseteq T(M_v) \\ &\implies T(M_u) \neq T(M_u) \cap T(M_v) \\ &\implies T(M_{u'}) \neq T(M_{v'}) \\ &\implies f(x) \notin B. \end{aligned}$$

   Falls $x \notin \{0,1\}^*\{\#\}\{0,1\}^*$, wähle $f(x) = x$. Somit ist f total und berechenbar und für alle $x \in \{0,1,\#\}^*$ gilt: $x \in A \iff f(x) \in B$.

   Bemerkung: Eine weitere Möglichkeit wäre die Wahl $T(M_{u'}) = T(M_v)$ und $T(M_{v'}) = T(M_u) \cup T(M_v)$ für $f(u\#v) = u'\#v'$ gewesen. Die Wahl $T(M_{u'}) = \emptyset$ und $T(M_{v'}) = T(M_u) \setminus T(M_v)$ erfüllt zwar die Äquivalenz, liefert aber eine im Allgemeinen nicht berechenbare Funktion, weil es kein algorithmisches Verfahren existiert, das für zwei gegebene Turingmaschinen M_1 und M_2 eine Turingmaschine M mit $T(M) = T(M_1) \setminus T(M_2)$ konstruiert. Würde ein solches Verfahren existieren, dann wären Typ-0-Sprachen unter Komplement abgeschlossen, was nicht stimmt.

3. Wähle $f(x) = u\#v$ mit u ein Gödelindex einer Turingmaschine mit $T(M_u) = \emptyset$ und v ein Gödelindex einer Turingmaschine, die den Bandinhalt löscht und sich dann wie M_x verhält.

   Dann ist f total und berechenbar und für alle $x \in \{0,1\}^*$ gilt

$$\begin{aligned} x \in \overline{H_0} &\implies M_x \text{ hält nicht auf } \varepsilon \\ &\implies M_v \text{ hält auf keine Eingabe} \\ &\implies T(M_v) = \emptyset \\ &\implies T(M_u) = T(M_v) \\ &\implies f(x) \in B \end{aligned}$$

und

$$\begin{aligned} x \notin \overline{H_0} &\implies M_x \text{ hält auf } \varepsilon \\ &\implies M_v \text{ hält auf jede Eingabe} \\ &\implies T(M_v) = \Sigma_v^* \\ &\implies T(M_u) \neq T(M_v) \\ &\implies f(x) \notin B, \end{aligned}$$

also $x \in \overline{H_0} \iff f(x) \in B$.

4. Falls $x = u\#v$ für $u, v \in \{0,1\}^*$, wähle $f(x) = u'\#v'$ mit $T(M_{u'}) = T(M_u) \cup T(M_v)$ und $T(M_{v'}) = T(M_u) \cap T(M_v)$.

   Dann gilt für alle $x \in \{0,1\}^*\{\#\}\{0,1\}^*$:

$$\begin{aligned} x \in B &\implies T(M_u) = T(M_v) \\ &\implies T(M_u) \cup T(M_v) = T(M_u) \cap T(M_v) \\ &\implies T(M_{u'}) = T(M_{v'}) \\ &\implies T(M_{u'}) \subseteq T(M_{v'}) \\ &\implies f(x) \in A \end{aligned}$$

   und

$$\begin{aligned} x \notin B &\implies T(M_u) \neq T(M_v) \\ &\implies T(M_u) \cup T(M_v) \neq T(M_u) \cap T(M_v) \\ &\implies T(M_u) \cup T(M_v) \nsubseteq T(M_u) \cap T(M_v) \\ &\implies T(M_{u'}) \nsubseteq T(M_{v'}) \\ &\implies f(x) \notin A. \end{aligned}$$

   Falls $x \notin \{0,1\}^*\{\#\}\{0,1\}^*$, wähle $f(x) = x$. Somit ist f total und berechenbar und für alle $x \in \{0,1,\#\}^*$ gilt: $x \in A \iff f(x) \in B$.

5. Aus $H_0 \leq A$ folgt, dass A nicht co-semi-entscheidbar ist. Aus $A \leq B$ folgt, dass B auch nicht co-semi-entscheidbar ist. Aus $\overline{H_0} \leq B$ folgt, dass B auch nicht semi-entscheidbar ist. Schließlich folgt aus $B \leq A$, dass A auch nicht semi-entscheidbar ist. Also sind sowohl A als auch B weder semi- noch co-semi-entscheidbar.

Aufgabe 40

Wir betrachten folgende Entscheidungsprobleme für Turingmaschinen:

(a) Das Leerheitsproblem EMP

Eingabe: Eine Turingmaschine M
Frage: Ist die von M akzeptierte Sprache leer?

(b) Das Endlichkeitsproblem FIN

Eingabe: Eine Turingmaschine M
Frage: Ist die von M akzeptierte Sprache endlich?

(c) Das Regularitätsproblem REG
Eingabe: Eine Turingmaschine M
Frage: Ist die von M akzeptierte Sprache regulär?

Zeigen Sie:

1. EMP ist unentscheidbar.
2. EMP $\leq$ FIN.
3. FIN $\leq$ REG.

Lösung

(Siehe Ergänzungsblatt 5, Aufgabe 4.)

Aufgabe 41

Seien Σ ein Alphabet und $u, v \in \Sigma^*$ Wörter über Σ. Wir betrachten die Menge $L_{u,v}$ aller Gödelindizes von Turingmaschinen, die u akzeptieren, aber v nicht, d. h. :

$$L_{u,v} = \{w \mid u \in T(M_w) \wedge v \notin T(M_w)\}\,.$$

1. Zeigen Sie: Wenn $u \neq v$, dann $H_0 \leq L_{u,v}$.
2. Zeigen Sie: Wenn $u \neq v$, dann $\overline{H_0} \leq L_{u,v}$.
3. Für welche u, v ist $L_{u,v}$ semi- bzw. co-semi-entscheidbar?

Lösung

1. Angenommen, $u \neq v$. Wähle $f(x)$ als Gödelindex einer Turingmaschine mit Eingabealphabet Σ, die auf Eingabe w wie folgt arbeitet:

   - Simuliere M_x auf ε.
   - Falls $w = u$, akzeptiere,
   - Falls $w \neq u$, gehe in eine Endlosschleife.

   Dann ist f total und berechenbar und für alle $x \in \{0,1\}^*$ gilt

$$\begin{aligned} x \in H_0 &\implies M_x \text{ hält auf } \varepsilon \\ &\implies T(M_{f(x)}) = \{u\} \\ &\implies u \in T(M_{f(x)}) \wedge v \notin T(M_{f(x)}) \\ &\implies f(x) \in L_{u,v} \end{aligned}$$

   und

$$\begin{aligned} x \notin H_0 &\implies M_x \text{ hält nicht auf } \varepsilon \\ &\implies T(M_{f(x)}) = \emptyset \\ &\implies u \notin T(M_{f(x)}) \\ &\implies f(x) \notin L_{u,v}, \end{aligned}$$

   also $x \in H_0 \iff f(x) \in L_{u,v}$.

2. Angenommen, $u \neq v$. Wähle $f(x)$ als Gödelindex einer Turingmaschine mit Eingabealphabet Σ, die auf Eingabe w wie folgt arbeitet:
   - Falls $w = u$, akzeptiere.
   - Simuliere M_x auf ε.
   - Akzeptiere.

   Dann ist f total und berechenbar und für alle $x \in \{0,1\}^*$ gilt

$$\begin{aligned} x \in \overline{H_0} &\implies M_x \text{ hält nicht auf } \varepsilon \\ &\implies T(M_{f(x)}) = \{u\} \\ &\implies u \in T(M_{f(x)}) \wedge v \notin T(M_{f(x)}) \\ &\implies f(x) \in L_{u,v} \end{aligned}$$

   und

$$\begin{aligned} x \notin \overline{H_0} &\implies M_x \text{ hält auf } \varepsilon \\ &\implies T(M_{f(x)}) = \Sigma^* \\ &\implies v \in T(M_{f(x)}) \\ &\implies f(x) \notin L_{u,v}, \end{aligned}$$

   also $x \in \overline{H_0} \iff f(x) \in L_{u,v}$.

3. Aus $H_0 \leq L_{u,v}$ folgt für alle u, v mit $u \neq v$, dass $L_{u,v}$ nicht co-semi-entscheidbar ist. Aus $\overline{H_0} \leq L_{u,v}$ folgt für alle u, v mit $u \neq v$, dass $L_{u,v}$ nicht semi-entscheidbar ist. Für alle u, v mit $u = v$ ist $L_{u,v}$ leer und somit entscheidbar.

Aufgabe 42

Welche der folgenden Aussagen sind für beliebige Alphabete Σ, Γ und Δ und beliebige Sprachen A über Σ, B über Γ und C über Δ richtig? Beweisen Sie Ihre Antworten.

1. Wenn $A \leq B$, dann $\overline{B} \leq \overline{A}$ gilt.
2. Wenn $A \leq B$, dann $\overline{A} \leq \overline{B}$ gilt.
3. Wenn B regulär und $A \leq B$, dann ist A regulär.
4. Wenn B co-semi-entscheidbar und $A \leq B$, dann ist A co-semi-entscheidbar.
5. Wenn B unentscheidbar, dann gilt $A \leq B$ für alle Sprachen A.
6. $A \leq A$
7. $A \leq \overline{A}$
8. Wenn $A \leq B$, dann $A \subseteq B$.
9. Wenn $A \subseteq B$, dann $A \leq B$.
10. Wenn $A \leq B$ und $B \leq C$, dann $A \leq C$.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 6, Aufgabe 4)

Aufgabe 43

Welche der folgenden Sprachen sind semi-entscheidbar und welche co-semi-entscheidbar? Beweisen Sie Ihre Antworten.

Hinweis: Für $w \in \{0,1\}^*$ sei φ_w die von M_w berechnete Funktion.

1. $L = \{w \mid M_w \text{ akzeptiert mindestens ein Wort } x \text{ mit } |x| \leq 100\}$
2. $L = \{w \mid M_w \text{ berechnet } \chi_K\}$
3. $L = \{w \mid M_w \text{ hält auf keine Eingabe in höchstens 100 Schritten}\}$
4. $L = \{w \mid \varphi_w \text{ ist an der Stelle 0 nicht definiert}\}$
5. $L = \left\{w \,\middle|\, \varphi_w \text{ reduziert } \overline{H_0} \text{ auf } K\right\}$
6. $L = \{w \mid \varphi_w(0) = 0\}$
7. $L = \{w \mid \varphi_w(w) = 0\}$
8. $L = \{w \mid \varphi_0(0) = w\}$
9. $L = \{w \mid T(M_w) \text{ ist semi-entscheidbar}\}$
10. $L = \{w \mid |T(M_w)| \leq 100\}$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 6, Aufgabe 5)

Aufgabe 44

Welche der folgenden Probleme sind semi- und welche co-semi-entscheidbar?

1. Problem A
   Eingabe: Eine DTM M
   Frage: Berechnet M die Ackermannfunktion a?
2. Problem B
   Eingabe: Eine DTM M
   Frage: Besitzt M mindestens 5 Zustände?
3. Problem C
   Eingabe: Eine DTM M.
   Frage: Besucht M, wenn auf ε gestartet, alle Zustände?

Lösung

(Siehe Ergänzungsblatt 6, Aufgabe 2.)

Aufgabe 45

Sei M eine DTM mit

$$\forall n \in \mathbb{N} \colon |\, \{w \in T(M) \,|\, |w| = n\} \,| = 1.$$

Zeigen Sie, dass $T(M)$ entscheidbar ist.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 7, Aufgabe 2)

Aufgabe 46

Seien Σ ein Alphabet und L eine Sprache über Σ. Zeigen Sie, dass L genau dann semi-entscheidbar ist, wenn sich L auf das allgemeine Halteproblem

$$H = \{w\#x \,|\, M_w \text{ hält auf } x\}$$

reduzieren lässt.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 7, Aufgabe 3)

Aufgabe 47

Geben Sie jeweils eine Funktion mit den geforderten Eigenschaften an oder begründen Sie kurz warum keine solche Funktion existieren kann.

1. Eine berechenbare Funktion, die nicht total ist.
2. Eine totale Funktion, die nicht berechenbar ist.
3. Eine Funktion, die weder berechenbar noch total ist.
4. Eine primitiv rekursive Funktion mit endlichem Definitionsbereich.
5. Eine primitiv rekursive Funktion mit endlichem Wertebereich.
6. Eine primitiv rekursive Funktion mit unentscheidbarem Definitionsbereich.
7. Eine totale und berechenbare Funktion, die nicht primitiv rekursiv ist.
8. Eine Funktion f, sodass $L = \{w \,|\, M_w \text{ berechnet } f\}$ entscheidbar ist.
9. Eine berechenbare Funktion f, sodass $L = \{w \,|\, M_w \text{ berechnet } f\}$ entscheidbar ist.
10. Eine nicht berechenbare Funktion mit endlichem Wertebereich.
11. Eine berechenbare Funktion mit einem nicht rekursiv aufzählbarem Wertebereich.
12. Eine nicht berechenbare Funktion mit endlichem Definitionsbereich.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 6, Aufgabe 3)

Das Postsche Korrespondenzproblem

(Vorlesungseinheiten 17-19)

Aufgabe 48

Geben Sie für jede der folgenden Instanzen P des Postschen Korrespondenzproblems über $\Sigma = \{0, 1\}$ eine Lösung kürzester Länge an oder begründen Sie, warum diese keine Lösung besitzen.

1. $P = ((100, 1), (1, 01), (1, 010), (10, 001))$
2. $P = ((100, 10), (01, 1), (1010, 101))$
3. $P = ((001, 00), (11, 110), (10, 101), (101, 1010))$
4. $P = ((001, 10), (01, 0100), (011, 11))$
5. $P = ((100, 11), (010, 101), (0, 01))$
6. $P = ((00, 100), (10, 101), (11, 01), (010, 0))$
7. $P = ((10, 110), (001, 101), (1, 010), (101, 01))$
8. $P = ((01, 1001), (10, 01), (100, 10))$

Lösung

1. (Siehe Ergänzungsblatt 6, Aufgabe 3.1.)
2. Diese Instanz besitzt keine Lösung, weil in jedem Paar die erste Komponente länger ist als die zweite.
3. (Siehe Ergänzungsblatt 6, Aufgabe 3.2.)
4. (Siehe Ergänzungsblatt 6, Aufgabe 3.3.)
5. (Siehe Ergänzungsblatt 6, Aufgabe 3.4.)
6. Lösungen kürzester Länge: $(4, 1, 3, 1)$ und $(4, 2, 3, 1)$.
7. Diese Instanz besitzt keine Lösung, weil in keinem Paar eine Komponente ein Präfix der anderen ist, d. h. eine Lösung kann mit keinem der Indizes beginnen.
8. Jede Lösung dieser Instanz muss mit Index 3 beginnen. Für die restlichen Stellen der Lösung kommt jeweils nur Index 2 infrage. Somit besitzt diese Instanz keine Lösung.

Aufgabe 49

In der Vorlesung wurde das *modifizierte Postsche Korrespondenzproblem* MPCP auf das *Postsche Korrespondenzproblem* PCP reduziert.

Zeigen Sie, dass auch PCP $\leq$ MPCP gilt.

Erinnerung: PCP und MPCP sind wie folgt definiert:

<u>PCP</u>
Eingabe: Ein Alphabet Σ und Wortpaare $(x_1, y_1), \ldots, (x_k, y_k) \in \Sigma^+ \times \Sigma^+$.
Frage: Gibt es eine Folge $(i_1, \ldots, i_n) \in [k]^n$ mit $n \in \mathbb{N} \backslash \{0\}$ und $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

<u>MPCP</u>
Eingabe: Ein Alphabet Σ und Wortpaare $(x_1, y_1), \ldots, (x_k, y_k) \in \Sigma^+ \times \Sigma^+$.
Frage: Besitzt diese PCP-Instanz eine Lösung $(i_1, \ldots, i_n)$ mit $i_1 = 1$?

Lösung

(Siehe Ergänzungsblatt 6, Aufgabe 4.)

Aufgabe 50

Zeigen Sie: PCP_Σ ist genau dann entscheidbar, wenn Σ ein unäres Alphabet ist.

Lösung

(Siehe Ergänzungsblatt 7, Aufgabe 1.)

Aufgabe 51

Gegeben sei die folgende Instanz K des Postschen Korrespondenzproblems:

$$K = ((0, 10), (01, 100), (11, 1)).$$

Ziel dieser Aufgabe ist es, die Idee des Beweis auf Vorlesungsfolie 18.5 anhand eines konkreten Beispiels zu illustrieren.

1. Geben Sie eine möglichs kurze Lösung von K an.
2. Geben Sie die Grammatiken G_1 und G_2 aus dem Beweis an.
3. Geben Sie ein möglichst kurzes Wort $w \in L(G_1) \cap L(G_2)$ an.
4. Geben Sie die Syntaxbäume von w bezüglich G_1 und G_2 an.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 4)

Der Gödelsche Unvollständigkeitssatz

(Vorlesungseinheiten 20-22)

Aufgabe 52

Geben Sie zu jeder der folgenden arithmetischen Formeln F an, welche Variablen frei und welche gebunden in F vorkommen, und finden Sie passende Belegungen Φ („Phi“) und Ψ („Psi“), sodass $\Phi(F)$ wahr und $\Psi(F)$ falsch ist.

1. $F = (\exists x((5+x) = y) \wedge (\exists y((3+y) = z))$
2. $F = (\forall z((z * x) = (z + y)))$
3. $F = (\forall y \exists z(y = (x + z)))$
4. $F = (((x + y) = (x * y)) \wedge \neg((x + y) = 0))$
5. $F = (\exists y(x = (2 + (6 * y))) \wedge \exists y(x = (5 + (9 * y))))$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 5)

Aufgabe 53

Geben Sie eine arithmetische Formel F mit der jeweiligen Eigenschaft an.

1. $F(x, y, z)$ ist wahr $\iff$ $x \leq y < z$
2. $F(x)$ ist wahr $\iff$ x ist die Summe von zwei Quadratzahlen
3. $F(x, y)$ ist wahr $\iff$ x teilt y
4. $F(x)$ ist wahr $\iff$ x ist prim

Hinweise:

- Sie dürfen Klammern weglassen, wenn klar ist, welche Formel gemeint ist.
- Man schriebt oft $F(x_{i_1}, \ldots, x_{i_n})$ statt F, um hervorzuheben, dass $x_{i_1}, \ldots, x_{i_n}$ genau die freien Variablen in F sind.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 6)

Aufgabe 54

Zeigen Sie mithilfe der Definition der arithmetischen Repräsentierbarkeit, dass folgende Funktionen f arithmetisch repräsentierbar sind.

1. $f\colon \mathbb{N} \to \mathbb{N}, x \mapsto \sqrt{x}$, falls x eine Quadratzahl ist (sonst undefiniert)
2. $f\colon \mathbb{N}^2 \to \mathbb{N}, (x, y) \mapsto \min(x, y)$
3. $f\colon \mathbb{N} \to \mathbb{N}, x \mapsto 3x^2 \dot{-} 2x$
4. $f\colon \mathbb{N}^2 \to \mathbb{N}, (x, y) \mapsto \mathrm{ggT}(x, y)$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 7)

Aufgabe 55

Sei $(n_0, n_1, n_2) = (1, 2, 0)$ eine endliche Zahlenfolge. Bestimmen Sie zwei natürliche Zahlen $a, b \in \mathbb{N}$ für die gilt:

$$\forall i \in \{0, 1, 2\} \colon G(a, b, i, n_i).$$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 8)

Aufgabe 56

Sei $(n_0, n_1, n_2, n_3) = (1, 0, 2, 3)$ eine endliche Zahlenfolge. Bestimmen Sie zwei natürliche Zahlen $a, b \in \mathbb{N}$ für die gilt:

$$\forall i \in \{0, 1, 2, 3\} \colon G(a, b, i, n_i).$$

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 9)

Aufgabe 57

Zeigen Sie mithilfe des $G(a, b, i, \cdot)$-Prädikats, dass die Fakultätsfunktion

$$f \colon \mathbb{N} \to \mathbb{N}, n \mapsto n!$$

arithmetisch repräsentierbar ist.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 8, Aufgabe 10)

Die Klassen P und NP

(23-28)

Aufgabe 58

Seien $\Sigma = \{a, b\}$ ein Alphabet, M_1 die 1-Band DTM

$$M_1 = (\{0, 1, 2, 3, 4\}, \Sigma, \Sigma \cup \{m, \Box\}, \delta, 0, \Box, \{4\})$$

mit

q	$\delta(q, a)$	$\delta(q, b)$	$\delta(q, m)$	$\delta(q, \Box)$
0	$(0, a, R)$	$(1, m, L)$		
1	$(2, m, R)$		$(1, m, L)$	
2		$(1, m, L)$	$(2, m, R)$	$(3, \Box, L)$
3			$(3, m, L)$	$(4, \Box, N)$

und L die von M_1 akzeptierte Sprache.

1. Geben Sie L in Mengenschreibweise an.
2. Geben Sie $\mathrm{time}_{M_1}(x)$ für alle $x \in L$ an.
3. Geben Sie eine 2-Band-DTM M_2 für L an, sodass für alle $x \in \Sigma^*$ gilt:
$$\mathrm{time}_{M_2}(x) \leq |x| + 1.$$
4. Gibt es eine Mehrband-DTM M für L, sodass die Ungleichung
$$\mathrm{time}_M(x) \leq |x|$$
für mindestens ein $x \in L$ gilt? Begründen Sie Ihre Antwort.

Lösung

(Siehe Ergänzungsblatt 7, Aufgabe 2.)

Aufgabe 59

Sie sind Doktorand und arbeiten gerade an Ihrer Dissertation über Komplexitätstheorie. Plötzlich reißt ein Kollege die Bürotür auf und platzt voller Entsetzen heraus:

> *„Eine Turingmaschine, die nur $n + 1$ Schritte machen darf, kann das Eingabewort nur einmal von links nach rechts durchlaufen und danach, wenn sie das erste Leersymbol nach dem Eingabewort sieht, in einem Schritt entscheiden, ob sie das Wort akzeptiert oder nicht. So ähnlich arbeiten doch endliche Automaten. Enthält dann $\mathsf{TIME}(n + 1)$ genau die regulären Sprachen?“*

Wie reagieren Sie darauf?

Lösung

(Siehe Ergänzungsblatt 7, Aufgabe 3.)

Aufgabe 60

Sie schreiben wieder an Ihrer Dissertation, bis Ihr Kollege wieder in Ihr Büro stürmt und stammelt:

> *„Enthält vielleicht $\mathsf{TIME}(n)$ genau die regulären Sprachen?“*

Wie reagieren Sie diesmal darauf?

Lösung

(Siehe Ergänzungsblatt 7, Aufgabe 4.)

Aufgabe 61

Seien Σ, Γ und Δ drei Alphabete, A eine Sprache über Σ, B eine Sprache über Γ und C eine Sprache über Δ.

Zeigen Sie:

1. $A \leq_p A$. (Reflexivität von $\leq_p$)
2. Wenn $A \leq_p B$ und $B \leq_p C$, dann $A \leq_p C$. (Transitivität von $\leq_p$)
3. Für $A = \emptyset$ und $B \neq \Gamma^*$ gilt $A \leq_p B$.
4. Für $A = \Sigma^*$ und $B \neq \emptyset$ gilt $A \leq_p B$.
5. Für $A \in \mathsf{P}$ und $\emptyset \neq B \neq \Gamma^*$ gilt $A \leq_p B$.
6. Wenn $A \leq_p B$ und A NP-hart, dann ist auch B NP-hart.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 9, Aufgabe 4)

Aufgabe 62

Seien true und false nullstellige logische Junktoren (auch: *Konstanten*) mit $\mathcal{A}(\text{true}) = 1$ und $\mathcal{A}(\text{false}) = 0$ für jede Belegung $\mathcal{A}$ Formeln, die nur Variablen, Konstanten und einen weiteren Junktor $\odot$ enthalten, nennen wir in dieser Aufgabe $\odot$-*Formeln*.

1. Passt die Belegung $\mathcal{A}$ zu Formeln F und G, dann gilt für die *Kontravalenz* $\oplus$:

   $$\mathcal{A}(F \oplus G) = \begin{cases} 0, & \text{falls } \mathcal{A}(F) = \mathcal{A}(G) \\ 1, & \text{falls } \mathcal{A}(F) \neq \mathcal{A}(G). \end{cases}$$

   Zeigen Sie, dass folgendes Entscheidungsproblem in P liegt:

   <u>XOR-SAT</u>
   Eingabe: Eine $\oplus$-Formel.
   Frage: Ist F erfüllbar?

2. Passt die Belegung $\mathcal{A}$ zu Formeln F und G, dann gilt für die *Exklusion* $\bar{\wedge}$:

   $$\mathcal{A}(F \,\bar{\wedge}\, G) = \begin{cases} 0, & \text{falls } \mathcal{A}(F) = 1 \text{ und } \mathcal{A}(G) = 1 \\ 1, & \text{falls } \mathcal{A}(F) = 0 \text{ oder } \mathcal{A}(G) = 0. \end{cases}$$

   Zeigen Sie, dass folgendes Entscheidungsproblem NP-vollständig ist:

   <u>NAND-SAT</u>
   Eingabe: Eine $\bar{\wedge}$-Formel.
   Frage: Ist F erfüllbar?

3. Passt die Belegung $\mathcal{A}$ zu Formeln F und G, dann gilt für die *Nihilition* $\bar{\vee}$:

$$\mathcal{A}(F \mathbin{\bar{\vee}} G) = \begin{cases} 0, & \text{falls } \mathcal{A}(F) = 1 \text{ oder } \mathcal{A}(G) = 1 \\ 1, & \text{falls } \mathcal{A}(F) = 0 \text{ und } \mathcal{A}(G) = 0. \end{cases}$$

   Zeigen Sie, dass folgendes Entscheidungsproblem NP-vollständig ist:

   <u>NOR-SAT</u>
   Eingabe: Eine $\bar{\vee}$-Formel.
   Frage: Ist F erfüllbar?

4. Passt die Belegung $\mathcal{A}$ zu Formeln F und G, dann gilt für die *Implikation* $\rightarrow$:

$$\mathcal{A}(F \rightarrow G) = \begin{cases} 0, & \text{falls } \mathcal{A}(F) = 1 \text{ und } \mathcal{A}(G) = 0 \\ 1, & \text{falls } \mathcal{A}(F) = 0 \text{ oder } \mathcal{A}(G) = 1. \end{cases}$$

   Zeigen Sie, dass folgendes Entscheidungsproblem NP-vollständig ist:

   <u>IMPLY-SAT</u>
   Eingabe: Eine $\rightarrow$-Formel.
   Frage: Ist F erfüllbar?

5. Passt die Belegung $\mathcal{A}$ zu Formeln F und G, dann gilt für die *Implikation* $\leftrightarrow$:

$$\mathcal{A}(F \leftrightarrow G) = \begin{cases} 0, & \text{falls } \mathcal{A}(F) \neq \mathcal{A}(G) \\ 1, & \text{falls } \mathcal{A}(F) = \mathcal{A}(G). \end{cases}$$

   Zeigen Sie, dass folgendes Entscheidungsproblem in P liegt:

   <u>IMPLY-SAT</u>
   Eingabe: Eine $\leftrightarrow$-Formel.
   Frage: Ist F erfüllbar?

Lösung

1. (Siehe Ergänzungsblatt 7, Aufgabe 5.1.)
2. (Siehe Ergänzungsblatt 7, Aufgabe 5.2.)

   Analog zum Beweis von NAND-SAT mit

$$F' = \begin{cases} F, & \text{falls } F \text{ atomar} \\ (G' \mathbin{\bar{\vee}} 0), & \text{falls } F = \neg G \text{ für eine boolesche Formel } G \\ ((G' \mathbin{\bar{\vee}} 0) \mathbin{\bar{\vee}} (H' \mathbin{\bar{\vee}} 0)), & \text{falls } F = G \wedge H \text{ für boolesche Formeln } G, H \\ ((G' \mathbin{\bar{\vee}} H') \mathbin{\bar{\vee}} 0), & \text{falls } F = G \vee H \text{ für boolesche Formeln } G, H. \end{cases}$$

3. Analog zum Beweis von NAND-SAT mit

$$F' = \begin{cases} F, & \text{falls } F \text{ atomar} \\ (G' \rightarrow 0), & \text{falls } F = \neg G \text{ für eine boolesche Formel } G \\ ((G' \rightarrow (H' \rightarrow 0)) \rightarrow 0), & \text{falls } F = G \wedge H \text{ für boolesche Formeln } G, H \\ ((G' \rightarrow 0) \rightarrow H'), & \text{falls } F = G \vee H \text{ für boolesche Formeln } G, H. \end{cases}$$

4. Analog zum Beweis von XOR-SAT.

Aufgabe 63

In der Vorlesung wurde gezeigt, dass folgendes Entscheidungsproblem für $k = 3$ NP-vollständig ist:

<u>kKNF-SAT</u>
Eingabe: Eine boolesche Formel F in KNF mit maximal k Literalen pro Klausel.
Frage: Ist F erfüllbar?

1. Zeigen Sie: 3KNF-SAT $\leq_p$ 5KNF-SAT.
2. Zeigen Sie: 5KNF-SAT $\leq_p$ 3KNF-SAT.

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 11, Aufgabe 6)

Aufgabe 64

Zeigen Sie für jedes $k \in \mathbb{N}$:

1. Das folgende Entscheidungsproblem liegt in P.

   <u>MAX-k-SAT</u>
   Eingabe: Eine boolesche Formel F.
   Frage: Gibt es ein Modell für F, das höchstens k Variablen in F mit 1 belegt?

2. Das folgende Entscheidungsproblem ist NP-vollständig.

   <u>MIN-k-SAT</u>
   Eingabe: Eine boolesche Formel F.
   Frage: Gibt es ein Modell für F, das mindestens k Variablen in F mit 1 belegt?

Lösung

(Siehe Ergänzungsblatt 8, Aufgabe 1.)

Aufgabe 65

Das Knotenfärbbarkeitsproblem (englisch *Vertex coloring problem*) besteht darin, die Knoten eines gegebenen Graphen so zu färben, dass keine zwei benachbarten Knoten mit derselben Farbe gefärbt werden.

Formal betrachten wir das folgende Entscheidungsproblem:

<u>Färbbarkeit</u>
Eingabe: Ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$.
Frage: Gibt es eine Abbildung $\varphi\colon V \to [k]$ mit $\varphi(u) \neq \varphi(v)$ für alle $\{u, v\} \in E$?

Des Weiteren sei k-Färbbarkeit das obige Problem für ein festes $k \in \mathbb{N}$. Aus der Vorlesung wissen wir, dass 3-Färbbarkeit NP-vollständig ist.

1. Zeigen Sie: 3-Färbbarkeit $\leq_p$ 4-Färbbarkeit.
2. Zeigen Sie: 4-Färbbarkeit $\leq_p$ 3-Färbbarkeit.

Lösung

(Siehe Ergänzungsblatt 8, Aufgabe 3.)

Aufgabe 66

Das Spezialisierungsproblem besteht darin, jeden Mitarbeiter eines Schichtbetriebs auf ein Arbeitsbereich zu spezialisieren, sodass in jeder Schicht für jeden Arbeitsbereich ein Spezialist vorhanden ist.

Formal betrachten wir das folgende Entscheidungsproblem:

SPEZIALISIERUNG
Eingabe: Endliche Mengen M, A und $S_1, \ldots, S_k \subseteq M$.
Frage: Gibt es eine Abbildung $s \colon M \to A$ mit $s(S_i) = A$ für alle $i \in [k]$?

Des Weiteren sei 2-SPEZIALISIERUNG das obige Problem für festes $k = 2$.

Zeigen Sie:

1. 2-SPEZIALISIERUNG $\in \mathsf{P}$.
2. SPEZIALISIERUNG ist NP-vollständig.

Lösung

(Siehe Ergänzungsblatt 8, Aufgabe 2.)

Aufgabe 67

Das folgende Problem besteht darin, für gegebene Listen von Studierenden, Lehrveranstaltungen, Terminen und Anmeldungen zu entscheiden, ob jeder Lehrveranstaltung ein Termin so zugeordnet werden kann, dass jeder Studierende einen überschneidungsfreien Stundenplan hat.

Formal definieren wir das Entscheidungsproblem STUNDENPLAN wie folgt:

STUNDENPLAN
Eingabe: Endliche Mengen S, L und T, sowie eine Relation $A \subseteq S \times L$.
Frage: Gibt es eine totale Funktion $g \colon L \to T$, sodass alle $\ell_1, \ell_2 \in L$ mit $\ell_1 \neq \ell_2$ und $\exists s \in S \colon (s, \ell_1), (s, \ell_2) \in A$ die Ungleichung $g(\ell_1) \neq g(\ell_2)$ erfüllen?

S, L, T und A stellen jeweils Mengen von Studierenden, Lehrveranstaltungen, Terminen und Anmeldungen dar. $(s, \ell) \in A$ besagt intuitiv, dass s sich für die Lehrveranstaltung ℓ angemeldet hat.

1. Zeigen Sie, dass STUNDENPLAN NP-vollständig ist.
2. Geben Sie eine Polynomialzeitreduktion von STUNDENPLAN auf SAT an.

Lösung

(Siehe Ergänzungsblatt 9, Aufgabe 1.)

Aufgabe 68

Sei $G = (V, E)$ ein Graph und $S \subseteq V$ eine beliebige Menge von Knoten. S heißt *stabil*, falls keine zwei Knoten aus S durch eine Kante verbunden sind, d. h. :

$$S \text{ stabil} \iff \forall u, v \in S \colon \{u, v\} \notin E.$$

Wir betrachten nun das folgende Problem:

<u>STABILITÄT</u>
Eingabe: Ein Graph G und eine Zahl $k \in \mathbb{N}$.
Frage: Gibt es eine stabile Menge S der Größe $|S| = k$?

Zeigen Sie, dass STABILITÄT NP-vollständig ist.

Lösung

(Siehe Ergänzungsblatt 11, Aufgabe 1.)

Aufgabe 69

Zeigen Sie, dass das folgende Entscheidungsproblem NP-vollständig ist:

Eingabe: Ein Graph $G = (V, E)$ und Zahlen $\ell, m \in \mathbb{N}$.
Frage: Gibt es eine Menge $S \subseteq V$ mit $|S| \leq \ell$ und $|\{e \in E \mid e \subseteq S\}| \geq m$?

Lösung

(Siehe Sommersemester 2018, Ergänzungsblatt 12, Aufgabe 3)

Aufgabe 70

Analog zur Definition von M_w bezeichne in dieser Aufgabe F_w die durch das Wort $w \in \{0, 1\}^*$ codierte boolesche Formel.

1. Zeigen Sie, dass folgende Sprache coNP-vollständig bezüglich $\leq_p$ ist:

$$\text{UNSAT} = \{w \mid F_w \text{ ist unerfüllbar}\}.$$

2. Die Komplexitätsklasse DP (*Difference Polynomial-Time*) sei wie folgt definiert:

$$\mathsf{DP} = \{A \setminus B \mid A, B \in \mathsf{NP}\}.$$

   (a) In welcher Beziehung stehen die den Klassen DP, NP und coNP zueinenader?

   (b) Geben Sie eine bezüglich $\leq_p$ DP-vollständige Sprache an und beweisen Sie die Korrektheit Ihrer Antwort.

Lösung

(Siehe Ergänzungsblatt 10, Aufgabe 1.)

Wichtige Beziehungen zwischen Komplexitätsklassen

(Vorlesungseinheiten 29-32)

Aufgabe 71

Eine *Sprachklasse* ist eine Menge von formalen Sprachen. Für eine Sprachklasse $\mathcal{C}$ sind $\mathsf{co}\mathcal{C}$ und $\overline{\mathcal{C}}$ ebenfalls Sprachklassen mit $\mathsf{co}\mathcal{C} = \{L \mid \overline{L} \in \mathcal{C}\}$ und $\overline{\mathcal{C}} = \{L \mid L \notin \mathcal{C}\}$.

1. Sei $\mathcal{C}$ eine beliebige Sprachklasse. Zeigen Sie:
   (a) $\mathsf{co}(\mathsf{co}\mathcal{C}) = \mathcal{C}$
   (b) $\overline{\mathsf{co}\mathcal{C}} = \mathsf{co}\overline{\mathcal{C}}$
2. Seien $\mathcal{C}_1$ und $\mathcal{C}_2$ beliebige Sprachklassen. Zeigen Sie:
$$\mathcal{C}_1 \subseteq \mathcal{C}_2 \iff \mathsf{co}\mathcal{C}_1 \subseteq \mathsf{co}\mathcal{C}_2.$$
3. Zeigen Sie, dass eine Sprachklasse $\mathcal{C}$ genau dann unter Komplement abgeschlossen ist, wenn $\mathcal{C} = \mathsf{co}\mathcal{C}$ gilt.
4. Geben Sie entweder eine Sprachklasse $\mathcal{C}$ mit der jeweiligen Eigenschaft an oder begründen Sie, warum keine solche Sprachklasse existieren kann.
   (a) $\mathcal{C} = \mathsf{co}\mathcal{C}$
   (b) $\mathcal{C} \subsetneq \mathsf{co}\mathcal{C}$
   (c) $\mathsf{co}\mathcal{C} \subsetneq \overline{\mathcal{C}}$
   (d) $\mathsf{co}\mathcal{C} \subsetneq \mathcal{C}$
   (e) $\overline{\mathcal{C}} = \mathsf{co}\mathcal{C}$
   (f) $\overline{\mathcal{C}} \subsetneq \mathsf{co}\mathcal{C}$

Lösung

(Siehe Ergänzungsblatt 9, Aufgabe 2.)

Aufgabe 72

Sei $f : \mathbb{N} \to \mathbb{N}$ eine beliebige Funktion. Welche der folgenden Aussagen implizieren welche? Beweisen Sie Ihre Antworten.

(a) f ist berechenbar

(b) f ist zeitkonstruierbar

(c) f ist platzkonstruierbar

Lösung

(Siehe Ergänzungsblatt 11, Aufgabe 2.)

Hierarchiesätze und die Sätze von Savitch, Immerman und Szelepcsényi

(Vorlesungseinheiten 32-36)

Aufgabe 73

Entscheiden Sie für jedes der gegebenen Klassenpaare, welche Klasse in der jeweils anderen als Teilmenge enthalten ist und welche nicht. Beweisen Sie Ihre Antworten.

1. $\mathsf{NTIME}(n^2)$ und $\mathsf{DSPACE}(n^3)$
2. $\mathsf{DTIME}(3n^2 + (\log n)^4)$ und $\mathsf{DTIME}(n^2 + 1)$
3. $\mathsf{coNSPACE}(2^n)$ und $\mathsf{DSPACE}(5^n)$
4. $\mathsf{NSPACE}\big(2^{(\log n)^2}\big)$ und $\mathsf{DSPACE}\big(n^{\mathcal{O}(\log n)}\big)$
5. $\mathsf{NTIME}(n)$ und $\mathsf{DTIME}(n^{n+2})$
6. $\mathsf{DSPACE}(n^3)$ und $\mathsf{NSPACE}(n)$
7. $\mathsf{DTIME}(n^2 + n \log n)$ und $\mathsf{DTIME}((n \log n)^2)$
8. $\mathsf{DSPACE}(2^{\mathcal{O}(n)})$ und $\mathsf{DSPACE}(3^{\mathcal{O}(n)})$
9. $\mathsf{NSPACE}((\log n)^2)$ und $\mathsf{DSPACE}(3^{(\log n)^3})$
10. $\mathsf{NSPACE}((\log n)^{\log n})$ und $\mathsf{NTIME}(n)$

Lösung

1. (Siehe Ergänzungsblatt 11, Aufgabe 4.1.)
2. (Siehe Ergänzungsblatt 11, Aufgabe 4.2.)
3. (Siehe Ergänzungsblatt 12, Aufgabe 1.1.)
4. (Siehe Ergänzungsblatt 12, Aufgabe 1.2.)
5. (Siehe Ergänzungsblatt 12, Aufgabe 1.3.)
6. (Siehe Sommersemester 2018, Ergänzungsblatt 12, Aufgabe 2.6)
7. (Siehe Sommersemester 2018, Ergänzungsblatt 12, Aufgabe 2.7)
8. (Siehe Sommersemester 2018, Ergänzungsblatt 12, Aufgabe 2.8)
9. (Siehe Sommersemester 2018, Ergänzungsblatt 12, Aufgabe 2.9)
10. (Siehe Sommersemester 2018, Ergänzungsblatt 12, Aufgabe 2.10)

Aufgabe 74

Begründen Sie die Inklusionsbeziehungen auf der letzten Seite im Foliensatz über Sprachenklassen.

Hinweis: Sie finden den Foliensatz unter

`https://www.fmi.uni-stuttgart.de/ti/teaching/s19/eti2/`.

Lösung

Bestimmt habe ich hier oder da was vergessen oder vertauscht. Bitte mit Vorsicht genießen und gefundene Fehler melden. Danke!

- Aus der Automatentheorie (letztes Semester) wissen wir:

$$\mathsf{FIN} \subseteq \mathsf{REG} \subseteq \mathsf{DCFL} \subseteq \mathsf{CFL} \subseteq \mathsf{CSL}. \qquad \text{und} \qquad \mathsf{CFL} \neq \mathsf{coCFL}.$$

- Wegen der polynomiellen Laufzeit des CYK-Algorithmus gilt $\mathsf{CFL} \subseteq \mathsf{P}$. Da aber P unter Komplement abgeschlossen ist, aber CFL nicht, können die zwei Klassen nicht gleich sein. Alternativ kann man $\mathsf{CFL} \neq \mathsf{P}$ mit dem Zeithierarchiesatz zeigen.
- Da jeder DEA durch eine DTM simuliert werden kann, die das Arbeitsband nicht verwendet, gilt:

$$\mathsf{REG} \subseteq \mathsf{DSPACE}(0) \subsetneq \mathsf{DSPACE}(\log n) = \mathsf{L}.$$

  Die zweite Inklusion ist nach dem Platzhierarchiesatz echt.
- DLBAs akzeptieren genau die Sprachen in $\mathsf{DSPACE}(n)$ und LBAs genau die Sprachen in $\mathsf{NSPACE}(n)$. Somit gilt

$$\mathsf{DCSL} = \mathsf{DSPACE}(n) \subseteq \mathsf{NSPACE}(n) = \mathsf{CSL}.$$

  Nach dem Satz von Savitch und dem Platzhierarchiesatz folgt

$$\mathsf{CSL} = \mathsf{NSPACE}(n) \subseteq \mathsf{DSPACE}(n^2) \subsetneq \mathsf{DSPACE}(n^3) \subseteq \mathsf{NPSPACE}$$

  und

$$\mathsf{NL} \subseteq \mathsf{DSPACE}((\log n)^2) \subsetneq \mathsf{DSPACE}(n) = \mathsf{DCSL}.$$

- Aus der Berechenbarkeitstheorie (erste Hälfte des aktuellen Semesters) wissen wir:

$$\mathsf{R} = \mathsf{RE} \cap \mathsf{coRE} \qquad \text{und} \qquad \mathsf{RE} \neq \mathsf{coRE}.$$

- Da R genau die Sprachen enthält, die man mit beliebig vielen Ressourcen entscheiden kann, gilt für jede Funktion $f\colon \mathbb{N} \to \mathbb{N}$:

$$\mathsf{DTIME}(f), \mathsf{NTIME}(f), \mathsf{DSPACE}(f), \mathsf{NSPACE}(f) \subseteq \mathsf{R}.$$

  Daraus folgt:

$$\mathsf{NEXPTIME} = \bigcup_{k \geq 1} \mathsf{NTIME}(2^{n^k}) \subseteq \mathsf{NTIME}(2^{n^n}) \subsetneq \mathsf{NTIME}(3^{n^n}) \subseteq \mathsf{R}$$

- Aus $\mathsf{DTIME}(f) \subseteq \mathsf{NTIME}(f) \subseteq \mathsf{DSPACE}(f)$ für $f(n) \geq n$ folgt

$$\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \qquad \text{und} \qquad \mathsf{EXPTIME} \subseteq \mathsf{NEXPTIME}.$$

- Aus $\mathsf{DSPACE}(f) \subseteq \mathsf{NSPACE}(f)$ für $f(n) \geq \log n$ folgt

$$\mathsf{L} \subseteq \mathsf{NL} \qquad \text{und} \qquad \mathsf{PSPACE} \subseteq \mathsf{NPSPACE}.$$

- Nach dem Satz von Savitch gilt für eine beliebige Sprache L:

$$\begin{aligned} L \in \mathsf{NPSPACE} &\implies \exists k \geq 1\colon L \in \mathsf{NSPACE}(n^k) \\ &\implies \exists k \geq 1\colon L \in \mathsf{DSPACE}((n^k)^2) = \mathsf{DSPACE}(n^{2k}) \\ &\implies \exists k' \geq 1\colon L \in \mathsf{DSPACE}(n^{k'}) \\ &\implies L \in \mathsf{PSPACE}. \end{aligned}$$

  Somit ist $\mathsf{PSPACE} = \mathsf{NPSPACE}$.

- Aus $\mathsf{NSPACE}(f) \subseteq \mathsf{DTIME}(2^{\mathcal{O}(f)})$ für $f(n) \geq \log n$ folgt:

  $\mathsf{NL} = \mathsf{NSPACE}(\log n) \subseteq \mathsf{DTIME}(2^{\mathcal{O}(\log n)}) = \bigcup_{k\geq 1} \mathsf{DTIME}(2^{k \log n}) = \bigcup_{k\geq 1} \mathsf{DTIME}(n^k) = \mathsf{P}.$

- Aus $\mathsf{NSPACE}(f) \subseteq \mathsf{DTIME}(2^{\mathcal{O}(f)})$ für $f(n) \geq \log n$ folgt zusammen mit $2^{\ell n^k} \in \mathcal{O}(2^{n^{k+1}})$ für alle $k, \ell \geq 1$:

$$\begin{aligned} \mathsf{NPSPACE} &= \bigcup_{k\geq 1} \mathsf{NSPACE}(n^k) \subseteq \bigcup_{k\geq 1} \mathsf{DTIME}(2^{\mathcal{O}(n^k)}) \\ &= \bigcup_{k,\ell\geq 1} \mathsf{DTIME}(2^{\ell n^k}) \subseteq \bigcup_{k\geq 1} \mathsf{DTIME}(2^{n^{k+1}}) = \mathsf{EXPTIME}. \end{aligned}$$

- Für Sprachklassen $\mathcal{C}, \mathcal{C}'$ mit $\mathcal{C} = \mathsf{co}\mathcal{C}$ gilt:

$$\mathcal{C} \subseteq \mathcal{C}' \iff \mathcal{C} = \mathsf{co}\mathcal{C} \subseteq \mathsf{co}\mathcal{C}' \qquad \text{und} \qquad \mathcal{C}' \subseteq \mathcal{C} \iff \mathsf{co}\mathcal{C}' \subseteq \mathsf{co}\mathcal{C} = \mathcal{C}.$$

  Aus der ersten Äquivalenz folgt $\mathsf{DCFL} \subseteq \mathsf{coCFL}$, $\mathsf{P} \subseteq \mathsf{coNP}$ und $\mathsf{EXPTIME} \subseteq \mathsf{coNEXPTIME}$, aber auch $\mathsf{CSL} \not\subseteq \mathsf{coCFL}$, $\mathsf{P} \not\subseteq \mathsf{coCFL}$ und $\mathsf{R} \not\subseteq \mathsf{coNEXPTIME}$. Aus der zweiten Äquivalenz folgt $\mathsf{coCFL} \subseteq \mathsf{CSL}$, $\mathsf{coCFL} \subseteq \mathsf{P}$, $\mathsf{coNP} \subseteq \mathsf{PSPACE}$ und $\mathsf{coNEXPTIME} \subseteq \mathsf{R}$, aber auch $\mathsf{coCFL} \not\subseteq \mathsf{DCFL}$.

- Nach dem Satz von Savitch und dem Platzhierarchiesatz gilt:

$$\mathsf{NL} \subseteq \mathsf{DSPACE}((\log n)^2) \subsetneq \mathsf{DSPACE}(n) \subseteq \mathsf{PSPACE}.$$

- Nach dem Zeithierarchiesatz gilt wegen $n^k \in \mathcal{O}(2^n)$ für alle $k \geq 1$:

$$\mathsf{P} \subseteq \mathsf{DTIME}(2^n) \subsetneq \mathsf{DTIME}(2^{n^2}) \subseteq \mathsf{EXPTIME}.$$

Der Lückensatz von Borodin

(Vorlesungseinheit 35)

Aufgabe 75

Welche der folgenden Implikationen gelten für beliebige Funktionen $f, g\colon \mathbb{N} \to \mathbb{N}$ und welche nicht? Beweisen Sie Ihre Antworten.

1. $\mathsf{DTIME}(f) = \mathsf{DTIME}(g) \implies f \in \Theta(g)$
2. $f \in \Theta(g) \implies \mathsf{DTIME}(f) = \mathsf{DTIME}(g)$

Hinweis: Es gilt $f \in \Theta(g)$ genau dann, wenn $f \in \mathcal{O}(g)$ und $g \in \mathcal{O}(f)$ gilt.

Lösung

(Siehe Ergänzungsblatt 12, Aufgabe 2.)

Aufgabe 76

Zeigen Sie, dass eine Funktion $f\colon \mathbb{N} \to \mathbb{N}$ mit $f \in \Omega(n)$ existiert, die nicht zeitkonstruierbar ist.

Lösung

Nach dem Satz von Borodin existiert eine Funktion $s\colon \mathbb{N} \to \mathbb{N}$ mit $s(n) \geq n+1$ und

$$\mathsf{DTIME}(s(n)) = \mathsf{DTIME}(2^{s(n)}).$$

Wegen $s(n) \geq n+1$ ist $s \in \Omega(n)$. Wir zeigen per Widerspruch, dass s nicht zeitkonstruierbar ist.

Angenommen, s ist zeitkonstruierbar. Dann folgt aus dem Zeithierarchiesatz wegen $s(n) \log s(n) \notin \Omega(2^{s(n)})$ und $2^{s(n)} \in \Omega(n \log n)$:

$$\mathsf{DTIME}(2^{s(n)}) \setminus \mathsf{DTIME}(s(n)) \neq \emptyset.$$

Dies steht im Widerspruch zu obiger Gleichung.

Die Translationssätze

(Vorlesungseinheit 37)

Aufgabe 77

In dieser Aufgabe betrachten wir die Klassen

$$\mathsf{DCSL} = \mathsf{DSPACE}(n) \qquad \text{und} \qquad \mathsf{CSL} = \mathsf{NSPACE}(n).$$

Zeigen Sie:

1. $\mathsf{NL} \subsetneq \mathsf{DCSL} \subseteq \mathsf{CSL} \subsetneq \mathsf{PSPACE}$
2. $\{\mathsf{DCSL}, \mathsf{CSL}\} \cap \{\mathsf{P}, \mathsf{NP}\} = \emptyset$

Lösung

(Siehe Ergänzungsblatt 12, Aufgabe 3.)

Aufgabe 78

Zeigen Sie folgende Implikation:

$$\mathsf{P} \subseteq \mathsf{L} \implies \mathsf{EXP} \subseteq \mathsf{PSPACE}.$$

Lösung

(Siehe Ergänzungsblatt 13, Aufgabe 1.)

Aufgabe 79

In dieser Aufgabe betrachten wir die Klasse

$$\mathsf{E} = \bigcup_{k \geq 1} \mathsf{DTIME}(2^{kn}).$$

Zeigen Sie:

1. $\mathsf{P} \subsetneq \mathsf{E} \subsetneq \mathsf{EXP}$
2. $\mathsf{CSL} \subseteq \mathsf{E}$
3. $\mathsf{E} \neq \mathsf{PSPACE}$

Hinweise:

- EXP wird auch $\mathsf{EXPTIME}$ genannt.
- $\mathsf{CSL} = \mathsf{NSPACE}(n)$.

Lösung

(Siehe Ergänzungsblatt 13, Aufgabe 2.)

Die Klasse NL

(Vorlesungseinheiten 38-39)

Aufgabe 80

Eine *starke Zusammenhangskomponente* (engl. *strongly-connected component*) eines gerichteten Graphen ist eine maximale Teilmenge C der Knoten, sodass jeder Knoten aus C von jedem anderen Knoten aus C erreichbar ist.

Zeigen Sie, dass das folgende Entscheidungsproblem NL-vollständig bezüglich $\leq_{\log}$ ist:

<u>SCC</u>
Eingabe: Ein gerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$.
Frage: Enthält G genau k starke Zusammenhangskomponenten?

Lösung

(Siehe Ergänzungsblatt 10, Aufgabe 2.)

Aufgabe 81

Sei $G = (V, E)$ ein gerichteter Graph. Ein *Pfad der Länge* ℓ ist ein Tupel $(v_0, \ldots, v_\ell) \in V^{\ell+1}$ mit $(v_{i-1}, v_i) \in E$ für alle $i \in [\ell]$. Ein Pfad, in dem die erste und letzte Komponente gleich sind, heißt *Kreis*. Kreise der Länge 0 nennt man *trivial*. Ein *gerichteter kreisfreier Graph* (engl. *directed acyclic graph*, kurz: *DAG*) ist ein gerichteter Graph, der keine nichttrivialen Kreise enthält.

Zeigen Sie, dass das folgende Entscheidungsproblem NL-vollständig bezüglich $\leq_{\log}$ ist:

DAGAP
Eingabe: Ein gerichteter Graph $G = (V, E)$ und zwei Knoten $s, t \in V$.
Frage: Ist G ein gerichteter kreisfreier Graph mit einem Pfad von s nach t?

Lösung

(Siehe Ergänzungsblatt 11, Aufgabe 3.)

Die Klasse PSPACE

(Vorlesungseinheit 40)

Aufgabe 82

Zeigen Sie, dass das folgende Problem PSPACE-vollständig bezüglich $\leq_{\log}$ ist:

LinearSpaceMembership
Eingabe: Eine 1-Band-TM $M = (Q, \Sigma, \Gamma, \delta, s, \square, F)$ und ein Wort $w \in \Sigma^*$.
Frage: Wird w von M akzeptiert, ohne dass ein Leersymbol $\square$ gelesen wird?

Lösung

(Siehe Ergänzungsblatt 13, Aufgabe 3.)

Alte Prüfungen

(Seit Wintersemester 2017/2018)

Aufgabe 83

Lösen Sie die Aufgaben der Modulprüfung *Berechenbarkeit und Komplexität* vom Wintersemester 2017/2018.

Lösung

1. Sei R die Klasse der entscheidbaren Sprachen.

   - Nein/Ja/Nein

     Die Sprache $L = \{w \mid \varepsilon \notin L(M_w)\}$ ist co-semi-entscheidbar, weil man durch die Simulation von M_w auf ε entscheiden kann, ob $\varepsilon \in L(M_w)$ gilt. Nach dem Satz von Rice ist L jedoch unentscheidbar und somit nicht semi-entscheidbar.

   - Ja/Ja/Ja

     Die Sprache $L = \{w \mid w \in L(M_w) \vee |w| \geq 3\}$ ist das Komplement einer endlichen Sprache und somit regulär.

   - Nein/Nein/Nein

     Man kann sowohl H_0 als auch $\overline{H_0}$ darauf reduzieren.

     Hinweis: Es ist eine gute Übung beide Reduktionen zu zeigen.

   - Ja/Ja/Ja

     Folgt direkt aus SAT $\in \mathsf{NP} \subseteq \mathsf{R}$.

   - Ja/Ja/Ja

     Folgt direkt aus $L \in \mathsf{NSPACE}(n^2) \subseteq \mathsf{R}$.

2. a) Sei L beliebig und M eine Turingmaschine mit $L(M) = L$. L existiert wegen $\mathcal{S} \neq \emptyset$ und M existiert, da $L \in \mathcal{R}$ semi-entscheidbar ist. Für die Reduktion wähle $f(x)$ als Kodierung einer Turingmaschine, die auf Eingabe w zuerst M_x auf ε und dann M auf w simuliert. f ist dann total und berechenbar und für alle Wörter x gilt:

$$\begin{aligned} x \in H_0 &\implies M_x \text{ hält auf } \varepsilon \\ &\implies L(M_{f(x)}) = L(M) \\ &\implies L(M_{f(x)}) \in \mathcal{S} \\ &\implies f(x) \in \mathcal{C}(\mathcal{S}) \end{aligned}$$

   und

$$\begin{aligned} x \notin H_0 &\implies M_x \text{ hält nicht auf } \varepsilon \\ &\implies L(M_{f(x)}) = \emptyset \\ &\implies L(M_{f(x)}) \notin \mathcal{S} \\ &\implies f(x) \notin \mathcal{C}(\mathcal{S}), \end{aligned}$$

   also $x \in H_0 \iff f(x) \in \mathcal{C}(\mathcal{S})$. □

   b) Sei $L = \{w \mid L(M_w) \in \mathsf{NSPACE}(n^2)\}$.

   Wegen $\emptyset \in \mathsf{NSPACE}(n^2)$ und $\mathsf{NSPACE}(n^2) \subsetneq \mathcal{R}$ gilt für $\mathcal{S} = \mathcal{R} \setminus \mathsf{NSPACE}(n^2)$ sowohl $\mathcal{S} \neq \emptyset$ als auch $\emptyset \notin \mathcal{S}$. Nach Teil a) gilt

$$H_0 \leq \{w \mid L(M_w) \in \mathcal{S}\} = \left\{w \,\middle|\, L(M_w) \notin \mathsf{NSPACE}(n^2)\right\} = \overline{L},$$

und somit $\overline{H_0} \leq L$. Da $\overline{H_0}$ nicht semi-entscheidbar ist, ist es L auch nicht. □

Wichtiger Hinweis:

In der Ergänzung wurde der folgende alternative Ansatz vorgeschlagen: Zeige mit dem Satz von Rice, dass L unentscheidbar ist und zeige danach, dass L co-semi-entscheidbar und demnach nicht semi-entscheidbar ist. Der Ansatz ist zwar korrekt, aber man kann die co-semi-Entscheidbarkeit von L nicht damit begründen, dass man eine Turingmaschine konstruieren kann, die auf Eingabe w ein Wort x rät, M_w auf x simuliert und akzeptiert, falls die Simulation mehr als $|x|^2$ Felder verwendet. Dass $L(M_w) \in \mathsf{NSPACE}(n^2)$ gilt heißt nicht, dass $\text{ntime}_{M_w}(x) \leq |x|^2$ für alle x gelten muss, sondern lediglich, dass M_w eine Sprache akzeptiert für die eine solche Turingmaschine existiert. M_w selber muss die Platzschranke nicht notwendigerweise einhalten.

3. a) f: Ja/Ja

   f wird vom Programm $x_0 := 1$; `LOOP` x_1 `DO` $x_0 := x_0 * 2$ `END` berechnet. Dieses kann sehr leicht von einem LOOP-Programm simuliert werden.

   g: Ja/Ja

   Da f LOOP-berechenbar ist, kann die Anweisung $x_0 := f(x_2) - x_1$ von einem LOOP-Programm simuliert werden.

   μg: Ja/Ja

   Es gilt $\mu g(y) = \min\{n \in \mathbb{N} \mid f(y) \dot{-} n = 0\} = \min\{n \in \mathbb{N} \mid f(y) \leq n\} = f(y)$, also $\mu g = f$.

   h: Ja/Ja

   Da f LOOP-berechenbar ist, kann die Anweisung $x_0 := x_1 + f(x_2)$ von einem LOOP-Programm simuliert werden.

   μh: Ja/Nein

   Wegen $\{n \in \mathbb{N} \mid n + f(y) = 0\} = \emptyset$ ist $\mu h = \Omega$ nicht total und somit nicht LOOP-berechenbar.

   a: Ja/Nein

   Wurde in der Vorlesung bewiesen.

   b: Ja/Nein

   Wäre b LOOP-berechenbar, dann wäre auch $a(x, y) = b(x, y) + x$ LOOP-berechenbar, was nicht stimmt.

   b)
   - Es gilt: $g(10, 3) = f(3) \dot{-} 10 = 2^3 \dot{-} 10 = 8 \dot{-} 10 = 0$.
   - Es gilt: $\mu g(10) = f(10) = 2^{10} = 1024$.
   - Wegen $\mu h = \Omega$ ist μh an der Stelle 10 nicht definiert.

4. a) Die Aussage ist richtig. Es gilt:

$$\mathsf{NSPACE}(n^3) \overset{(1)}{\subseteq} \mathsf{DSPACE}(n^6) \overset{(2)}{\subsetneq} \mathsf{DSPACE}(n^6 \log n).$$

Erläuterungen

(1) Nach dem Satz von Savitch gilt wegen $n^3 \in \Omega(\log n)$: $\mathsf{NSPACE}(n^3) \subseteq \mathsf{DSPACE}(n^6)$.

(2) Aus $n^6 \in \mathcal{O}(n^6 \log n)$ folgt $\mathsf{DSPACE}(n^6) \subseteq \mathsf{DSPACE}(n^6 \log n)$ und aus $n^6 \notin \Omega(n^6 \log n)$ und $n^6 \log n \in \Omega(\log n)$ folgt nach dem Platzhierarchiesatz: $\mathsf{DSPACE}(n^6) \not\supseteq \mathsf{DSPACE}(n^6 \log n)$.

b) Die Aussage ist falsch. Angenommen, die Aussage ist richtig. Dann folgt:

$$\begin{aligned}\mathsf{DTIME}(n) &= \mathsf{NSPACE}(n) = \mathsf{NSPACE}(\mathcal{O}(n)) \\ &= \bigcup_{c \in \mathbb{N}} \mathsf{NSPACE}(cn) = \bigcup_{c \in \mathbb{N}} \mathsf{DTIME}(cn) = \mathsf{DTIME}(\mathcal{O}(n)),\end{aligned}$$

was bekanntlich falsch ist.

5. (Siehe Sommersemester 2018, Ergänzungsblatt 14, Aufgabe 1.4)

6. (Siehe Sommersemester 2018, Ergänzungsblatt 15, Aufgabe 1.9)

Aufgabe 84

Lösen Sie die Aufgaben der Modulprüfung *Berechenbarkeit und Komplexität* bzw. *Theoretische Informatik II* vom Sommersemester 2018.

Lösung

1. Sei R die Klasse der entscheidbaren Sprachen.

   - Nein/Ja/Nein

     Der Satz von Rice ist zwar nicht anwendbar, aber mit der Reduktion aus dem Beweis des Satzes kann $\overline{H_0}$ auf die Sprache reduziert werden.

     Ein Semi-Entscheidungsalgorithmus simuliert M_w auf ww und gibt 1 zurück, wenn die Simulation eine akzeptierende Kofiguration erreicht.

   - Ja/Ja/Ja

     Für jede TM existiert eine unterschiedliche TM (sogar unendlich viele), die dieselbe Sprache akzeptiert. Demnach ist die gegebene Sprache genau $\{0,1\}^*$ und somit entscheidbar.

   - Nein/Nein/Nein

     Man kann sowohl H_0 als auch $\overline{H_0}$ darauf reduzieren.

     Hinweis: Es ist eine gute Übung beide Reduktionen zu zeigen.

- Ja/Ja/Ja

  Es gilt: $\mathrm{QBF} \in \mathsf{PSPACE} \subseteq \mathsf{R}$.

- Nein/Ja/Nein

  Ist aus der Vorlesung bekannt, dass MPCP semi-, aber nicht co-semi-entscheidbar ist.

2. Es gilt $L = \mathcal{C}(\mathcal{S})$ für

$$\mathcal{S} = \left\{ f \in \mathcal{R} \,\middle|\, D_f \neq \emptyset \wedge \forall x \in D_f \colon x^2 \in D_f \wedge x^3 \notin D_f \right\}.$$

   Man sieht, dass die Funktion $f \colon \Sigma^* \to \Sigma^*$ mit

$$f(w) = \begin{cases} w, & \text{falls } |w| \text{ eine Zweierpotenz ist} \\ \perp & \text{sonst} \end{cases}$$

   in $\mathcal{S}$ enthalten ist. Wegen $\Omega \in \mathcal{R} \setminus \mathcal{S}$ ist $\mathcal{S}$ nichttrivial und L nach dem Satz von Rice unentscheidbar.

3. Wir zeigen, dass keine Lösung existiert. Angenommen, es existiert eine Lösung. Sei $s = (i_1, \ldots, i_n)$ eine Lösung minimaler Länge n. Für diese muss offensichtlich $i_1 = i_n = 1$ gelten. Da $||x_i| - |y_i|| \leq 1$ für alle $i \in \{1, 2, 3\}$ gilt, gibt es wegen $|x_{i_1}| = |y_{i_1}| + 1$ und $|x_{i_1} \ldots x_{i_{n-1}}| = |y_{i_1} \ldots y_{i_{n-1}}| - 1$ ein $k \in \{2, \ldots, n-1\}$, sodass $(i_1, \ldots, i_k)$ ebenfalls eine Lösung ist. Das ist ein Widerspruch zur Minimalität der Länge von s.

4. - Ja

     Multiplikation und (modifizierte) Differenz sind LOOP-berechenbar, somit auch $f_1(n) = n \cdot n \cdot n \dot{-} n$.

   - Ja

     Es gilt $f_2(0) = a(1, 0) + 1 = a(0, 1) + 1 = 2$ und $f_2(n + 1) = a(1, n + 1) + 1 = a(0, a(1, n)) + 1 = a(1, n) + 2 = f_2(n) + 1$. Somit ist $f_2(n) = n + 2$ trivialerweise LOOP-berechenbar.

   - Ja

     Die Komposition von zwei LOOP-berechenbaren Funktionen ist wieder LOOP-berechenbar. Da $f \colon \mathbb{N} \to \mathbb{N}$, $f(n) = 2^n$ bekanntlich LOOP-berechenbar ist, ist es $f_3 = f \circ f$ auch.

   - Nein

     Wäre χ_H LOOP-berechenbar, dann wäre χ_H (Turing-)berechenbar und H somit entscheidbar.

5. Möglicher Ansatz:

   (1) Zeige mit den Translationssätzen die Implikation

$$\mathsf{DTIME}(2^n) \subseteq \mathsf{PSPACE} \implies \mathsf{DTIME}\big(2^{n^2}\big) \subseteq \mathsf{PSPACE}.$$

(2) Nimm an, dass $\mathsf{DTIME}(2^n) = \mathsf{PSPACE}$ gilt und folgere daraus, dass die Inklusionskette $\mathsf{DTIME}\big(2^{n^2}\big) \subseteq \mathsf{PSPACE} \subseteq \mathsf{DTIME}(2^n)$ gilt.

(3) Begründe mit dem Zeithierarchiesatz, warum dies ein Widerspruch ist.

6. Pseudocode:

Eingabe: Erfüllbare SAT-Instanz F mit Variablen $x_1, \ldots, x_m$.
Ausgabe: Belegung $\mathcal{A}\colon \{x_1, \ldots, x_m\} \to \{0,1\}$ mit $\mathcal{A}(F) = 1$.

```
𝒜 := ∅
for i = 1, ..., m
    if A(F ∧ x_i) = 1
        𝒜 := 𝒜 ∪ {(x_i, 1)}
        F := F ∧ x_i
    else
        𝒜 := 𝒜 ∪ {(x_i, 0)}
        F := F ∧ ¬x_i
return 𝒜
```

7. a) Pseudocode:

Eingabe: DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$.

```
p := s
rate nichtdeterministisch ein q ∈ Q
while p ≠ q
    rate nichtdeterministisch ein a ∈ Σ
    p := δ(p, a)
rate nichtdeterministisch ein a ∈ Σ
p := δ(p, a)
while p ≠ q
    rate nichtdeterministisch ein a ∈ Σ
    p := δ(p, a)
while p ∉ F
    rate nichtdeterministisch ein a ∈ Σ
    p := δ(p, a)
akzeptiere
```

Korrektheit

Der Algorithmus akzeptiert die Eingabe $\mathcal{A}$ genau dann, wenn ein Zustand $q_i \in Q$ und Wörter $u, v, w \in \Sigma^*$ existieren mit $|v| \geq 1$, $\hat{\delta}(s, u) = q_i$, $\hat{\delta}(q_i, v) = q_i$ und $\hat{\delta}(q_i, w) \in F$. Dies ist genau dann der Fall, wenn die von $\mathcal{A}$ akzeptierte Sprache unendlich ist.

Einhaltung der Platzschranke

Seien O. B. d. A. $Q = [\ell]$ und $\Sigma = [m]$ und sei n die Länge der Codierung von $\mathcal{A}$. Der Algorithmus verwendet nur die Variablen a, p und q. Wegen $a, p, q \leq n$

können ihre Werte bei Verwendung der Binärdarstellung in $\mathcal{O}(\log n)$ Platz gespeichert werden.

b) Sei $\mathsf{DFA\text{-}FIN}$ das gegebene Problem. Wir zeigen $\overline{\mathrm{GAP}} \leq_{\log} \mathsf{DFA\text{-}FIN}$. Für eine Codierung x einer GAP-Instanz (G, s, t) (O. B. d. A. mit $G = ([m], E)$) definiere $f(x) = \langle \mathcal{A} \rangle$ für $\mathcal{A} = (Q, \Sigma, \delta, s, \{t\})$ mit $Q = [m+1]$, $\Sigma = [m]$ und $\delta \colon Q \times \Sigma \to Q$ mit

$$\delta(u, v) = \begin{cases} v, & \text{falls } u \leq m \text{ und } (u, v) \in E \cup \{(t, s)\} \\ m+1 & \text{sonst.} \end{cases}$$

Aus der Konstruktion folgt, dass im DEA $\mathcal{A}$ genau dann ein Übergang von u nach v existiert, wenn $(u, v) \in E$ oder $(u, v) = (t, s)$ gilt.

f ist offensichtlich total und in logarithmischem Platz berechenbar. Zu zeigen ist für jede Codierung x einer GAP-Instanz (G, s, t) mit $G = (V, E)$ die Äquivalenz

$$x \in \overline{\mathrm{GAP}} \iff f(x) \in \mathsf{DFA\text{-}FIN}.$$

$\Longrightarrow$

Sei $x \in \overline{\mathrm{GAP}}$, d. h. $x \notin \mathrm{GAP}$. Dann existiert in G kein Weg von s nach t und somit auch kein Wort $w \in \Sigma^*$ mit $\hat{\delta}(s, w) = t$. Dann wird kein Wort von $\mathcal{A}$ akzeptiert, d. h. es gilt $f(x) \in \mathsf{DFA\text{-}FIN}$.

$\Longleftarrow$

Sei $x \notin \overline{\mathrm{GAP}}$, d. h. $x \in \mathrm{GAP}$. Dann existiert in G ein Weg von s nach t und somit ein Wort $w \in \Sigma^*$ mit $\hat{\delta}(s, w) = t$. Dann gilt $\hat{\delta}(s, ws) = \delta(\hat{\delta}(s, w), s) = \delta(t, s) = s$ und somit $\hat{\delta}(s, (ws)^k w) = t$ für alle $k \in \mathbb{N}$. Dann werden unendlich viele Wörter von $\mathcal{A}$ akzeptiert, d. h. es gilt $f(x) \notin \mathsf{DFA\text{-}FIN}$.

Bemerkung: Da NL unter Komplement abgeschlossen ist, ist das Komplement eines NL-vollständiges Problem wieder NL-vollständig. Insbesondere ist also $\overline{\mathrm{GAP}}$ NL-vollständig.

8. falsch, falsch, falsch, wahr, wahr, falsch, wahr, falsch, wahr, wahr.

Aufgabe 85

Lösen Sie die Aufgaben der Modulprüfung *Berechenbarkeit und Komplexität* bzw. *Theoretische Informatik II* vom Wintersemester 2018/2019.

Lösung

1. Sei L das gegebene Problem.

   a) Ja.

   χ'_L kann wie folgt berechnet werden:

Eingabe:	TM M und Zahl $\ell \in \mathbb{N}$.

```
for t = 0, 1, 2, ...
    for all w ∈ Σ^ℓ
        simuliere t Schritte von M auf w
        if M akzeptiert
            return 1
```

b) Nein.

Man kann H_0 auf L mittels $f(w) = \langle M_w, 0 \rangle$ reduzieren. Für alle $w \in \{0,1\}^*$ gilt nämlich:

$$\begin{aligned} w \in H_0 &\iff M_w \text{ hält auf } \varepsilon \\ &\iff M_w \text{ hält auf mindestens einem Wort der Länge } 0 \\ &\iff f(w) \in L. \end{aligned}$$

2. • Ja/Nein/Nein

Diese Sprache ist semi-entscheidbar, weil man durch Simulation von M_w auf u entscheiden kann, ob $U \in T(M_w)$ gilt. Nach dem Satz von Rice ist sie jedoch unentscheidbar und somit nicht co-semi-entscheidbar.

• Ja/Ja/Ja

Diese Sprache ist das Komplement einer endlichen Sprache. Da jede endliche Sprache regulär ist, die Klasse der regulären Sprachen unter Komplement abgeschlossen ist und jede reguläre Sprache entscheidbar ist, ist die gegebene Sprache entscheidbar.

• Nein/Nein/Nein

Man kann sowohl H_0 als auch $\overline{H_0}$ darauf reduzieren.

Hinweis: Es ist eine gute Übung beide Reduktionen zu zeigen.

• Ja/Ja/Ja

Auch diese Sprache ist das Komplement einer endlichen Sprache.

3. • Ja

Potentieren und (modifizierte) Differenz sind LOOP-berechenbar und somit auch f_1.

Bemerkung: Man könnte hier auch damit argumentieren, dass die Funktion an der Stelle 0 nicht definiert und somit nicht total ist. In diesem Fall wäre *Nein* die richtige Antwort. Für mehr Informationen zur Frage, ob der Audruck 0^0 definiert ist oder nicht siehe beispielsweise

`https://en.wikipedia.org/wiki/Zero_to_the_power_of_zero`.

• Ja

Es gilt $f_2(n) = 1$, d. h. $f = c_1^1$.

- Ja

  Es gilt $f_3(n) = 2^n$.

- Ja

  Man kann sich überlegen warum die Funktionen $\text{mod}_2 \colon \mathbb{N} \to \mathbb{N}$, $n \mapsto n \bmod 2$ und $\text{div}_2 \colon \mathbb{N} \to \mathbb{N}$, $n \mapsto \lfloor \frac{n}{2} \rfloor$ LOOP-berechenbar sind und wie mit ihnen zeigen kann, dass die gegebene Funktion das ebenfalls ist.

4. falsch, wahr, falsch, wahr, wahr, wahr.

5. a) RE

   Aus der Vorlesung wissen wir, dass PCP semi-entscheidbar, aber nicht entscheidbar ist.

   b) PSPACE

   Man sieht leicht, dass TAUT coNP-vollständig ist. Wegen $\mathsf{coNP} \subseteq \mathsf{PSPACE}$ ist TAUT $\in \mathsf{PSPACE}$. Wäre TAUT $\in \mathsf{NP}$, dann wären NP und coNP gleich, was nach Aufgabenstellung ausgeschlossen werden darf.

   c) L

   1-GAP kann von einer logarithmisch platzbeschränkten DTM entschieden werden, die wie folgt arbeitet:

   Eingabe: Gerichteter Graph $G = (V, E)$ und Knoten $s, t \in V$.

   ```
   u := s
   while u ≠ t
       for all v ∈ V
           if (u, v) ∈ E
               u := v
   akzeptiere
   ```

6. *Hinweis*: Die Begriffe *NP-schwierig* und *NP-hart* sind synonym.

   a) Ja.

   Sei $L \in \mathsf{NP}$ beliebig. Dann existiert eine polynomiell Zeitbeschränkte NTM $M = (Q, \Sigma, \Gamma, \delta, s, \square, F)$ mit $T(M) = L$. Mittels $f(w) = \langle M \rangle \# w$ kann L auf das Halteproblem H reduziert werden. Diese Reduktionsfunktion ist total und kann in konstantem Platz (und somit in polynomieller Zeit) berechnet werden.

   b) $\Longrightarrow$

   Angenommen, es gilt $\mathsf{P} = \mathsf{NP}$. Dann ist jede nichttriviale Sprache NP-schwierig, insbesondere auch jede nichttriviale Sprache aus R.

   $\Longleftarrow$

   Angenommen, es gilt $\mathsf{P} \neq \mathsf{NP}$. Wähle $L \in \mathsf{P}$ beliebig, aber nichttrivial (z. B. $L = \{a\}$). Wegen $\mathsf{P} \subseteq \mathsf{R}$ ist L entscheidbar. Wegen $\mathsf{P} \neq \mathsf{NP}$ kann L nicht

NP-schwierig sein.

7. Sei L das gegebene Problem.

   a) Eine NTM kann (1) eine Eingabe w der Länge ℓ raten und dann (2) t Schritte von M auf w simulieren. Wird w nicht innerhalbt dieser Zeit akzeptiert, dann geht die NTM in einen Endzustand.

      Sei n die Länge der Eingabecodierung. Da die Zahlen ℓ und t unär kodiert sind, gilt $\ell, t \leq n$. Somit benötigt Schritt (1) Linearzeit und (2) Polynomialzeit in n.

   b) (Diese Aufgabe bezieht sich auf Übungsblatt 5, Präsenzübung 2 vom Sommersemester 2018.)

8. Möglicher Ansatz:

   (1) Zeige mit den Translationssätzen die Implikation

   $$\mathsf{DTIME}(n^3) \subseteq \mathsf{NL} \implies \mathsf{DTIME}(n^6) \subseteq \mathsf{NL}.$$

   (2) Nimm an, dass $\mathsf{DTIME}(n^3) = \mathsf{NL}$ gilt und folgere daraus, dass die Inklusionskette $\mathsf{DTIME}(n^6) \subseteq \mathsf{NL} \subseteq \mathsf{DTIME}(n^3)$ gilt.

   (3) Begründe mit dem Zeithierarchiesatz, warum dies ein Widerspruch ist.