Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 2022

66115

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Theoretische Informatik, Algorithmen**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 13

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Algorithmen

Aufgabe 1 (Asymptotische Notation) [25 PUNKTE]

Sortieren Sie die folgenden Funktionen nach ihrem asymptotischen Wachstum, sodass $f_i(n) \in O(f_{i+1}(n))$ gilt. Markieren Sie auch, ob $f_i(n) \in \Theta(f_{i+1}(n))$ gilt.

$$n\log_2(2^n), \quad 4n^2, \quad n\sqrt{n^3}, \quad 2^{\sqrt{n}}, \quad n^2\log_2(n), \quad \frac{n^2}{\log_2(n)}$$

Aufgabe 2 (Laufzeit) [15 PUNKTE]

Gegeben ist der folgende Algorithmus (in Java-Notation):

```
1 boolean func(int n) {
2     for (int i = 2; i * i <= n; i += 1) {
3         if (n % i == 0)
4             return false;
5     }
6     return true;
7 }
```

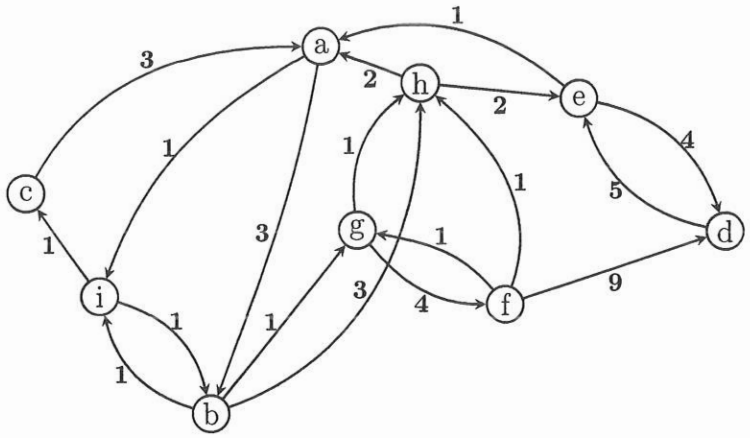
Hinweis: Das Zeichen % steht für den Modulo-Operator.

a) Beschreiben Sie, was dieser Algorithmus bei der Eingabe der ganzen Zahl n allgemein berechnet.

b) Welche bestmögliche asymptotische obere Schranke besitzt die Worst-Case-Laufzeit dieses Algorithmus in Abhängigkeit des Parameters n? Begründen Sie Ihre Antwort!

Fortsetzung nächste Seite!

Aufgabe 3 (Dijkstra) [30 PUNKTE]



Eine Iteration in Dijkstras Algorithmus entfernt genau einen Knoten aus der Priority-Queue. Angenommen nach vier Iterationen auf dem obigen Graphen sieht der Inhalt der Priority-Queue wie folgt aus:

$(a, 3), (d, 7), (b, \infty), (i, \infty), (c, \infty)$

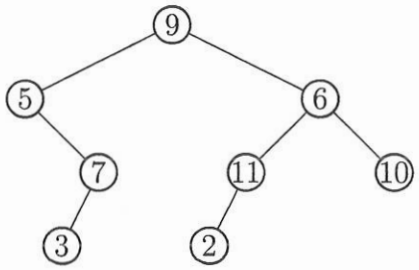
a) Nennen Sie den Startknoten, von dem der Algorithmus gestartet wurde! Begründen Sie Ihre Antwort!

b) Führen Sie die restlichen Iterationen durch und geben Sie jeweils den Inhalt der Priority-Queue nach jeder Iteration an!

Fortsetzung nächste Seite!

Aufgabe 4 (Algorithmenentwurf) [50 PUNKTE]

a) Geben Sie die Postorderreihenfolge des Baums an!



b) Zeigen Sie durch Angabe eines Gegenbeispiels, dass aus der Postorderreihenfolge der Knoten eines binären Baums (mit paarweise verschiedenen Schlüsseln) die Struktur des Baums nicht eindeutig rekonstruiert werden kann.

c) Im Folgenden sind die Postorder- und Inorderreihenfolge eines binären Baums gegeben. Nehmen Sie an, dass die Schlüssel der Knoten paarweise verschieden sind. Ziel ist es nun einen Algorithmus zu implementieren, der die eindeutige Struktur des (binären) Baumes wiedergibt.

- Überlegen Sie sich zunächst, wo die Wurzel innerhalb der Postordertravesierung zu finden ist.

- Bestimmen Sie nun die Postorder- und Inordertravesierung des rechten und linken Teilbaums.

- Entwickeln Sie nun den Algorithmus und geben Sie ihn in einer objektorientierten Programmiersprache Ihrer Wahl oder in Pseudocode an.
  Implementieren Sie dazu eine Methode *construct*, die ein Array *post* mit der Postorderreihenfolge und ein Array *in* mit der Inorderreihenfolge bekommt (und ggf. die Länge dieser Arrays) und den zugehörigen Baum berechnet.

Fortsetzung nächste Seite!

Teilaufgabe II: Theoretische Informatik

Aufgabe 1 (Gemischte Fragen) [35 PUNKTE]

Geben Sie für jede der folgenden Aussagen an, ob sie wahr oder falsch sind. Beweisen Sie Ihre Antwort.

a) Für jede Sprache L gilt $L^+ = L^\star \setminus \{\varepsilon\}$.

b) Sei L eine Sprache und $\equiv_L$ die zugehörige Myhill-Nerode Relation. Wenn L regulär ist, so sind alle Äquivalenzklassen von $\equiv_L$ endlich.

c) Die Sprache $L = \{a^n b^m \mid n, m \in \mathbb{N}; n \neq m\}$ ist kontextfrei.
Hinweis: $\mathbb{N}$ ={1, 2, 3,...}

d) Sei M eine Turingmaschine und $x \in \Sigma^\star$ ein Eingabewort. Sei

$$L = \begin{cases} \Sigma^\star & M \text{ hält auf } x \\ \emptyset & \text{sonst.} \end{cases}$$

Dann ist die Sprache L entscheidbar.

e) Wenn $P = NP$ gilt, so sind alle Probleme aus P NP-vollständig.

f) Angenommen es gibt eine polynomielle Reduktion f von einem NP-vollständigem Problem A auf ein Problem B. Dann folgt aus der Existenz von f, dass B in NP liegt.

g) Angenommen es gibt eine polynomielle Reduktion f von einem NP-vollständigem Problem A auf ein Problem B. Dann folgt aus der Existenz von f, dass B NP-schwer ist.

Aufgabe 2 (Reguläre Sprachen) [25 PUNKTE]

a) Geben Sie einen deterministischen endlichen Automaten (DEA) für die von folgendem regulären Ausdruck dargestellte Sprache über dem Alphabet $\Sigma = \{a, b\}$ an:

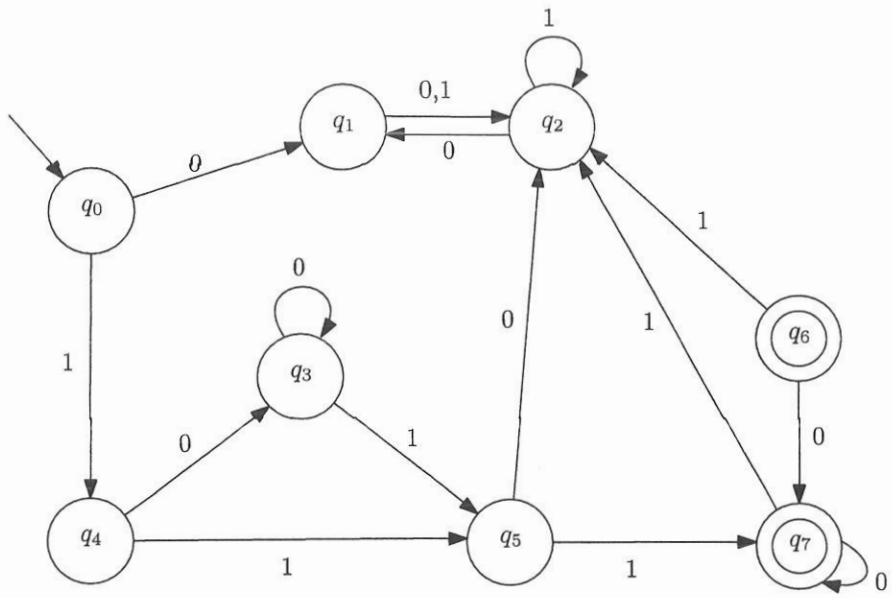
$$a^\star \cup (ab)^\star$$

(Alternative Schreibweise: $a^\star + (ab)^\star$)

Hinweis: Sie können zuerst einen nicht-deterministischen endlichen Automaten (NEA) konstruieren und diesen dann in einen DEA umwandeln.

b) Geben Sie einen DEA mit minimaler Anzahl an Zuständen an, der die selbe Sprache akzeptiert wie folgender DEA. Dokumentieren Sie Ihr Vorgehen geeignet.

Fortsetzung nächste Seite!



c) Sei L eine reguläre Sprache über dem Alphabet Σ. Zeigen Sie, dass dann die Sprache $L' = \{w_1 w_2 \in \Sigma^* \mid w_1 \in L, w_2 \in \Sigma^*\}$ regulär ist.

Aufgabe 3 (Chomsky-Hierarchie) [30 PUNKTE]

Bestimmen Sie für die folgenden Sprachen jeweils, ob sie regulär sind und ob sie kontextfrei sind. Beweisen Sie Ihre Antworten. Für „Ja"-Antworten genügt es eine geeignete Beschreibung (Grammatik/regulärer Ausdruck) oder einen geeigneten Akzeptor (Automat) ohne Korrektheitsbeweis anzugeben.

a) $L_1 = \{a^n b^m \mid n, m \in \mathbb{N}; n - m \geq 100\}$

b) $L_2 = \{a^n b^m \mid n, m \in \mathbb{N}; n \cdot m \geq 100\}$

c) $L_3 = \{a^n b^m c^l \mid n, m, l \in \mathbb{N}; n \cdot m = l\}$

Hinweis: $\mathbb{N}$ ={1, 2, 3,...}

Aufgabe 4 (Entscheidbarkeit) [30 PUNKTE]

Entscheiden Sie für die folgenden Sprachen, ob sie entscheidbar und ob sie semi-entscheidbar sind. Beweisen Sie jeweils Ihre Antwort. $\langle M \rangle$ bezeichnet dabei die Gödelnummer der Turingmaschine M.

a) $L_1 = \{\langle M_1 \rangle \# \langle M_2 \rangle \mid \varepsilon \in L(M_1) \cap L(M_2)\}$

b) $L_2 = \{\langle M_1 \rangle \# \langle M_2 \rangle \mid \varepsilon \in L(M_1) \setminus L(M_2)\}$

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Algorithmen

Aufgabe 1 (Doppelt-verkettete Listen) [34 PUNKTE]

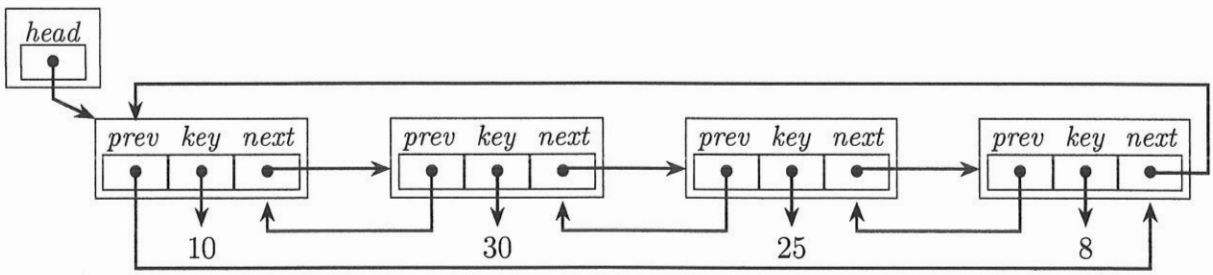
Elemente einer zyklischen, doppelt-verketteten Liste seien durch Objekte vom Typ *Element* repräsentiert, wobei jedes Element E die Attribute

- $E.key$ für den Schlüsselwert des Elements,
- $E.next$ für das nächste Element in der Liste und
- $E.prev$ für das vorherige Element in der Liste

besitzt. Aufgrund der zyklischen Verkettung gilt: Für das letzte Listenelement zeigt *next* auf das erste Listenelement und für das erste Listenelement zeigt *prev* auf das letzte Listenelement.

Eine Liste L wird durch das Attribut $L.head$ repräsentiert, welches auf das erste Listenelement zeigt. Eine leere Liste wird dadurch repräsentiert, dass $L.head = null$ gilt.

Eine Illustration für die Liste mit Schlüsselwerten $10, 30, 25, 8$ ist



Geben Sie für jede der folgenden Operationen einen Algorithmus in Pseudo-Code an, welcher die jeweilige Operation durchführt. Schätzen Sie die jeweilige Laufzeit ab und erläutern Sie die Funktionsweise Ihrer Algorithmen.

a) Aneinanderhängen zweier zyklischer doppelt-verketteter Listen L_1, L_2.

b) Verdoppeln aller Schlüsselwerte mit Schlüsselwert k

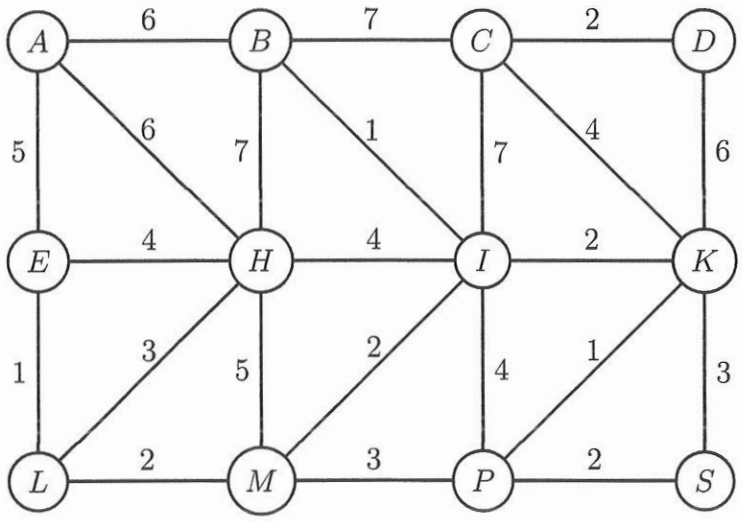
c) Löschen aller Listenelemente mit Schlüsselwert k

d) Umdrehen der Reihenfolge der Listenelemente

Fortsetzung nächste Seite!

Aufgabe 2 (Minimale Spannbäume) [20 PUNKTE]

Finden Sie einen minimalen Spannbaum für den folgenden Graphen, indem Sie den Algorithmus von Kruskal per Hand ausführen. Erläutern Sie die Funktionsweise des Kruskal-Algorithmus allgemein und dokumentieren Sie die Ausführung des Algorithmus für den Graphen schrittweise.



Fortsetzung nächste Seite!

Aufgabe 3 (Laufzeitanalysen, Komplexität, O-Notation) [31 PUNKTE]

a) Ordnen Sie die folgenden Funktionen in der Reihenfolge ihres asymptotischen Wachstums, so dass $f_i \in O(f_{i+1})$ gilt.

$$n!, \quad \log_2(5n^2), \quad n^{42}, \quad 2n\log_2(5n), \quad \sqrt{n}\log_2(n^4)$$

b) Sei A ein Feld mit $n \geq 2$ Einträgen $A[1], \ldots A[n]$. Bestimmen Sie für jeden der folgenden in Pseudo-Code gegebenen Algorithmen mithilfe der Laufzeitanalyse eine möglichst gute obere Schranke (in O-Notation) für dessen Laufzeit in Abhängigkeit des Parameters n.

Begründen Sie Ihre Antwort jeweils.

(i)
```
for i := 1; i ≤ n; i := i + 1 do
    A[i] = i;
end-for
for j := 1; j ≤ n; j := j + 1 do
    for k := 1; k ≤ n; k := k + 1 do
        A[j] = A[j] + A[k];
    end-for
end-for
```

(ii)
```
for i := 2; i ≤ n; i := i * i do
    A[i − 1] := A[i − 1] + 1;
end-for
```

(iii)
```
for i := 0; i ≤ n^2; i := i + n do
    A[1] = A[1] + 1;
end-for
```

(iv)
```
for i := 1; i ≤ n; i := i + 1 do
    for j := 1; j ≤ n; j := j * 2 do
        A[j] := A[i] + A[j];
    end-for
end-for
```

Fortsetzung nächste Seite!

Aufgabe 4 (Äpfel mit Birnen vergleichen) [35 PUNKTE]

Gegeben sei eine geordnete Folge von Früchten, welche aufsteigend nach ihrem Gewicht angeordnet sind. Früchte können Äpfel und Birnen sein. Gesucht ist jenes Paar, bestehend aus einem Apfel und einer Birne, dessen Gewichtsunterschied am kleinsten ist.

Die Folge der Länge n mit n$\geq$2 von Früchten ist in einem Feld A mit den Indizes von 1 bis n abgespeichert (die Folge ist daher $A[1],\ldots,A[n]$). Für jeden Eintrag $A[i]$ gibt es zwei Attribute $A[i].Art \in \{\text{Apfel}, \text{Birne}\}$ und $A[i].Gewicht \in \mathbb{N}$ für das Gewicht (in Gramm). In der Folge der Früchte gibt es mindestens einen Apfel und mindestens eine Birne.

a) Geben Sie einen Algorithmus in Pseudo-Code an, der in Laufzeit $O(n)$ zwei Indizes i und j als Ausgabe liefert, so dass $\{A[i], A[j]\}$ ein Paar aus einem Apfel und einer Birne ist und der Gewichtsunterschied von $A[i]$ und $A[j]$ minimal ist. Erläutern Sie die Funktionsweise und begründen Sie die Laufzeit und die Korrektheit Ihres Algorithmus.

b) Nehmen Sie an, dass alle Äpfel leichter sind als alle Birnen, d. h. in A kommen zunächst alle Äpfel, dann alle Birnen. Geben Sie unter dieser Annahme einen Algorithmus für das Problem aus Teilaufgabe a) an, der Laufzeit $o(n)$ hat (also insbesondere eine echt bessere Laufzeit als $O(n)$ hat). Schätzen Sie die Laufzeit Ihres Algorithmus ab und erläutern Sie Ihr Vorgehen.

Fortsetzung nächste Seite!

Teilaufgabe II: Theoretische Informatik

Aufgabe 1 (Kurzfragen) [25 PUNKTE]

Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie jeweils Ihre Entscheidung kurz. Falls eine Aussage sich durch ein konkretes Beispiel (bzw. Gegenbeispiel) beweisen (bzw. widerlegen) lässt, geben Sie eines an. Nur auf die richtige Angabe von „wahr" oder „falsch" gibt es keine Punkte.

a) Sei n eine natürliche Zahl und X eine Sprache, die alle Wörter der Länge $\geq n$ enthält. Dann ist X regulär.

b) Sei M ein nichtdeterministischer endlicher Automat (NFA) über dem Alphabet Σ, der die Sprache X akzeptiert. Wenn man aus allen Endzuständen Nichtendzustände macht und umgekehrt, erhält man einen NFA für das Komplement von X, also für die Sprache $\Sigma^* \setminus X$.

c) Sei X eine Sprache, die von einem *nichtdeterministischen* Kellerautomaten mit Endzuständen erkannt wird. Dann ist X kontextfrei, aber nicht deterministisch kontextfrei.

d) Beschreiben oder skizzieren Sie alle minimalen deterministischen endlichen Automaten (DFA), bei denen alle Zustände Endzustände sind. Begründen Sie, weshalb Ihre Lösung alle diese DFAs erfasst.

e) Nennen Sie eine konkrete reguläre Sprache, die mindestens zwei unendlich große Myhill–Nerode-Klassen besitzt. Geben Sie diese zwei Klassen explizit an und begründen Sie, warum diese wirklich Myhill–Nerode-Klassen der Sprache sind.

Fortsetzung nächste Seite!

Aufgabe 2 (Kontextfreie Sprachen) [40 PUNKTE]

1. Wir betrachten die Sprache $L = \{a^k bc^l \mid k, l \in \mathbb{N}_0; k \geq l \geq 0\}$ über dem Alphabet $\Sigma = \{a, b, c\}$.

   a) Geben Sie eine kontextfreie Grammatik für L an. Sie müssen nicht beweisen oder begründen, dass die Grammatik L erzeugt. Verwenden Sie höchstens drei Nichtterminale. Produktionen der Form $A \to \varepsilon$ sind erlaubt.

   b) Zeichnen Sie einen deterministischen Kellerautomaten, der L mit Endzustand erkennt. Verwenden Sie nicht mehr als vier Zustände.

   Hinweis: Geben Sie Übergänge in der Form $x, X/w$ an, wobei $x \in \{a, b, c\}$ das gelesene Zeichen der Eingabe ist, X das oberste Kellersymbol und w die Sequenz an Symbolen, durch die X ersetzt wird, wobei das oberste Symbol links steht. Geben Sie auch an, welches Symbol initial auf dem Keller steht.

   c) Zeigen Sie, dass L nicht regulär ist.

2. Wir betrachten die kontextfreie Grammatik $G = (\{S\}, \{a, b\}, P, S)$ mit den Produktionen P= $\{S \to baa \mid aSba \mid SS\}$.

   Zeigen Sie: Jedes Wort $w \in L(G)$ erfüllt $|w|_a = 2|w|_b$, d.h. es enthält doppelt so viele Buchstaben a wie b.

   Verwenden Sie hierfür eine geeignete Induktion über die Ableitung $S \to^* w$. Geben Sie in jedem Induktionsfall klar an, welches die Induktionshypothesen sind (falls vorhanden), was zu zeigen ist und an welchen Stellen Sie die Induktionshypothesen verwenden.

Aufgabe 3 (DFA-Konstruktion) [25 PUNKTE]

Gegeben sind eine Sprache $X \subseteq \Sigma^*$ und ein Zeichen $c \in \Sigma$. Es wird die Sprache

$$X_c := \{ucv \mid uv \in X\}$$

definiert, deren Wörter dadurch entstehen, dass in ein Wort aus X an genau einer Stelle c einfügt wird.

Beispiel: $\{aa, abd\}_c = \{caa, aca, aac, cabd, acbd, abcd, abdc\}$

Zeigen Sie, dass X_c regulär ist, falls X regulär ist. Geben Sie hierfür eine Konstruktion an, die aus einem NFA für X einen NFA für X_c macht. Beweisen Sie die Korrektheit Ihrer Konstruktion formal.

Erinnerung: Ein NFA akzeptiert ein Wort $w = w_1 \dots w_n$ gdw. es einen Lauf $q_0 \xrightarrow{w_1} \dots \xrightarrow{w_n} q_n$ gibt, der im Startzustand q_0 beginnt und in einem Endzustand q_n endet.

Fortsetzung nächste Seite!

Aufgabe 4 (Entscheidbarkeit und NP) [30 PUNKTE]

1. Zeigen Sie: Wenn $X \in \text{NP}$ ist, dann ist auch $X^* \in \text{NP}$.

2. Entscheiden Sie für jede der folgenden Sprachen, ob sie entscheidbar ist. Falls eine Sprache nicht entscheidbar ist, geben sie zusätzlich an, ob sie oder ihr Komplement semi-entscheidbar sind.

   Begründen Sie ihre Antworten jeweils knapp. Formale Beweise sind nicht gefordert, aber nicht Offensichtliches muss kurz begründet werden.

   a) $L_1 = \{w \mid M_w[\varepsilon] \text{ hält in } \leq |w| \text{ Schritten}\}$

   b) $L_2 = \{w \mid M_w \text{ hält für höchstens 10 Eingaben}\}$

   c) $L_3 = \{w \mid \varphi_w \text{ ist WHILE-berechenbar}\}$

   Hinweis zur Notation:

   - ε bezeichnet das leere Wort.
   - „Semi-entscheidbar“ ist synonym zu „rekursiv aufzählbar“.
   - M_w bezeichnet die Turing-Maschine, die durch das Wort w kodiert wird.
   - $M_w[x]$ bezeichnet die Ausführung von M_w auf der Eingabe x.
   - φ_w bezeichnet die von M_w berechnete Funktion, d.h. $\varphi_w(x)$ ist das Ergebnis von $M_w[x]$ falls die Ausführung hält, und $\perp$ sonst.