Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 9

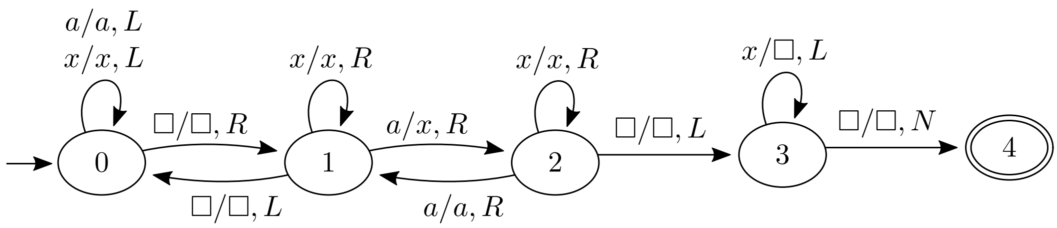
$w'\#w_\perp \in \mathcal{H}_{NEQ}$

- Die Aufgaben werden in folgender Reihenfolge korrigiert: H9.1, H9.2.
- Die Knobelaufgabe bitte separat auf Moodle abgeben. Sie wird korrigiert.

Aufgabe H9.1. (*Jeu de mots*) 4+4+4 Punkte

Ihr Freund, der Archäologe Jasper Vazarie, hat einige Programme ausgegraben, und braucht Ihre Hilfe, um deren Bedeutung zu verstehen. Begründen Sie jeweils Ihre Antwort und beschreiben Sie, wieso die Programme funktionieren.

(a) Folgende Maschine wurde als Grabbeigabe (neben zwei Ringen) in Uruk gefunden. Welche Sprache $L \subseteq \{a\}^*$ akzeptiert sie?



(b) Überlieferungen zufolge hatte der Hochpriester von Nakbé folgendes Programm verwendet, um seine Entscheidungen zu begründen. Welche Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet es?
Hinweis: Zu Beginn steht der Input in x_1, und alle anderen Register sind 0. Der Output steht am Ende in x_0. (siehe Definition 5.20 auf Folie 252)

$x_0 := z + 1$
$i := z + 1$
$t := x_1 - 1$
while $t \neq 0$ **do**
  $i := i + 2$
  $t := i + 0$
  while $t \neq 0$ **do**
    $x_0 := x_0 + 1$
    $t := t - 1$
  $t := x_1 + 0$
  $k := x_0 + 0$
  while $k \neq 0$ **do**
    $k := k - 1$
    $t := t - 1$

(c) Folgende Wegbeschreibung wurde neben einer römischen Straße im heutigen Italien entdeckt. Welche Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet sie?

Geata:	**if** $x_1 \leq 1$ **goto** Roma
	$p := 1$
Labici:	$p := p + 1$
	$t := x_1 \,\texttt{MOD}\, p$
	if $t \neq 0$ **goto** Labici
Ardea:	$x_1 := x_1 \,\texttt{DIV}\, p$
	$t := x_1 \,\texttt{MOD}\, p$
	if $t = 0$ **goto** Ardea
	$x_0 := x_0 + 1$
	$x_1 := x_1 * x_1$
	goto Geata
Roma:	`HALT`

Hinweise: Zu Beginn steht der Input in x_1, und alle anderen Register sind 0. Der Output steht am Ende in x_0. Beachten Sie, dass hier die syntaktischen Abkürzungen von Folie 245 verwendet wurden. Insbesondere ist $x \,\texttt{DIV}\, y$ die (abgerundete) Division von x durch y, und $x \,\texttt{MOD}\, y$ der verbleibende Rest. Außerdem verwenden wir $x \leq k$ und $x \neq k$ für ein Register x und $k \in \mathbb{N}$, mit der offensichtlichen Bedeutung.

Lösungsskizze.

(a) $\{a^{2^n} : n \in \mathbb{N}\}$.

Die TM geht das Wort von links nach rechts durch, und ersetzt jedes zweite a durch ein x. Falls es eine ungerade Anzahl an a gibt, prüft sie, dass es genau ein a ist (indem sie das erste a durch ein x ersetzt und schaut, dass danach nur x übrigbleiben), und akzeptiert. Ansonsten wiederholt sich der Prozess. Damit wird die Anzahl an a in jedem Schritt halbiert. Falls es am Anfang eine Zweierpotenz war, bleibt irgendwann ein einzelnes a übrig und die Eingabe wird akzeptiert, ansonsten bleibt irgendwann eine größere ungerade Zahl an a übrig, und die TM akzeptiert nicht. Auf Eingabe ε hängt die TM, diese wird also (korrekterweise) nicht akzeptiert.

(b) $f(n) = \min\{i^2 : i \in \mathbb{N}_{>0}, i^2 \geq n\}$. Wir vereinfachen das Programm zunächst, indem wir `WHILE`-Konstrukte in übliche mathematische Notation überführen.

$x_0 := 1$
$i := 1$
$t := x_1 - x_0$
while $t \neq 0$ **do**
  $i := i + 2$
  $x_0 := x_0 + i$
  $t := x_1 - x_0$

Nun ist auch ersichtlich, dass wir die Bedingung $t \neq 0$ zu $x_0 < x_1$ umformen können (man beachte, dass Subtraktion nicht negativ werden kann). Register i iteriert über die ungeraden Zahlen und wird in x_0 aufsummiert, und die Summe der ungeraden Zahlen sind die Quadratzahlen.

(c) $f(n) = |\{p \in \mathbb{N} : p \text{ prim } \wedge \frac{n}{p} \in \mathbb{N}\}|$, also die Anzahl der Primzahlen, die n teilt. Z.B. ist $45 = 3^2 \cdot 5$ und somit $f(45) = |\{3, 5\}| = 2$.

Insbesondere ermittelt Labici die kleinste Zahl $p \in \mathbb{N}_{\geq 2}$ die x_1 teilt. Dies muss eine Primzahl sein. In Ardea wird dann x_1 so oft wie möglich durch p geteilt. Das Ergebnis ist also nicht mehr durch p teilbar. Anschließend wird x_1 quadriert; dies ist jedoch irrelevant, da x_1^2 und x_1 durch genau die gleichen Primzahlen teilbar sind.

Aufgabe H9.2. (*Unbegrenzte Macht*) 2+2+3 Punkte

Eines Nachts kam ein Bewohner eines anderen Universums zu Dr. Evilsparza und gab ihm eine Maschine, die die Grenzen unseres Universums transzendiert. Sie konnte nicht nur das Halteproblem lösen, sondern *jedes* Problem! Dr. Evilsparza war begeistert, doch als er diese Maschine benutzen wollte, um die Welt zu erobern, wachte er auf.

Wir definieren nun verschiedene Versionen von *unendlichen Turing-Maschinen* (∞TMs). Sei $L \subseteq \{a\}^*$ eine beliebige Sprache. Geben Sie für jede der Versionen eine ∞TM $M = \{Q, \{a\}, \Gamma, \delta, q_0, \Box, F\}$ mit $L(M) = L$ an und beschreiben Sie kurz Ihre Idee. (Sie dürfen die ∞TM mithlife einer Skizze angeben.)

(a) Eine ∞TM ist genauso definiert wie eine TM, mit dem Unterschied, dass die Menge Q der Zustände unendlich sein darf.

(b) Eine ∞TM ist eine ∞-Band-TM, also eine TM mit unendlich vielen Bändern. Die Übergangsfunktion ist also eine Funktion $\delta : Q \times \Gamma^\omega \to Q \times \Gamma^\omega \times \{L, R, N\}^\omega$, die von allen Bändern gleichzeitig liest, auf alle Bänder gleichzeitig schreibt, und sich auf allen Bändern gleichzeitig bewegt. Am Anfang steht die Eingabe auf dem ersten Band und alle anderen Bänder sind leer.

Erinnerung: Sei A ein Alphabet. Für ein Wort $w \in A^+$ ist w^ω das unendliche Wort $www\cdots$. A^ω ist die Menge aller unendlichen Wörter über A, d.h. die Menge $\{w_1w_2w_3\cdots \mid \forall i \in \mathbb{N}_+ : w_i \in A\}$.

(c) Eine ∞TM ist genauso definiert wie eine TM, mit dem Unterschied, dass das Bandalphabet Γ unendlich sein darf.

Lösungsskizze.

(a) Idee: Man kann die Zustände von M in einer unendlichen Linie anordnen und dann von den „richtigen“ zu einem Endzustand gehen. Genauer: Man definiert

$$
\begin{aligned}
Q &= \{q_i : i \in \mathbb{N}\} \cup \{q_f\} & \\
\delta(q_i, a) &= (q_{i+1}, a, R) & \text{für } i \in \mathbb{N} \\
\delta(q_i, \Box) &= (q_f, \Box, N) & \text{für } i \in \mathbb{N},\ a^i \in L
\end{aligned}
$$

Für $i \in \mathbb{N}$ mit $a^i \notin L$ ist $\delta(q_i, \Box)$ undefiniert. Die Menge der Endzustände ist $F \coloneqq \{q_f\}$.

(b) Idee: Man schreibt die Eingabe „senkrecht“ hin, d.h. jedes Zeichen auf ein eigenes Band. (Das ist nicht unbedingt nötig, es ist nur eine mögliche Lösung.) Genauer: $Q = \{q_0, q_f\}$, $F = \{q_f\}$, $\Gamma = \{\Box, a\}$ und

$$
\delta(q_0, a^i\Box^\omega) = (q_0, a^{i+1}\Box^\omega, RN^\omega) \text{ für alle } i \in \mathbb{N}_+,
$$

$$
\delta(q_0, \Box a^i\Box^\omega) = \begin{cases} (q_f, \Box a^i\Box^\omega, N^\omega) & \text{falls } a^i \in L \\ (q_0, \Box a^i\Box^\omega, N^\omega) & \text{falls } a^i \notin L \end{cases} \text{ für alle } i \in \mathbb{N}.
$$

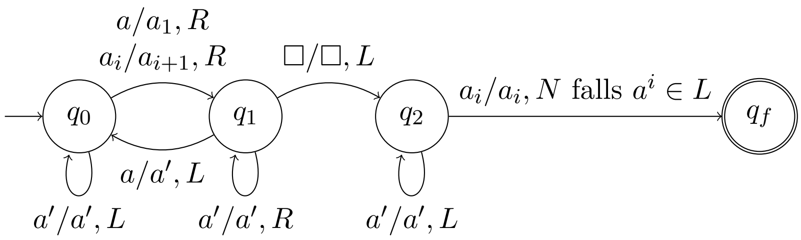
(c) Idee: Für jede Eingabe a^i wird ein eindeutiges Zeichen auf das Band geschrieben. Abhängig von diesem Zeichen wird das Wort dann akzeptiert/nicht akzeptiert. Hier wird eine mögliche Lösung beschrieben.

Definiere $Q = \{q_0, q_1, q_2, q_f\}$, $F = \{q_f\}$, $\Gamma = \{\Box, a, a'\} \cup \{a_i \mid i \in \mathbb{N}_+\}$ und

$$\begin{aligned}
\delta(q_0, a) &= (q_1, a_1, R),\ \delta(q_1, a) = (q_0, a', L),\\
\delta(q_0, a') &= (q_0, a', L),\ \delta(q_0, a_i) = (q_1, a_{i+1}, R) \text{ für alle } i \in \mathbb{N}_+,\\
\delta(q_1, a') &= (q_1, a', R),\ \delta(q_1, \Box) = (q_2, \Box, L),\\
\delta(q_2, a') &= (q_2, a', L),\ \delta(q_2, a_i) = \begin{cases} q_f \text{ falls } a^i \in L \\ \bot \text{ falls } a^i \notin L. \end{cases}
\end{aligned}$$

Die ∞TM M liest das Wort von links nach rechts, bis sie ein a findet oder das Wort zu Ende ist. Falls ein a gefunden wird, wird dieses zu a' (d.h. als „gezählt“ markiert), der „Zähler“ wird um 1 inkrementiert und das Wort wird nochmals gelesen; falls das Wort zu Ende ist, geht M im Zustand q_2 zurück zum „Zähler“ und entscheidet dann, ob die Eingabe akzeptiert wird.

Falls $\varepsilon \in L$, fügen wir noch den Übergang $\delta(q_0, \Box) = (q_f, \Box, N)$ hinzu.



Quizaufgabe H9.3. (*Gelangwhiled?*) unkorrigiert (8 Punkte)

Doras Kindergartenprofessor steht leider im Stau, und die Kindergartenkinder wissen nicht, was sie mit ihrer Zeit anfangen sollen. Ehe es einen Aufstand gibt, beschließen Sie die Kinder zu unterhalten, indem Sie die Konvertierung von `WHILE`-Programmen zu deterministischen Turingmaschinen erklären.

(a) Sei $\Sigma := \{0, 1\}$. Geben Sie eine TM M^0 an, die die Funktion $f : \Sigma^* \to \Sigma^*$ berechnet, mit $f(w) = 0$ für alle w.

(b) Geben Sie eine TM M^- an, die eine Funktion $f : \Sigma^* \to \Sigma^*$ berechnet, wobei $f(\mathit{bin}(i)) := \mathit{bin}(i - 1)$ für alle $i \in \mathbb{N}_{>0}$ gilt, und $f(0) := 0$.

(Wie aus der Vorlesung bekannt, bezeichnet $\mathit{bin}(i)$ die Binärdarstellung von i, also das kürzeste $w \in \{0, 1\}^+$ mit $(w)_2 = i$.)

Im Folgenden seien die TMs aus (a) und (b) so modifiziert, dass sie auf einem beliebigen (aber festen) Band einer k-Band-TM operieren. Wir nennen die Version von M^-, die auf Band i operiert, $\mathsf{dec}(i)$. Ebenso sei $\mathsf{setzero}(i)$ die Version von M^0, die auf Band i arbeitet. Zusätzlich verwenden wir $\mathsf{inc}(i)$ für die angepasste TM aus Beispiel 5.13 (Binär +1), und $\mathsf{iszero}(i)$ für die „Band $i = 0$?“ TM von Folie 247.

(c) Seien $n, i \in \mathbb{N}$ fix. Konstruieren Sie nun eine k-Band TM $\mathsf{add}(i, n)$, die die Binärzahl auf Band i um n erhöht, und eine k-Band TM $\mathsf{sub}(i, n)$, die die Binärzahl auf Band i um n senkt (statt negativen Ergebnissen ergibt sich 0).

Hinweis: Hier (und im Folgenden) bietet sich die Notation zur sequentiellen Komposition aus der Vorlesung an (Folie 246).

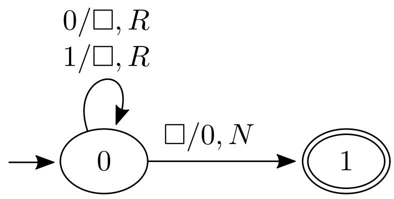
(d) Konstruieren Sie eine k-Band TM $\mathsf{copy}(i, j)$, die das `WHILE`-Programm $x_i := x_j$ simuliert.

(e) Konvertieren Sie nun folgendes `WHILE`-Programm, das eine Funktion $f : \mathbb{N} \to \mathbb{N}$ berechnet, zu einer k-Band TM:

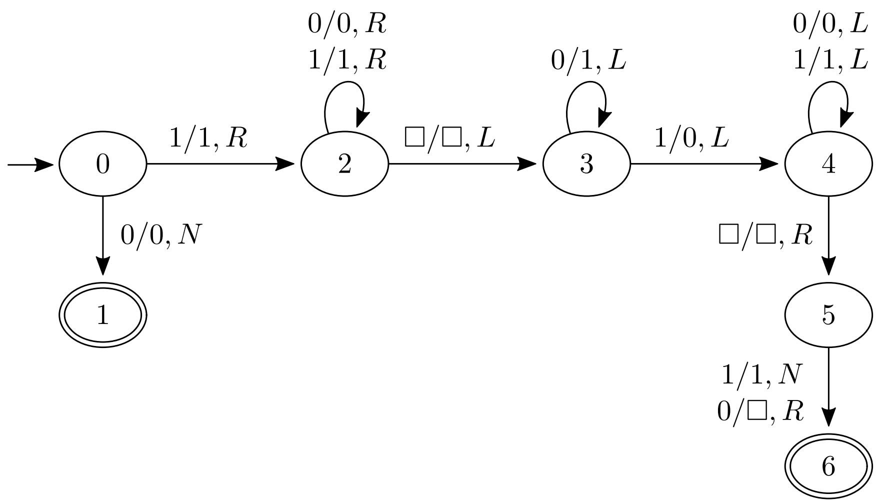
while $x_1 \neq 0$ **do**
  $x_2 := 0$
  while $x_1 \neq 0$ **do**
    $x_1 := x_1 - 2$
    $x_2 := x_2 + 1$
  $x_1 := x_2$
  $x_0 := x_0 + 1$

Lösungsskizze.

(a)



(b)



(c) Wir schalten einfach $\mathsf{inc}(i)$ (bzw. $\mathsf{dec}(i)$) n-mal hintereinander:

$$\mathsf{add}(i, n)\colon \quad \to \mathsf{inc}(i) \to \mathsf{inc}(i) \to \cdots \to \mathsf{inc}(i) \to \quad (n\text{-mal})$$

$$\mathsf{sub}(i, n)\colon \quad \to \mathsf{dec}(i) \to \mathsf{dec}(i) \to \cdots \to \mathsf{dec}(i) \to \quad (n\text{-mal})$$

Alternativ kann man die TMs auch rekursiv definieren, z.B. $\mathsf{add}(i, 1) := \mathsf{inc}(i)$ und

$$\mathsf{add}(i, n)\colon \quad \to \mathsf{add}(i, n-1) \to \mathsf{inc}(i) \to$$

(d) Wir wählen uns eine Hilfsvariable x_k. Zuerst implementieren wir eine TM $\mathsf{move}(k, i, j)$, die das Programm $x_k := x_j;\ x_i := x_i + x_j;\ x_j := 0$ implementiert:

$$\begin{tikzcd}
{} \arrow[r] & \mathsf{setzero}(k) \arrow[r] & \mathsf{iszero}(j) \arrow[r, "\text{nein}"] \arrow[d, "\text{ja}"] & \mathsf{dec}(j) \arrow[r] & \mathsf{inc}(k) \arrow[r] & \mathsf{inc}(i) \arrow[lll, bend right] \\
& & {} & & &
\end{tikzcd}$$

Die TM für $x_i := x_j$ erhalten wir nun als

$$\longrightarrow \mathsf{move}(k, i, j) \longrightarrow \mathsf{move}(i, j, k) \longrightarrow$$

(e)

$$\begin{tikzcd}
{} \arrow[r] & \mathsf{iszero}(1) \arrow[r, "\text{nein}"] \arrow[d, "\text{ja}"'] & \mathsf{setzero}(2) \arrow[r] & \mathsf{iszero}(1) \arrow[r, "\text{nein}"] \arrow[d, "\text{ja}"] & \mathsf{sub}(1,2) \arrow[r] & \mathsf{inc}(2) \arrow[ll, bend right] \\
& {} & & \mathsf{copy}(1,2) \arrow[r] & \mathsf{inc}(0) \arrow[uuulll, bend left] &
\end{tikzcd}$$

Knobelaufgabe H9.4. (*Permutation*)

Sei $\Sigma := \{a, b\}$. Für ein Wort $w \in \Sigma^*$ mit $w = w_1 w_2 ... w_n$ definieren wir nun die Menge der *Permutationen* von w als

$$\mathrm{Perm}(w) := \{w_{f(1)} w_{f(2)} ... w_{f(n)} \mid f : \{1, ..., n\} \to \{1, ..., n\} \text{ bijektiv}\}$$

Beispielweise gilt $\mathrm{Perm}(abaa) = \{aaab, aaba, abaa, baaa\}$. Wir erweitern dies auf Sprachen, für $L \subseteq \Sigma^*$ gilt also $\mathrm{Perm}(L) := \bigcup_{w \in L} \mathrm{Perm}(w)$.

Zeigen Sie, dass $\mathrm{Perm}(L)$ kontextfrei ist, wenn L regulär ist.

Lösungsskizze. Man kann aus einem DFA M für L einen PDA M' konstruieren. M' hat als Zustände genau die Zustände von M, und verwendet den Stapel als Zähler. Es gibt folgende Transitionen:

- M' kann ein a einlesen, und den gleichen Übergang wie M ausführen.
- M' kann ein b einlesen, und den Zähler um 1 inkrementieren.
- M' kann den b-Übergang von M ausführen, und den Zähler um 1 senken.
- Wenn M' in einem Finalzustand von M ist und der Zähler auf 0 steht, kann M' akzeptieren.

Wir beschreiben dies nun formal und beweisen die Korrektheit der Konstruktion. Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA für L.

Wir definieren einen PDA $M' = (Q, \Sigma, \{Z_0, \mathtt{+}, \mathtt{-}\}, q_0, Z_0, \delta', F)$, sodass $L_\varepsilon(M') = L(M)$. Die Transitionen sind:

$$\begin{aligned}
\delta'(q, a, X) &:= (\delta(q, a), X) && \text{für } q \in Q, X \in \Gamma && (1)\\
\delta'(q, b, \mathtt{+}) &:= (q, \varepsilon) && \text{für } q \in Q && (2)\\
\delta'(q, b, X) &:= (q, \mathtt{-}X) && \text{für } q \in Q, X \in \{\mathtt{-}, Z_0\} && (3)\\
\delta'(q, \varepsilon, X) &:= (\delta(q, b), \mathtt{+}X) && \text{für } q \in Q, X \in \{\mathtt{+}, Z_0\} && (4)\\
\delta'(q, \varepsilon, \mathtt{-}) &:= (\delta(q, b), \varepsilon) && \text{für } q \in Q && (5)\\
\delta'(q, \varepsilon, Z_0) &:= (q, \varepsilon) && \text{für } q \in F && (6)
\end{aligned}$$

Nun zeigen wir die Korrektheit von M'. Dazu definieren wir $f : \mathbb{Z} \to \Gamma^*$ wie folgt.

$$f(n) := \begin{cases} \texttt{+}^n Z_0 & \text{falls } n \geq 0 \\ \texttt{-}^{-n} Z_0 & \text{sonst} \end{cases}$$

Der Stack $f(n)$ repräsentiert also einen Wert n. Man überzeugt sich leicht per Induktion, dass der Stackinhalt von M' stets ein Wort in $f(\mathbb{Z})$ ist. Wir definieren außerdem $g : \Sigma^* \to \mathbb{N}^2$, $g(w) := \binom{|w|_a}{|w|_b}$.

Wir behaupten

$$(q_0, w', Z_0) \to^*_{M'} (q, \varepsilon, f(m)) \Leftrightarrow \exists w : g(w) = g(w') - \binom{0}{m} \wedge \delta(q_0, w) = q$$

für alle $w' \in \Sigma^*$, $q \in Q$ und $m \in \mathbb{Z}$.

„$\Rightarrow$“: Der Beweis erfolgt per Induktion über die Länge der Transitionssequenz auf der linken Seite.

Für die Basis gilt $w' = \varepsilon$, $q = q_0$ und $m = 0$, und die rechte Seite folgt über $w := \varepsilon$.

Für den Schritt sei $(q_0, w', Z_0) \to^{n+1}_{M'} (q, \varepsilon, f(m))$. Dann gibt es $p \in Q$, $x \in \Sigma \cup \{\varepsilon\}$, $d \in \{-1, 0, 1\}$ mit $w' = ux$ und

$$(q_0, ux, Z_0) \to^n_{M'} (p, x, f(m-d)) \to_{M'} (q, \varepsilon, f(m))$$

Nach IA gibt es außerdem ein w mit $g(w) = g(u) - \binom{0}{m-d} \wedge \delta(q_0, w) = p$. Wir betrachten nun alle Möglichkeiten für die letzte Transition von M':

(1) $x = a$, $d = 0$, $q = \delta(p, a)$, und somit $g(wa) = g(w) + g(a) = g(u) + g(a) - \binom{0}{m} = g(w') - \binom{0}{m}$ und $\delta(q_0, wa) = \delta(q, a) = p$.

(2,3) $x = b$, $d = 1$, $q = p$, und es gilt $g(w) = g(u) - \binom{0}{m-1} = g(ub) - \binom{0}{m} = g(w') - \binom{0}{m}$.

(4,5) $x = \varepsilon$, $d = -1$, $q = \delta(p, b)$, also folgt $g(wb) = g(w) + g(b) = g(u) + g(b) - \binom{0}{m+1} = g(u) + \binom{0}{m} = g(w') + \binom{0}{m}$.

(6) Dies widerspricht dem resultierenden Stackinhalt $f(m) \neq \varepsilon$.

„$\Leftarrow$“: Zunächst beobachten wir $m \leq |w'|_b \leq |w'|$. Wir zeigen die Aussage per Induktion über $|w'|$. Die Basis folgt unmittelbar: $w' = \varepsilon$, $m = 0$. Es folgt $w = \varepsilon$, $q = q_0$ und $(q_0, \varepsilon, Z_0) \to^*_{M'} (q_0, \varepsilon, Z_0)$ gilt.

Für den Schritt wählen wir nun w entsprechend der Aussage und unterscheiden folgende Fälle:

- $w = ub$. Dann gilt $g(u) = g(w') - \binom{0}{m+1}$ und nach IA folgt $(q_0, w', Z_0) \to^*_{M'} (\delta(q_0, u), \varepsilon, f(m+1))$. Durch Ausführen von (4,5) erreichen wir $(q, \varepsilon, f(m))$.
- $w = ua$. Wir zerlegen $w''ab^k := w$, mit $w'' \in \Sigma^*, k \in \mathbb{N}$. Nach IA folgt $(q_0, w', Z_0) \to^*_{M'} (\delta(q_0, u), ab^k, f(m-k))$. Dur Ausführen von (1) und k-mal (2,3) erreichen wir $(q, \varepsilon, f(m))$.

Damit ist die Behauptung bewiesen. Es gilt:

$$
\begin{aligned}
&w' \in L_\varepsilon(M') \\
&\Leftrightarrow \exists q \in Q : (q_0, w', Z_0) \rightarrow^*_{M'} (q, \varepsilon, \varepsilon) \\
&\Leftrightarrow \exists q \in F : (q_0, w', Z_0) \rightarrow^*_{M'} (q, \varepsilon, Z_0) \\
&\Leftrightarrow \exists q \in F, w \in \Sigma^* : g(w) = g(w') \wedge \delta(q_0, w) = q \\
&\Leftrightarrow \exists w \in \Sigma^* : g(w) = g(w') \wedge w \in L \\
&\Leftrightarrow w' \in \mathrm{Perm}(L)
\end{aligned}
$$