

Einführung in die theoretische Informatik
Sommersemester 2019 – Übungsblatt Lösungsskizze 8

AUFGABE 8.1. (*Wichtige Begriffe*)

Stufe A

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

- intuitiv berechenbar
- totale/partielle/echt partielle Funktion
- nicht-deterministische/deterministische Turing-Maschine
- abzählbar
- überabzählbar
- Konfiguration einer Turing-Maschine
- akzeptierte Sprache einer Turing-Maschine
- Turing-berechenbar

AUFGABE 8.2. (*Pumping Lemma für kontextfreie Sprachen üben*)

Stufe C

Zeigen Sie durch einen Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen, dass folgende Sprachen nicht kontextfrei sind:

- (a) $L_1 = \{a^{2^i} \mid i \geq 0\}$
(b) $L_2 = \{a^i b^j a^k \mid j = \max\{i, k\}\}$

Lösungsskizze

- (a) $L_1 = \{a^{2^i} \mid i \geq 0\}$

- Wir nehmen an, dass L_1 kontextfrei ist und führen diese Annahme zum Widerspruch.
- Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L_1 .
- Sei $z = a^{2^n}$, d.h., $z \in L_1$ und $|z| \geq n$.
- Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1) vx \neq \varepsilon \quad (2) |vwx| \leq n \quad (3) \forall i \in \mathbb{N}_0. uv^i wx^i y \in L_1$$

- Wegen (2) gilt $|uv^2wx^2y|_a \leq 2^n + n < 2^n + 2^n = 2^{n+1}$.
- Wegen (1) gilt $|uv^2wx^2y|_a > 2^n$.
- Folglich ist $uv^2wx^2y = a^k$ mit $2^n < k < 2^{n+1}$, also $a^k \notin L_1$. Dies ist ein Widerspruch zu (3).
- Die ursprüngliche Annahme kann also nicht gelten. Folglich ist L_1 nicht kontextfrei.

- (b) $L_2 = \{a^i b^j a^k \mid j = \max\{i, k\}\}$

- Wir nehmen an, dass L_2 kontextfrei ist und führen diese Annahme zum Widerspruch.
- Sei $n \geq 1$ eine Pumping-Lemma-Zahl für die kontextfreie Sprache L_2 .
- Sei $z = a^n b^n a^n$, d.h., $z \in L_2$ und $|z| \geq n$.
- Gemäß Pumping-Lemma gibt es dann also eine Zerlegung $z = uvwxy$ mit Wörtern $u, v, w, x, y \in \Sigma^*$ und den folgenden Eigenschaften:

$$(1) vx \neq \varepsilon \quad (2) |vwx| \leq n \quad (3) \forall i \in \mathbb{N}_0. uv^i wx^i y \in L_2$$

- Wir unterscheiden folgende Fälle:
 - $|vx|_a = 0$: Wegen (1) gilt $|vx|_b = k > 0$. Dann ist aber $uv^0wx^0y = a^n b^{n-k} a^n$ und da $n - k \neq \max\{n, n\}$ für $k > 0$ ist $uv^0wx^0y \notin L_2$. Dies ist ein Widerspruch zu (3).
 - $|vx|_a > 0$: Wir machen eine weitere Fallunterscheidung nach $|vx|_b$:
 - * $|vx|_b = 0$: Da $|vx|_a = k > 0$ und da kein b in vx auftritt, ist $uv^2wx^2y \in \{a^{n+k} b^n a^n, a^n b^n a^{n+k}\}$. Für beide Worte gilt, dass $|uv^2wx^2y|_b = n \neq n + k$, und folglich $uv^2wx^2y \notin L_2$. Dies ist ein Widerspruch zu (3).
 - * $|vx|_b > 0$: Wegen (2) kann vx nicht beide Blöcke von as berühren. Folglich ist $uv^0wx^0y \in \{a^{n-k} b^{n-l} a^n, a^n b^{n-l} a^{n-k}\}$, wobei $k = |vx|_a$ und $l = |vx|_b$. Da sowohl $k > 0$ als auch $l > 0$, gilt $n - k < n$ und $n - l < n$. Folglich ist die Anzahl an bs nicht gleich der Anzahl an as im größeren Block, und deshalb ist $uv^0wx^0y \notin L_2$. Dies ist ein Widerspruch zu (3).
- Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L_2 nicht kontextfrei.

Definition (Alternative Akzeptanzbedingungen für Turing-Maschinen)

In der Vorlesung wurde die Annahme gemacht, dass die Übergangsfunktion δ einer Turingmaschine folgende Eigenschaft erfüllt:

$$\delta(q, a) \text{ ist nicht definiert für alle } q \in F, a \in \Gamma.$$

Sei \mathcal{M}_A die Menge der Turingmaschinen, die diese Annahme erfüllen, und sei \mathcal{M} die Menge aller Turingmaschinen. Es gilt somit $\mathcal{M}_A \subsetneq \mathcal{M}$.

Für $M \in \mathcal{M}$ mit $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ definiere:

- $L_F(M) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in F. (\varepsilon, q_0, w) \xrightarrow{*}_M (\alpha, f, \beta)\}$.
(Menge der Wörter, für die die Maschine einen Endzustand irgendwann besucht.)
- $L_H(M) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, q \in Q. (\varepsilon, q_0, w) \xrightarrow{*}_M (\alpha, q, \beta) \text{ und } \delta(q, \text{first}(\beta)) \text{ ist nicht definiert}\}$.
(Menge der Wörter, für die die Maschine hält.)
- $L_{HF}(M) = \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^*, f \in F. (\varepsilon, q_0, w) \xrightarrow{*}_M (\alpha, f, \beta) \text{ und } \delta(f, \text{first}(\beta)) \text{ ist nicht definiert}\}$.
(Menge der Wörter, für die die Maschine in einem Endzustand hält.)

AUFGABE 8.3. (TM für Sprache)

Stufe C

Geben Sie für die beiden angegebenen Sprachen die jeweils passende TM M_i an.

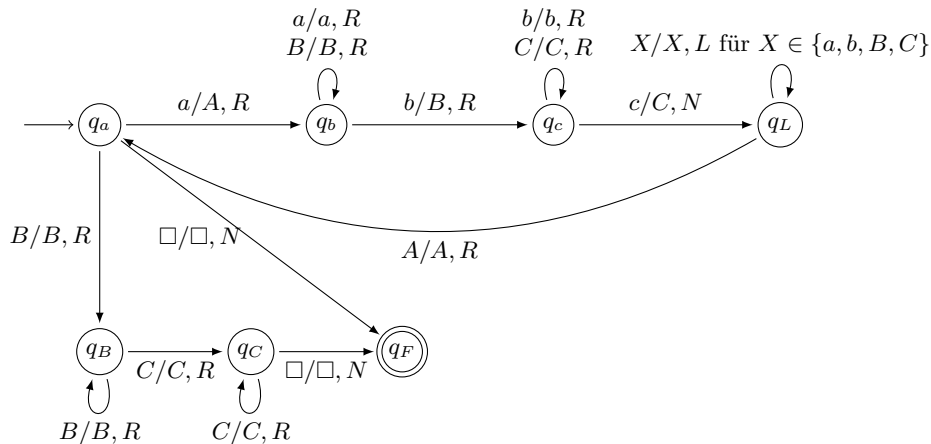
$$(a) L_F(M_1) = \{a^n b^n c^n \mid n \in \mathbb{N}_0\} \quad (b) L_F(M_2) = \{a^n b^{n^2} \mid n \in \mathbb{N}_0\}$$

Lösungsskizze

Idee: Ersetze für jedes a je ein a , b und c durch x , wobei nach dem Ersetzen die restlichen Buchstaben der gleichen Art unverändert gelassen werden. Wenn am Ende nur noch x auf dem Band stehen, terminiere. Sonst bleibt die Berechnung in einem Nichtendzustand stecken.

Wir schreiben \square für eine leere Bandzelle.

(a) Sei TM $M_1 = (\{q_a, q_b, q_c, q_L, q_B, q_C, q_F\}, \{a, b, c\}, \Sigma \cup \{A, B, C, \square\}, \delta, q_a, \square, \{q_F\})$.



(b) Idee: Benutze den a -Bereich auch als Zähler. Für jedes a müssen $|w|_a$ viele bs gelöscht werden.

Zwei Arten zu markieren:

- Ersetzen von a durch d zeigt an, dass für dieses a bereits $|w|_a$ gelöscht werden/wurden.
- Ersetzen von a bzw d durch a' bzw d' zeigt an, dass im momentanen Durchlauf für dieses a bzw d bereits ein b gelöscht wurde.

Ablauf: Für jedes a , führe folgendes durch:

- Ersetze es durch d (iii).
- Für jedes d und a auf dem Band (iv) und (viii):
 - Markiere es als d' bzw a' (v).
 - Lösche das rechte b (vi) und (vii).
- Entferne alle Markierungen a' bzw d' (ii)

Eine Wiederholung löscht $|w|_a$ bs ; da es für jedes a wiederholt wird, werden $|w|_a \cdot |w|_a$ bs gelöscht. Wenn also am Ende kein b mehr auf dem Band steht, können wir im Endzustand f terminieren.

(Alternativ: Schreibe b^{n^2} zu $b^{(n-1)^2}$ um, in dem man ein a und $2n - 1$ b löscht.)

(i) Terminiere sofort bei leerer Eingabe, sonst gehe zu (iii):

$$q_0 \xrightarrow{\square/\square, N} f \quad q_0 \xrightarrow{a/a, N} q$$

(ii) Laufe einmal bis ganz nach links und lösche alle Markierungen auf dem Weg dorthin

$$p \xrightarrow{\begin{smallmatrix} a'/a,L \\ d'/d,L \end{smallmatrix}} p \quad p \xrightarrow{\square/\square,R} q$$

(iii) Laufe zum ersten a von links und lösche es:

$$q \xrightarrow{d/d,R} q \quad q \xrightarrow{a/d,L} r$$

(iv) Laufe wieder zum linkesten Symbol:

$$r \xrightarrow{d/d,L} r \quad r \xrightarrow{\square/\square,R} s$$

(v) Markiere es:

$$s \xrightarrow{\begin{smallmatrix} d'/d',R \\ a'/a',R \end{smallmatrix}} s \quad s \xrightarrow{\begin{smallmatrix} d/d',R \\ a/a',R \end{smallmatrix}} t$$

(vi) Suche rechtestes b :

$$t \xrightarrow{\begin{smallmatrix} a/a,R \\ b/b,R \\ d/d,R \end{smallmatrix}} t \quad t \xrightarrow{\square/\square,L} u$$

(vii) Lösche es:

$$u \xrightarrow{b/\square,L} v$$

(viii) Suche rechtestes a' bzw. d' :

$$v \xrightarrow{\begin{smallmatrix} b/b,L \\ a/a,L \\ d/d,L \end{smallmatrix}} v \quad v \xrightarrow{\begin{smallmatrix} a'/a',R \\ d'/d',R \end{smallmatrix}} w$$

Falls noch unmarkiertes a oder d übrig, markiere es und wechsele in Zustand t (goto (vi)); falls noch ein b übrig ist, wechsele in Zustand p (goto (ii))

$$w \xrightarrow{\begin{smallmatrix} a/a',R \\ d/d',R \end{smallmatrix}} t \quad w \xrightarrow{b/b,L} p \quad w \xrightarrow{\square/\square,L} x$$

(ix) ansonsten akzeptiere, falls Eingabe nur noch aus markierten d' besteht:

$$x \xrightarrow{d'/d',N} f$$

AUFGABE 8.4. (TM Berechnung)

Konstruieren Sie eine Turingmaschine $T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, mit $\Sigma = \{\mid\}$, die eine eingegebene Strichzahl verdoppelt.

Stufe B

Lösungsskizze

Die Berechnung erfolgt, indem jeweils ein Strich am linken Anfang der Strichfolge durch eine Markierung x ersetzt wird und anschließend am rechten Ende der Strichfolge eine Markierung y angefügt wird. Falls keine Strichzeichen mehr vorhanden sind, werden alle Zeichen in Striche umgewandelt.

Sei $T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ mit $Q = \{q_0, q_1, q_2, q_3, q_4, q_f\}$, $\Sigma = \{\mid\}$, $\Gamma = \{\mid, x, y, \square\}$ und $F = \{q_f\}$.

Die Zustände und Zeichenmengen kann man auch der folgenden Tabelle der Übergangsfunktion δ entnehmen.

Übergang	Kommentar
$\delta(q_0, \square) \rightarrow (q_3, \square, L)$	Nichts mehr zu verdoppeln
$\delta(q_0, \mid) \rightarrow (q_1, x, R)$	\mid wird verdoppelt, x ist ein Hilfszeichen
$\delta(q_0, y) \rightarrow (q_0, y, R)$	Überspringe Hilfszeichen y
$\delta(q_1, \square) \rightarrow (q_2, y, N)$	Schreibe \mid an rechten Rand, markiert durch Hilfszeichen y
$\delta(q_1, \mid) \rightarrow (q_1, \mid, R)$	An den rechten Rand gehen
$\delta(q_1, y) \rightarrow (q_1, y, R)$	An den rechten Rand gehen
$\delta(q_2, \mid) \rightarrow (q_2, \mid, L)$	Zurück zum nächsten \mid
$\delta(q_2, y) \rightarrow (q_2, y, L)$	Zurück zum nächsten \mid
$\delta(q_2, x) \rightarrow (q_0, x, R)$	Nächstes \mid verdoppeln oder halten
$\delta(q_3, y) \rightarrow (q_3, \mid, L)$	Hilfszeichen durch \mid ersetzen
$\delta(q_3, x) \rightarrow (q_3, \mid, L)$	Hilfszeichen durch \mid ersetzen
$\delta(q_3, \square) \rightarrow (q_f, \square, R)$	Halten

AUFGABE 8.5. (TM Akzeptanzbedingungen)

Stufe B

Begründen Sie folgende Aussagen, indem Sie eine passende Konstruktion angeben.

- Für jede Turing-Maschine $M \in \mathcal{M}_A$ gibt es eine Turing-Maschine $M' \in \mathcal{M}$ mit $L_F(M) = L_H(M')$.
- Für jede Turing-Maschine $M \in \mathcal{M}$ gibt es eine Turing-Maschine $M' \in \mathcal{M}_A$ mit $L_H(M) = L_F(M')$.
- Für jede Turing-Maschine $M \in \mathcal{M}_A$ gibt es eine Turing-Maschine $M' \in \mathcal{M}$ mit $L_F(M) = L_F(M')$.
- Für jede Turing-Maschine $M \in \mathcal{M}$ gibt es eine Turing-Maschine $M' \in \mathcal{M}_A$ mit $L_F(M) = L_F(M')$.
- Für jede Turing-Maschine $M \in \mathcal{M}$ gibt es eine Turing-Maschine $M' \in \mathcal{M}$ mit $L_F(M) = L_H(M')$.

Lösungsskizze

- Füge einen neuen Fangzustand q_t ein und mache die Transitionsfunktion total für Nichtendzustände. Formal:

$$Q' = Q \cup \{q_t\} \quad \delta' = \delta \cup \{(q_t, a, q_t, a, N) \mid a \in \Gamma\} \cup \{(q, a, q_t, a, N) \mid a \in \Gamma \wedge q \in Q \setminus F \wedge \delta(q, a) \text{ ist undefiniert}\}$$

- Füge einen neuen Endzustand q_f ein und mache die Transitionsfunktion total (außer für q_f). Formal:

$$Q' = Q \cup \{q_f\} \quad F' = \{q_f\} \quad \delta' = \delta \cup \{(q, a, q_f, a, N) \mid a \in \Gamma \wedge q \in Q \wedge \delta(q, a) \text{ ist undefiniert}\}$$

- Trivial, da $\mathcal{M}_A \subseteq \mathcal{M}$.
- Lösche alle ausgehenden Kanten von Endzuständen. Formal:

$$\delta' = \delta \setminus F \times \Gamma \times Q \times \Gamma \times \{L, R, N\}$$

- Lösche alle ausgehenden Kanten von Endzuständen und mache die Transitionsfunktion total für Nichtendzustände. Formal:

$$Q' = Q \cup \{q_t\} \\ \delta' = (\delta \cup \{(q_t, a, q_t, a, N) \mid a \in \Gamma\} \cup \{(q, a, q_t, a, N) \mid a \in \Gamma \wedge q \in Q \setminus F \wedge \delta(q, a) \text{ ist undefiniert}\}) \setminus F \times \Gamma \times Q \times \Gamma \times \{L, R, N\}$$

AUFGABE 8.6. (Aussagen über TMs)

Stufe B

Entscheiden Sie, ob die folgenden Aussagen korrekt sind und begründen Sie Ihre Antwort kurz.

- Es gibt eine Turingmaschine, die den Kopf nie weiter als vier Schritte von der Startposition weg bewegt und eine unendliche Sprache akzeptiert.
- Sei $M \in \mathcal{M}_A$ eine Turingmaschine. Dann existiert eine TM $M' \in \mathcal{M}$, so dass $L_F(M) = L_F(M')$ und M' hat nur einen Zustand.
- Sei $M \in \mathcal{M}_A$ eine Turingmaschine, die ihren Kopf immer nur nach links bewegt. Dann gilt:

$$L_F(M) \in \{A\Sigma^* \mid A \subseteq \Sigma\}$$

Lösungsskizze

- Ja. Sei $q_0 \in F$, dann erkennt die TM Σ^* .
- Nein. Entweder $q_0 \in F$, dann erkennt die TM M' die Sprache Σ^* , oder $q_0 \notin F$, dann erkennt M' die Sprache \emptyset . Somit können keine weiteren Sprachen erkannt werden, also auch nicht die jeder beliebigen TM M .
- Nein. Es kann zwar nur das erste Zeichen gelesen werden, aber auch dessen Abwesenheit kann erkannt werden. Somit haben wir:

$$L_F(M) \in \{A\Sigma^* \mid A \subseteq \Sigma\} \cup \{\{\varepsilon\} \cup \{A\Sigma^*\} \mid A \subseteq \Sigma\}$$

AUFGABE 8.7. (2-PDA und TMs)

Stufe D

Im Folgenden sagen wir, dass ein 2-PDA ein PDA ist, der 2 Stacks zur Verfügung hat. In jedem Schritt kann der PDA in Abhängigkeit vom aktuellen Zustand, dem gelesenen Eingabezeichen und den Symbolen, die zuoberst auf jedem der beiden Stacks liegen, in einen neuen Zustand wechseln und jeden der Stacks wie im Fall eines gewöhnlichen PDAs modifizieren.

- Geben Sie eine formale Definition für 2-PDAs an.
- Geben Sie eine Sprache an, die von einem 2-PDA, aber von keinem 1-PDA akzeptiert wird.
- Geben Sie eine allgemeine Übersetzung von einem PDA A in eine Turingmaschine M an, so dass $L_\varepsilon(A) = L_F(M)$. Verwenden Sie bei Bedarf eine TM mit mehreren Bändern.
- Beschreiben Sie wie Sie das Verfahren aus (c) erweitern können, so dass ein 2-PDA A in eine Turingmaschine M übersetzt werden kann.
- Geben Sie eine allgemeine Übersetzung von einer Turingmaschine M in einen 2-PDA A an, so dass $L_F(M) = L_F(A)$.
- Zeigen Sie unter Verwendung der vorherigen Ergebnisse, dass jeder k -PDA A ($k \geq 3$) von einem 2-PDA A' simuliert werden kann, d.h. $L_\varepsilon(A) = L_\varepsilon(A')$. Somit können beliebig viele Stacks immer durch genau zwei Stacks simuliert werden.

- (a) $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0^2, F)$ mit $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^2 \rightarrow \mathcal{P}_e(Q \times (\Gamma^*)^2)$.
- (b) Ein 2-PDA kann die Sprache $L = \{a^n b^n c^n \mid n \in \mathbb{N}_0\}$ akzeptieren, indem er zuerst alle a s auf den ersten, dann alle b s auf den zweiten Stack schiebt, dann für jedes c jeweils ein Zeichen von beiden Stacks poppt und schließlich nur dann akzeptiert, falls nach Lesen der Eingabe beide Stacks leer sind. Wie aber in der Vorlesung gezeigt worden ist, ist L nicht kontextfrei und somit existiert kein PDA, der L akzeptiert.
- (c) Sei $P = (Q_P, \Sigma, \Gamma, \delta_P, q_0, \perp)$ ein PDA. Wir konstruieren nun ein TM $M = (Q_M, \Sigma \cup \Gamma, \Sigma \cup \Gamma \cup \{\square\}, \delta_M, q_0, \square, \{f\})$ mit $Q_M = Q_P \cup Q_P \times \Gamma_P \cup \{f\}$, so dass M gerade $L_\varepsilon(P)$ akzeptiert. Wir nehmen an, dass der PDA mit leerem Keller akzeptiert und ein explizites Bottom-Symbol \perp verwendet. Die TM simuliert den Stack des PDA auf einem zweiten Band, wobei das zweite Band oBdA. zu Beginn mit \perp initialisiert ist. Das rechteste Symbol des zweiten Bandes entspricht dann dem obersten Symbol auf dem Stack des simulierten PDA. Damit:

$$\begin{aligned} \delta_M(p, a, X) &= \{(q, a, Y, R, N) \mid pX \xrightarrow{a} qY \in \delta_P\} \\ &\quad \cup \{(q, a, \square, R, L) \mid pX \xrightarrow{a} q\varepsilon \in \delta_P\} \\ &\quad \cup \{(q, Y), a, Z, R, R) \mid pX \xrightarrow{a} qYZ \in \delta_P\} \\ \delta_M(p, a, X) &= \{(q, a, Y, N, N) \mid pX \xrightarrow{\varepsilon} qY \in \delta_P\} \\ &\quad \cup \{(q, a, \square, N, L) \mid pX \xrightarrow{\varepsilon} q \in \delta_P\} \\ &\quad \cup \{(q, Y), a, Z, N, R) \mid pX \xrightarrow{\varepsilon} qYZ \in \delta_P\} \end{aligned}$$

zzgl.

$$\begin{aligned} \delta_M((q, Y), a, \square) &= \{(q, a, Y, N, N)\} && \text{für alle } a \in \Sigma \cup \{\square\}, q \in Q_P, Y \in \Gamma_P \\ \delta_M(q, \square, \square) &= \{(f, \square, \square, N, N)\} && \text{für alle } q \in Q_P \end{aligned}$$

- (d) Wir verwenden anstatt einer 2-Band eine 3-Band TM für einen 2-PDA.
- (e) Idee: Jeder 2-PDA kann eine TM mit einem Band simulieren, indem der erste Stack alle Symbole links vom Lesekopf, der zweite Stack alle Symbole rechts vom Lesekopf speichert. Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ die zu simulierende TM. Dann konstruieren wir A folgend:

$$A = (Q \cup \{q_{\text{read}}, q_{\text{rev}}\}, \Sigma, \Gamma \cup \{\perp\}, \delta, q_{\text{read}}, (\perp, \perp), F)$$

Zuerst kopieren wir die Eingabe auf das Band:

$$\begin{aligned} q_{\text{read}}(X, \perp) &\xrightarrow{\varepsilon} q_{\text{rev}}(X, \perp) && q_{\text{read}}(X, \perp) \xrightarrow{a} q_{\text{read}}(aX, \perp) && \text{für alle } a \in \Sigma, X \in \Sigma \cup \{\perp\} \\ q_{\text{rev}}(\perp, Y) &\xrightarrow{\varepsilon} q_0(\perp, Y) && q_{\text{rev}}(X, Y) \xrightarrow{\varepsilon} q_{\text{rev}}(\varepsilon, XY) && \text{für alle } X \in \Sigma \end{aligned}$$

Wir simulieren wie folgt die TM:

$$\begin{aligned} q(X, Y) &\xrightarrow{\varepsilon} p(\varepsilon, XZ) && \text{für alle } X \in \Gamma, (q, Y, p, Z, L) \in \delta \\ q(X, Y) &\xrightarrow{\varepsilon} p(X, Z) && \text{für alle } X \in \Gamma, (q, Y, p, Z, N) \in \delta \\ q(X, Y) &\xrightarrow{\varepsilon} p(ZX, \varepsilon) && \text{für alle } X \in \Gamma, (q, Y, p, Z, R) \in \delta \end{aligned}$$

Weiterhin kann bei Bedarf auf \perp immer ein \square erzeugt werden.

$$q(\perp, X) \xrightarrow{\varepsilon} q(\square\perp, X) \quad q(X, \perp) \xrightarrow{\varepsilon} q(X, \square\perp) \quad \text{für alle } q \in Q, X \in \Gamma \cup \{\perp\}$$

- (f) Eine $(k + 1)$ -Band-TM kann jeden k -PDA simulieren. Jede $k + 1$ -Band-TM kann von einer 1-Band-TM simuliert werden. Damit kann jeder 2-PDA jeden k -PDA simulieren und ist so mächtig wie jede TM.