Marco Kunze (`makunze@cs.tu-berlin.de`)
Sebastian Nowozin (`nowozin@cs.tu-berlin.de`)

WS 2001/2002
19. 12. 2001

Lösungen zur 2. Projektaufgabe TheGI1

Definition: Keller-Turingmaschinen

Eine *Keller-Turingmaschine* ist eine nichtdeterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \Box, k, \Delta, q_A, E)$, die folgende Bedingungen erfüllt:

- $k = 2$

  Sie besitzt genau ein Eingabe- und genau ein Arbeitsband
- $Q \times \Gamma^2 \times Q \times \Gamma^2 \times \{L\} \times \{L, N, R\} \cap \Delta = \emptyset$

  Das heißt, für das Eingabeband gibt es nur die Kopfbewegungen Rechts und Nicht-Bewegung.
- $Q \times \Gamma^2 \times Q \times \Gamma \times (\Gamma \setminus \{\Box\}) \times \{L, N, R\} \times \{L\} \cap \Delta = \emptyset$

  Das heißt, für jede Linksbewegung auf dem Arbeitsband muss ein Blank-Symbol $\Box$ geschrieben werden, es darf keine Überführungsrelation geben, bei der der eine Linksbewegung auf dem Arbeitsband ausgeführt wird und kein Blank-Symbol geschrieben wird.
- $Q \times \Gamma^2 \times Q \times \Gamma \times \{\Box\} \times \{L, N, R\} \times \{R\} \cap \Delta = \emptyset$

  Das heißt, für jede Rechtsbewegung auf dem Arbeitsband muss ein Symbol aus $\Gamma \setminus \{\Box\}$ geschrieben werden.

NTM für L

Wir definieren eine nichtdeterministische Keller-Turingmaschine (siehe Abbildung 1) für $L = \{a^n b^m c^k \mid n, m, k \in \mathbb{N}^+ \text{und } (n = m) \text{ oder } (n = k)\}$
Berechnung für $w = aabcc$:

$$\begin{aligned}
(q_A, \rhd aabcc, \rhd \Box) &\vdash (q_A, a \rhd abcc, a \rhd \Box) \\
&\vdash (q_3, aa \rhd bcc, a \rhd \Box) \\
&\vdash (q_4, aab \rhd cc, \rhd a) \\
&\vdash (q_4, aabc \rhd c, \rhd \Box) \\
&\vdash (q_5, aabcc \rhd \Box, \rhd \Box) \\
&\vdash (q_E, aabcc \rhd \Box, \rhd \Box)
\end{aligned}$$

Berechnung für $w = aabc$:

$$(q_A, \rhd aabc, \rhd \Box) \vdash (q_A, a \rhd abc, a \rhd \Box)$$

Es müssen jetzt beide möglichen Abzweigungen betrachtet werden.

- Variante $q_A \to q_1$:

$$(q_1, aa \rhd bc, \rhd a) \vdash (q_1, aab \rhd c, \rhd \Box)\|$$

- Variante $q_A \to q_3$:

$$\begin{aligned}
(q_3, aa \rhd bc, a \rhd \Box) &\vdash (q_4 aab, \rhd c, \rhd a) \\
&\vdash (q_4, aabc \rhd \Box, a \rhd \Box)\|
\end{aligned}$$

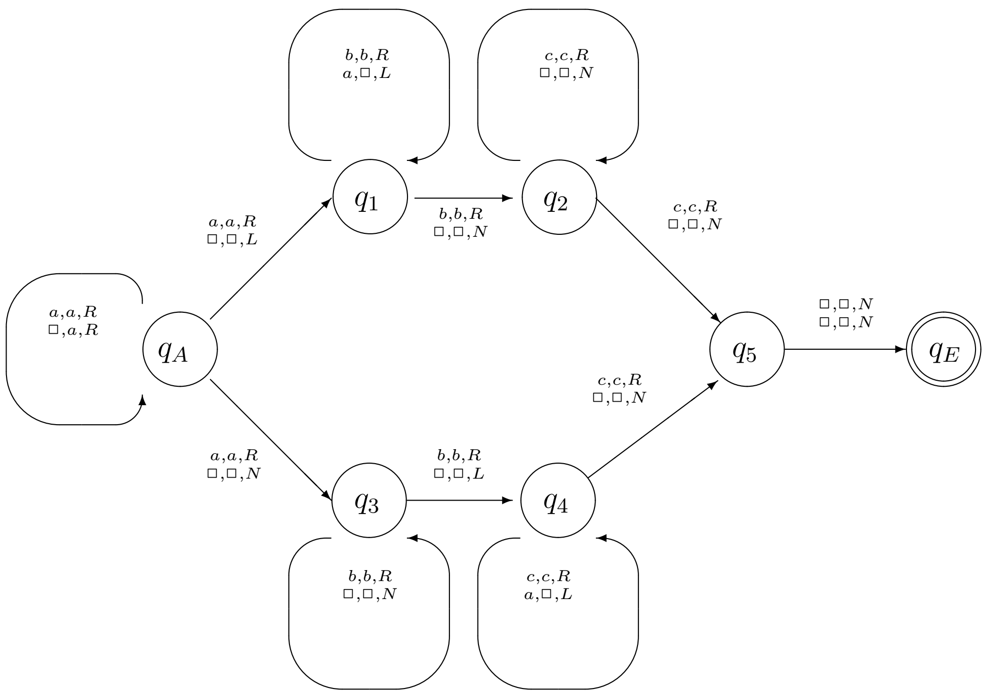

Abbildung 1: Nichtdeterministische Keller-Maschine M

Beweis für M

Beschreibung der Maschine:

$q_A \to q_A$	Kopiere die a's vom Eingabeband zum Arbeitsband
$q_A \to q_1$	Fordere ein a auf dem Eingabeband und spule das Arbeitsband eine Einheit zurück
$q_1 \to q_1$	Lösche für jedes b auf dem Eingabeband ein a vom Arbeitsband und gehe jeweils einen Schritt nach rechts beziehungsweise links
$q_1 \to q_2$	Fordere ein b auf dem Eingabeband
$q_2 \to q_2$	Lies ein c auf dem Eingabeband
$q_2 \to q_5$	Fordere ein c auf dem Eingabeband
$q_A \to q_3$	Fordere ein a auf dem Eingabeband
$q_3 \to q_3$	Lies ein b auf dem Eingabeband
$q_3 \to q_4$	Fordere ein b auf dem Eingabeband und spule das Arbeitsband eine Einheit zurück
$q_4 \to q_4$	Lösche für jedes c auf dem Eingabeband ein a vom Arbeitsband und gehe jeweils einen Schritt nach rechts beziehungsweise links
$q_4 \to q_5$	Fordere ein c auf dem Eingabeband
$q_5 \to q_E$	Fordere ein $\square$ auf dem Eingabeband (Wortende)

Beweis für $L = L(M)$:

- Behauptung $L \subseteq L(M)$

  Sei $w \in L$, also $w = a^n b^m c^k$ mit $n, m, k \in \mathbb{N}^+$ und entweder $n = m$ oder $n = k$. Dann ist zu zeigen, dass für w auch eine akzeptierende Berechnung in M existiert.

  – Für Fall $n = m$, also $w = a^n b^n c^k$:

$$\begin{aligned} (q_A, \rhd a^n b^n c^k, \rhd \square) &\vdash^{n-1} (q_A, a^{n-1} \rhd ab^n c^k, a^{n-1} \rhd \square) \\ &\vdash (q_1, a^n \rhd b^n c^k, a^{n-1} \rhd \square) \end{aligned}$$

$$
\begin{aligned}
&\vdash^{n-1} && (q_1, a^n b^{n-1} \rhd bc^k, \rhd \Box) \\
&\vdash && (q_2, a^n b^n \rhd c^k, \rhd \Box) \\
&\vdash^{k-1} && (q_2, a^n b^n c^{k-1} \rhd c, \rhd \Box) \\
&\vdash && (q_5, a^n b^n c^k \rhd \Box, \rhd \Box) \\
&\vdash && (q_E, a^n b^n c^k \rhd \Box, \rhd \Box) \quad \|
\end{aligned}
$$

– Für Fall $n = k$, also $w = a^n b^m c^n$:

$$
\begin{aligned}
(q_A, \rhd a^n b^m c^n, \rhd \Box) &\vdash^{n-1} && (q_A, a^{n-1} \rhd ab^m c^n, a^{n-1} \rhd \Box) \\
&\vdash && (q_3, a^n \rhd b^m c^n, a^{n-1} \rhd \Box) \\
&\vdash^{m-1} && (q_3, a^n b^{m-1} \rhd bc^n, \rhd \Box) \\
&\vdash && (q_4, a^n b^m \rhd c^n, \rhd \Box) \\
&\vdash^{n-1} && (q_4, a^n b^m c^{n-1} \rhd c, \rhd \Box) \\
&\vdash && (q_5, a^n b^m c^n \rhd \Box, \rhd \Box) \\
&\vdash && (q_E, a^n b^m c^n \rhd \Box, \rhd \Box) \quad \|
\end{aligned}
$$

Damit ist gezeigt, dass für alle w aus L eine Berechnung in M existiert. Also $w \in L(M) \Rightarrow L \subseteq L(M)$.

- Behauptung: $L(M) \subseteq L$

  Beweis: Sei $w \in L$. Also gibt es eine Berechnung von M zur Eingabe w_j (j sei die Länge von w)

  $$(q_A, \rhd w_j, \rhd \Box) \vdash_M^* (q_E, w_j \rhd \Box, \rhd \Box)$$

  Die Berechnung sieht so aus:

  – Für Fall "oben": Zuerst wird a^n auf das Arbeitsband kopiert:

  $$(q_A, \rhd w_j, \rhd \Box) \vdash^n (q_A, a^n \rhd w_{j-n}, a^n \rhd \Box)$$

  Ein a wird für den Übergang nach q_1 verwendet, somit ist sichergestellt, dass die Anzahl der a's > 0 ist:

  $$\vdash (q_1, a^n a \rhd w_{j-n-1}, a^{n-1} \rhd a)$$

  Nun wird für jedes b auf dem Eingabeband ein a auf dem Arbeitsband gelöscht, somit ist sichergestellt, dass die Anzahl der b's gleich der Anzahl der a's ist:

  $$\vdash^n (q_1, a^n ab^n \rhd w_{j-2n-1}, \rhd \Box)$$

  Ein b wird für den Übergang nach q_2 verwendet, somit ist sichergestellt, dass die Anzahl der b's > 0 ist:

  $$\vdash (q_2, a^n ab^n b \rhd w_{j-2n-2}, \rhd \Box)$$

  Die c's werden überlesen:

  $$\vdash^k (q_2, a^n ab^n bc^k \rhd w_{j-2n-2-k}, \rhd \Box)$$

  Ein c wird für den Übergang nach q_5 verwendet, somit ist sichergestellt, dass die Anzahl der c's > 0 ist:

  $$\vdash (q_5, a^n ab^n bc^k c \rhd \Box, \rhd \Box)$$

  Abschließend wird überprüft, ob Eingabe- und Arbeitsband leer sind:

  $$\vdash (q_E, a^n ab^n bc^k c \rhd \Box, \rhd \Box)$$

Da bei jedem Übergang von q_A nach q_E mindestens ein a, b und c gefordert werden, ist gewährleistet, dass nur $w = a^x b^y c^z$ mit $x, y, z \in \mathbb{N}^+$ akzeptiert werden.

Ebenso wird durch diese Übergänge eindeutig bestimmt, dass die a, b und c nur in dieser Reihenfolge im Wort vorkommen dürfen.

Der letzte Schritt stellt außerdem sicher, dass nach den c's (dem c) keine Zeichen mehr folgen.

Damit ist der Fall "oben" bewiesen.

– Für Fall "unten":

Zuerst wird a^n auf das Arbeitsband kopiert:

$$(q_A, \rhd w_j, \rhd \Box) \vdash^n (q_A, a^n \rhd w_{j-n}, a^n \rhd \Box)$$

Ein a wird für den Übergang nach q_1 verwendet, somit ist sichergestellt, dass die Anzahl der a's > 0 ist:

$$\vdash (q_3, a^n a \rhd w_{j-n-1}, a^n \rhd \Box)$$

Die b's werden überlesen:

$$\vdash^m (q_3, a^n a b^m \rhd w_{j-n-1-m}, a^n \rhd \Box)$$

Ein b wird für den Übergang nach q_4 verwendet, somit ist sichergestellt, dass die Anzahl der b's > 0 ist:

$$\vdash (q_4, a^n a b^m b \rhd w_{j-n-2-m}, a^{n-1} \rhd a)$$

Nun wird für jedes c auf dem Eingabeband ein a auf dem Arbeitsband gelöscht, somit ist sichergestellt, dass die Anzahl der c's gleich der Anzahl der a's ist:

$$\vdash^n (q_4, a^n a b^m b c^n \rhd w_{j-2n-2-m}, \rhd \Box)$$

Ein c wird für den Übergang nach q_5 verwendet, somit ist sichergestellt, dass die Anzahl der c's > 0 ist:

$$\vdash (q_5, a^n a b^m b c^n c \rhd \Box, \rhd \Box)$$

Abschließend wird überprüft, ob Eingabe- und Arbeitsband leer sind:

$$\vdash (q_E, a^n a b^m b c^n c \rhd \Box, \rhd \Box)$$

Da bei jedem Übergang von q_A nach q_E mindestens ein a, b und c gefordert werden, ist gewährleistet, dass nur $w = a^x b^y c^z$ mit $x, y, z \in \mathbb{N}^+$ akzeptiert werden.

Ebenso wird durch diese Übergänge eindeutig bestimmt, dass die a, b und c nur in dieser Reihenfolge im Wort vorkommen dürfen.

Der letzte Schritt stellt außerdem sicher, dass nach den c's (dem c) keine Zeichen mehr folgen.

Damit ist der Fall "oben" bewiesen.

Daraus folgt zusammenfassend: $L(M) \subseteq L$.

Also ist $L = L(M)$

Kontextfreie Grammatik → Keller-Maschine

Konstruktion

Sei G mit $G = (N, T, S, P)$ eine beliebige kontextfreie Grammatik. Dann kann eine äquivalente Keller-Turingmaschine M konstruiert werden, so dass $L(G) = L(M)$.
Hierzu konstruiert man M mit

- $\Sigma = T$

  Die Eingabesymbole von M sind die terminalen Symbole der Grammatik G.

- $\Gamma = T \cup N \cup \{\square\}$

  Das Bandalphabet besteht aus allen Symbolen der Grammatik plus einem Blank-Symbol.

Die allgemeine Form der zur Simulation einer kontextfreien Grammatik verwendeten Keller-Maschine ist in Abbildung 2 zu sehen.

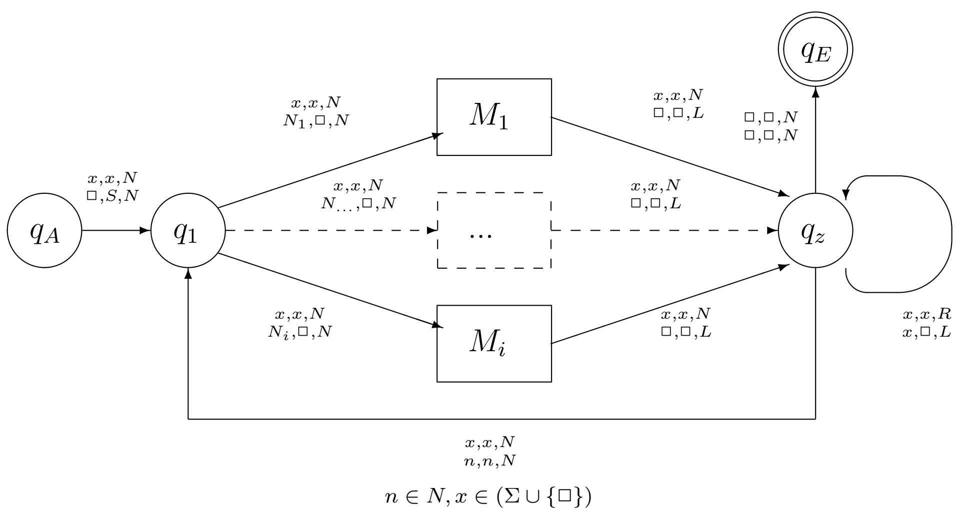

Abbildung 2: Keller-Maschine M zur Simulation einer kontextfreien Grammatik, mit $x \in (\Sigma \cup \{\square\})$, $n \in N$

Dabei besitzt M folgende wichtige Komponenten

- Die Produktionsmaschinen M_1, M_2, …, M_i

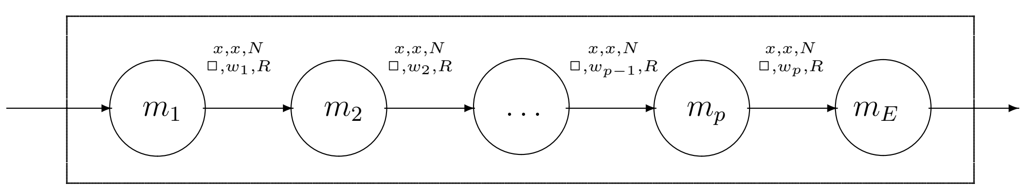

Abbildung 3: Produktionsmaschinen M_i

Die Produktionsmaschinen M_1, M_2, …, M_i, entsprechen jeweils genau einer Produktion der Grammatik. Die allgemeine Form einer solchen Untermaschine ist in Abbildung 3 gezeigt. Dabei gilt für jede Produktion $N_i \to v$, mit $N_i \in N$ und $v \in (N \cup T)^\star$:

$$w = w_1 w_2 w_3 \dots w_p = v^{-1}$$

Damit ist w die Umkehrung des produzierten Wortes der Produktion.

Dies wird benötigt, da wir - wie wir später zeigen werden - mit der Maschine ausschließlich am Ende des Kellerbandes, also am linken Wortanfang des erzeugten Wortes arbeiten. Wenn wir die rechten Seiten der Produktionen umkehren, produzieren wir damit also die Linksableitungen der ursprünglichen Grammatik (siehe Lemma 6.3, Skript). Alternativ könnte man auch unsere Definition der Kellerturingmaschinen aus dem ersten Aufgabenteil so ändern, dass der Keller nach links wächst. Für unsere allgemeine Maschine verwenden wir jedoch einen nach rechts wachsenden Keller.

- Der Vergleichszustand q_z

  Der Vergleichszustand q_z erkennt terminale Symbole auf dem Eingabe- und Kellerband. Er entfernt solange terminale Symbole vom rechten Rand des Kellerbandes wie das jeweils rechteste Symbol mit dem aktuellen Symbol des Eingabebandes übereinstimmt. Dabei wird auch der Lesekopf des Eingabebandes einen Schritt nach rechts bewegt. Hierbei gibt es drei mögliche Abbruchbedingungen:

  - q_z hat alle Symbole vom Kellerband entfernt und befindet sich am Ende des Wortes auf dem Eingabeband. Dann wird in den Zustand q_E überführt, der immer terminiert und akzeptiert. Damit akzeptiert M das Eingabewort.
  - q_z hat zwar alle Symbole vom Kellerband entfernt, aber befindet sich nicht am Ende des Eingabewortes. Dann terminiert M in q_z nicht akzeptierend.
  - q_z stösst beim Entfernen der Symbole vom Kellerband auf ein Nonterminal. Dann führt sie zurück in den "Produktor", q_1, der Keller-Maschine, der die Berechnungen bildet.

Der Nichtdeterminismus der Maschine wird benötigt für die Übergänge von q_1 zu den Produktionsmaschinen. Sind in der kontextfreien Grammatik mehrere Produktionen für ein Nonterminales Symbol gegeben, so ist die Wahl der Produktionsmaschine hier nicht eindeutig. Bei einer NTM ist das natürlich kein Problem.

Beweis

Um die Äquivalenz der von der allgemeinen Kellerturingmaschine M akzeptierten Sprache und der von der kontextfreien Grammatik G erzeugten Sprache zu zeigen, beweisen wir zwei Richtungen:

- $L(G) \subseteq L(M)$

  Sei $w \in T^\star$ ein Wort der von G erzeugten Sprache. Das heisst $w \in L(G)$, also ist w in G ableitbar und es existiert eine Ableitungskette. Dann müssen wir zeigen, dass w von M akzeptiert wird. Dazu reicht es aus, eine akzeptierende Berechnung in M bei Eingabe von w zu finden.

  Für die Maschine nehmen wir ohne Beschränkung der Allgemeinheit die Links-Ableitung des Wortes w in der Grammatik G an. Die Maschine vollführt auf ihrem Arbeitsband die Ableitungsschritte nach, die bei einer Linksableitung in der Grammatik G auftauchen. Dabei ist der Inhalt des Kellerbands immer von rechts nach links, also rückwärts zu lesen.

  Jeder Ableitungsschritt für das Links-Ableiten eines Wortes in einer kontextfreien Grammatik hat die Form:

  $$unv \Rightarrow upv$$

  $$n \in N \quad u \in T^\star \quad v, p \in \{N \cup T\}^\star$$

  Das längste Wort, dass nur aus terminalen Symbolen besteht und gleich dem Anfang von p ist, bezeichnen wir im Folgenden als t. Also $p = tp_{rest}$, mit $t \in T^\star, p, p_{rest} \in \{N \cup T\}^\star$ und wenn $p_{rest} \neq \lambda$, dann $p_0 \in N$.

Da für ein $t \neq \lambda$, t immer nur aus terminalen Zeichen besteht, und die Produktionen keinen Kontext verändern können (sie sind ja per Definition *kontextfrei*), ist zwingend ut das Teilwort, mit dem w anfängt.

Das rechte Ende des Kellerbandes enthält nach jedem Ableitungsschritt immer das jeweilige t^{-1} dieses Schrittes. Dies muss zwingend der Fall sein, da die Maschine mit Hilfe der Produktionsmaschinen den Ableitungsschritt nachvollzogen hat. Aufgrund der Konstruktion von M_1, M_2, …, M_i ist klar, dass wenn die Berechnung die Produktionsmaschinen verlässt, die jeweilige Produktion auf dem Kellerband simuliert wurde:

$$(q_1, u \rhd w_{rest}, v \rhd N\Box) \vdash^+ (q_z, w_{alt} \rhd w_{rest}, vp_{rest}^{-1}t^{-1} \rhd)$$

In der akzeptierenden Berechnung entfernt q_z nun diese übereinstimmenden terminalen Symbole der Grammatik vom Kellerband, und liest sie auf dem Eingabeband:

$$(q_z, u \rhd tw_{rest2}, vp_{rest}^{-1}t^{-1} \rhd \Box) \vdash^\star (q_z, ut \rhd w_{rest2}, vp_{rest}^{-1} \rhd \Box)$$

Das heisst jeder Ableitungsschritt kann durch die Maschine nachvollzogen werden. Also ist bei Erreichen des Endes des Eingabewortes im Zustand q_z, auch der Lesekopf des Kellerbands auf einem Blank positioniert. Daher wird der Übergang nach q_E vollführt und die Maschine akzeptiert das Eingabewort w.

Also $L(G) \subseteq L(M)$.

- $L(M) \subseteq L(G)$

  Sei w^{-1} von M akzeptiert. Also gibt es eine Konfigurationsfolge, die mit der Anfangskonfiguration $(q_A, \rhd w^{-1}, \rhd \Box)$ startet und in einer akzeptierenden Konfiguration endet.

  Die Konfigurationsfolge entspricht genau einer Ableitungskette in der Grammatik. Hierzu zerlegen wir die Konfigurationsfolgen in Teilstücke. Ein solches Teilstück reicht immer von dem Erreichen des Zustands q_1 bis zum Verlassen des Zustandes q_z. Da es durch die Konstruktion der Maschine notwendig ist, dass bei den Übergängen, die von q_1 nach q_z führen, eine Produktionsmaschine durchlaufen werden muss, hat dieses Teilstück immer die Form:

$$(q_1, u \rhd v, r^{-1} \rhd n) \vdash^+ (q_z, u \rhd v, r^{-1}z^{-1} \rhd \Box)$$
$$u, v \in T^\star, \quad r, z \in \{T \cup N\}^\star, n \in N$$

  Durch die Konstruktion der Produktionsmaschine entspricht dieses Teilstück genau einem Ableitungsschritt der Form:

$$nr \Rightarrow_G zr$$

  In der akzeptierenden Berechnung werden jetzt alle terminalen Symbole oben vom Keller (das heisst am linken Wortanfang) entfernt, bis ein Nonterminal oder das Wortende auftritt. Dieser Schritt ist bei der Ableitung, die w erzeugt nicht nötig, da alle terminalen Symbole per Definition nicht weiter ersetzt werden können.

  Für jedes Teilstück existiert also ein entsprechender Ableitungsschritt in der Grammatik. Da in jedem dieser Teilstücke in q_z alle terminalen Symbole am Kellerende mit denen des Eingabewortes verglichen und gegebenenfalls entfernt werden, muss beim letzten Schritt das Wortende erreicht werden. Die Maschine akzeptiert das Wort w^{-1}, indem jetzt ein Übergang $q_z \rightarrow q_E$ stattfindet. Analog geschieht dies beim letzten Ableitungsschritt in der Grammatik, da keine Nonterminale mehr im Wort stehen, und w damit erzeugt ist.

  Also gibt es eine Ableitungskette $S \Rightarrow_G^\star w$, was zu beweisen war.

  Also $L(M) \subseteq L(G)$.

Beispielmaschine

Für unsere Beispielgrammatik G benutzen wir folgende Produktionen:

$$\begin{aligned} S &\rightarrow aSb && (1)\\ S &\rightarrow cSd && (2)\\ S &\rightarrow \lambda && (3) \end{aligned}$$

Daraus konstruieren wir mit Hilfe der vorher festgelegten Konstruktionsvorschriften eine Maschine M (siehe Abbildung 4). Dabei entspricht Produktion (1) den Zuständen q_2, q_3, q_4, q_5, Produktion (2) den Zuständen q_6, q_7, q_8, q_9, und die Produktion (3) dem Zustand q_{10}.

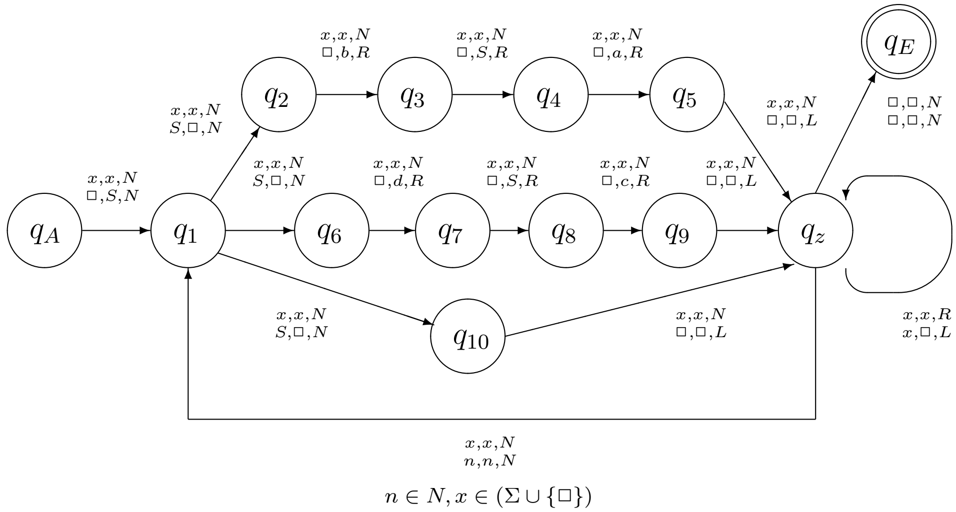

Abbildung 4: Nichtdeterministische Keller-Maschine für die spezielle Grammatik G

Die Maschine akzeptiert das Eingabewort w genau dann, wenn $w \in L(G)$. Also ist $L(G) = L(M)$. Als Beispiel nehmen wir das Wort "$acdb$" und "λ":

$$\begin{aligned}
(q_A, \triangleright\, acdb, \triangleright\, \square) &\vdash (q_1, \triangleright\, acdb, \triangleright\, S) \vdash (q_2, \triangleright\, acdb, \triangleright\, \square)\\
&\vdash (q_3, \triangleright\, acdb, b \triangleright \square) \vdash (q_4, \triangleright\, acdb, bS \triangleright \square)\\
&\vdash (q_5, \triangleright\, acdb, bSa \triangleright \square) \vdash (q_z, \triangleright\, acdb, bS \triangleright a)\\
&\vdash (q_z, a \triangleright cdb, b \triangleright S) \vdash (q_1, a \triangleright cdb, b \triangleright S)\\
&\vdash (q_6, a \triangleright cdb, b \triangleright \square) \vdash (q_7, a \triangleright cdb, bd \triangleright \square)\\
&\vdash (q_8, a \triangleright cdb, bdS \triangleright \square) \vdash (q_9, a \triangleright cdb, bdSc \triangleright \square)\\
&\vdash (q_z, a \triangleright cdb, bdS \triangleright c) \vdash (q_1, ac \triangleright db, bd \triangleright S)\\
&\vdash (q_{10}, ac \triangleright db, bd \triangleright \square) \vdash (q_z, ac \triangleright db, b \triangleright d)\\
&\vdash (q_z, acd \triangleright b, \triangleright\, b) \vdash (q_z, acdb \triangleright \square, \triangleright\, \square)\\
&\vdash (q_E, acdb \triangleright \square, \triangleright\, \square)|
\end{aligned}$$

$$\begin{aligned}
(q_A, \triangleright\, \square, \triangleright\, \square) &\vdash (q_1, \triangleright\, \square, \triangleright\, S) \vdash (q_10, \triangleright\, \square, \triangleright\, \square)\\
&\vdash (q_z, \triangleright\, \square, \triangleright\, \square) \vdash (q_E, \triangleright\, \square, \triangleright\, \square)|
\end{aligned}$$