

- 1) Wir verwandeln den Automaten in eine Turingmaschine, die das Band genau einmal von links nach rechts durchläuft (und dabei exakt die gleichen Zustände durchläuft wie der Automat) ohne jemals das Band zu verändern und ohne jemals wieder nach links zu gehen. Sobald sie am Ende angelangt ist, geht sie entweder in den akzeptierenden Zustand t oder den verwerfenden Zustand r , je nachdem, ob der aktuelle Zustand ein Endzustand (also in F) ist oder nicht.

Wir fügen also zu dem Automaten einfach zwei neue Zustände t (akzeptierend) und r (verwerfend) hinzu und wandeln jedem Übergang von einem Zustand q mit gelesenen Zeichen a zu einem Zustand q' um zu einem Übergang von q und aktuellem Bandzeichen a , der wieder a zurück auf das Band schreibt, nach rechts geht, und in den Zustand q' .

Wir müssen außerdem zumindest im Startzustand einen Übergang hinzufügen, der für das Bandsymbol \vdash einen Schritt nach rechts geht, da die Turingmaschine für Eingabewort $w = w_1 \dots w_n$ mit Bandinhalt $\vdash w_1 \dots w_n \sqcup^\infty$ startet und dem Schreib-/Lesekopf auf dem Anfangszeichen \vdash , während der Automat auf w_1 startet.

Formaler gesagt: Wir definieren die Maschine M mit

$$M = (Q \cup \{t, r\}, \Sigma, \Sigma \cup \{\vdash, \sqcup\}, \vdash, \sqcup, \delta', s, t, r)$$

wobei die Übergangsfunktion δ' definiert ist als

$$\delta'(q, a) = \begin{cases} (\delta(q, a), a, R) & \text{if } a \in \Sigma \\ (q, \vdash, R) & \text{if } a = \vdash \\ (t, \sqcup, R) & \text{if } q = \sqcup \text{ and } q \in F \\ (r, \sqcup, R) & \text{if } q = \sqcup \text{ and } q \notin F \end{cases}$$

für alle $q \in Q$ und $a \in \Sigma \cup \{\vdash, \sqcup\}$ und natürlich $\delta'(t, a) = (t, a, R)$ und $\delta'(r, a) = (r, a, R)$ für alle $a \in \Sigma \cup \{\vdash, \sqcup\}$.

- 2) Die Grammatik G beschreibt die Sprache $\{ab\} \cup \{a^n cb^n \mid n > 0\}$.

Folgende Turing Maschine M akzeptiert $L(G)$:

$M = (\{s, t, r, v, u, X, Y, Z, H, E\}, \{a, b, c\}, \{\vdash, \sqcup, a, b, c, x\}, \vdash, \sqcup, \delta, s, t, r)$ mit δ wie folgt

δ	\vdash	a	b	c	x	\sqcup
s	(s, \vdash, R)	(v, a, R)	-	-	-	-
v	-	(X, a, L)	(u, b, R)	(X, c, L)	(X, x, L)	(X, \sqcup, L)
u	-	(X, a, L)	(X, b, L)	(X, c, L)	(X, x, L)	+
X	(Y, \vdash, R)	(X, a, L)	(X, b, L)	(X, c, L)	(Y, x, R)	(X, \sqcup, L)
Y	-	(Z, x, R)	-	(E, c, R)	(Y, x, R)	-
Z	-	(Z, a, R)	(Z, b, R)	(Z, c, R)	(H, x, L)	(H, \sqcup, L)
H	-	-	(X, x, L)	-	-	-
E	-	-	-	-	(E, x, R)	+
t	+	+	+	+	+	+
r	-	-	-	-	-	-

wobei $- := (r, *, R)$ und $+ := (t, *, R)$.

M kann online¹ simuliert werden:

```

s - - r s
s a a r t

t b b r u
t * * l X

u _ _ r halt-accept
u * * l X

; X = gehe bis zum linken Ende
X - - r Y
X x x r Y
X * * l X

; Y = a ersetzen oder mittleres c lesen wenn alle a mit x ersetzt
Y x x r Y
Y a x r Z
Y c c r E

; Z = gehe bis zum rechten Ende
Z a a r Z
Z b b r Z
Z c c r Z
Z x x l H
Z _ _ l H

; H = b ersetzen und zum linken Ende
H b x l X

; E = ueberpruefen ob alle Zeichen nach dem c ersetzt
E x x r E
E _ _ r halt-accept

* * * * halt-reject

```

¹<https://morphett.info/turing/turing.html?3e73c7613df4d5c91d4d7286cbc477cb>

Wir haben den folgenden Satz in der Vorlesung kennengelernt:

Sei L eine Sprache, die von einer TM akzeptiert wird. Dann ist L rekursiv aufzählbar.

Da unsere TM M die Sprache $L(G)$ akzeptiert, ist $L(G)$ rekursiv aufzählbar.

3) *Lösung.* Die folgende Turingmaschine M akzeptiert die gewünschte Sprache

$$M = (\{X, Y, Z, t, r\}, \{[,]\}, \{[,], x, \sqcup, \vdash\}, \sqcup, \vdash, \delta, X, t, r),$$

wobei die Übergangsfunktion δ wie folgt definiert ist:

δ	\vdash	$[$	$]$	x	\sqcup
X	(X, \vdash, R)	$(X, [, R)$	(Y, x, L)	(X, x, R)	(Z, \sqcup, L)
Y	-	(X, x, R)	-	(Y, x, L)	-
Z	(t, \vdash, R)	-	-	(Z, x, L)	-
t	+	+	+	+	+
r	-	-	-	-	-

wobei $- := (r, *, R)$ und $+ := (t, *, R)$.

Für jede schließende Klammer $]$ wird zu ihrer linken nach einer öffnenden Klammer $[$ gesucht, wobei die jeweiligen Klammern durch das Symbol x ersetzt werden. M akzeptiert die Eingabe genau dann wenn alle Klammernpaare durch x ersetzt werden können.

M kann online² simuliert werden:

```

X - - r X
X [ [ r X
X ] x l Y
X x x r X
X - - l Z

Y [ x r X
Y x x l Y

Z - - r halt-accept
Z x x l Z

* * * * halt-reject

```

Um Klammernausdrücke über den Symbolen $[,], (,)$ zu beschreiben genügt es, M so zu adaptieren, dass man mit Hilfe von zusätzlichen Zuständen zwischen den Klammern unterscheidet.

□

²<https://morphett.info/turing/turing.html?52dc6672a74a8b67d46bcb69674b6aea>