Grundlagen der Theoretischen Informatik

Till Mossakowski

Fakultät für Informatik
Otto-von-Guericke Universität
Magdeburg

Wintersemester 2014/15

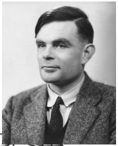
Rekursiv aufzählbare Sprachen

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_R$ which will be called "m-configurations". The machine is supplied with a "tape" (the analogue of paper) running through it, and divided into sections (called "squares") each capable of bearing a "symbol". At any moment there is just one square, say the r-th, bearing the symbol $\mathfrak{S}(r)$ which is "in the machine". We may call this square the "scanned square". The symbol on the scanned square may be called the "scanned symbol". The "scanned symbol" is the only one of which the machine is, so to speak, "directly aware". However, by altering its m-configuration the machine can effectively remember some of the symbols which it has "seen" (scanned) previously. The possible behaviour of the machine at any moment is determined by the m-configuration q_n and the scanned symbol $\mathfrak{S}(r)$. This pair $q_n, \mathfrak{S}(r)$ will be called the "configuration": thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (*i.e.* bears no symbol) the machine writes down a new symbol on the scanned square: in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or left. In addition to any of these operations the m-configuration may be changed. Some of the symbols written down

Deterministische Turing-Maschinen

Definition:
Eine deterministische Turing-Maschine ist ein 7-Tupel $(K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$, *wobei gilt:*

- K *ist eine endliche Menge von Zuständen,*
- Σ *ist ein Alphabet, das Eingabealphabet, das das Blanksymbol* $\sqcup$ *nicht enthält,*
- Γ *ist ein Alphabet, das Bandalphabet, das alle Elemente von* Σ *und das Blanksymbol* $\sqcup$ *enthält,*
- $s \in K$ *ist der Startzustand,*
- $q_{\text{accept}} \in K$ *ist der akzeptierende Halteszustand,*
- $q_{\text{reject}} \in K$ *ist der verwerfende Halteszustand,* $q_{\text{reject}} \neq q_{\text{accept}}$,
- $\delta : (K - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \to K \times \Gamma \times \{L, R, S\}$ *ist die Übergangsfunktion.*




GREAT. A WAREHOUSE FILLED WITH MILES AND MILES OF REWRITABLE TAPE! WHAT ARE WE EVER GOING TO DO WITH THIS, ALAN?
...ALAN?

$$\delta : (K - \{q_{\mathsf{accept}}, q_{\mathsf{reject}}\}) \times \Gamma \mapsto K \times \Gamma \times \{L, R, S\}$$

Wie in Turings Originalarbeit beschrieben, ändert eine Turing-Maschine ihren Zustand abhängig vom aktuellen Zustand und dem Symbol unter dem Schreib-Lesekopf. Dabei wird das aktuelle Symbol unter dem Schreib-Lesekopf durch ein Symbol ersetzt und der Schreib-Lesekopf bewegt sich entweder eine Position nach links (L) oder eine nach rechts (R) oder gar nicht, d.h., der Kopf bleibt stehen (S).

Erreicht die Maschine einen Haltezustand, so gibt es keine weiteren Aktionen.

Wie Turings Originalmaschine besitzt unsere Maschine ein beidseitig unendliches Band:

Im Anfangszustand befindet sich jede Turing-Maschine $M = (K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ im Zustand s und die Eingabe $w \in \Sigma^*$ steht auf $|w|$ aufeinanderfolgenden Quadraten des Eingabebandes. Auf allen übrigen Quadraten steht das Blanksymbol $\sqcup$. Der Schreib-Lesekopf befindet sich über dem ersten Symbol des Eingabewortes, bzw. über dem Blanksymbol, falls die Eingabe das leere Wort ε ist.

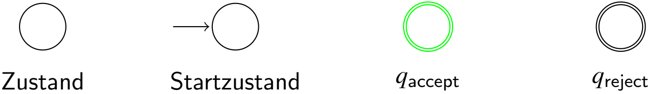
Beispiel:

$$M = (\{q, q_a, q_b, q_{\text{accept}}, q_{\text{reject}}\}, \{a,b\}, \{a,b,\sqcup\}, \delta, q, q_{\text{accept}}, q_{\text{reject}})$$

δ	a	b	$\sqcup$
q	$(q_a, \sqcup, R)$	$(q_b, \sqcup, R)$	$(q_{\text{accept}}, \sqcup, S)$
q_a	(q_a, a, R)	(q_b, a, R)	$(q_{\text{accept}}, a, R)$
q_b	(q_a, b, R)	(q_b, b, R)	$(q_{\text{accept}}, b, R)$

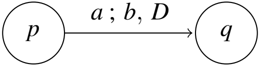
M verschiebt ein Eingabewort $w \in \{a,b\}^*$ auf dem Eingabeband um eine Position nach rechts.

Graphische Darstellung der Übergangsfunktion:

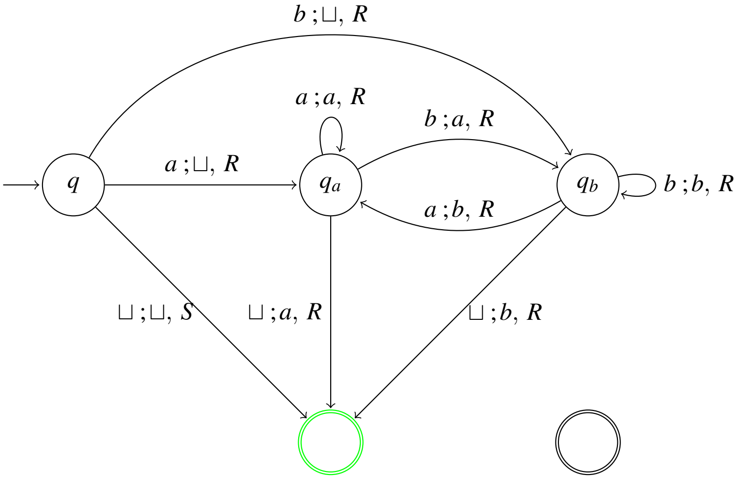


$$\delta(p,a) = (q,b,D)$$

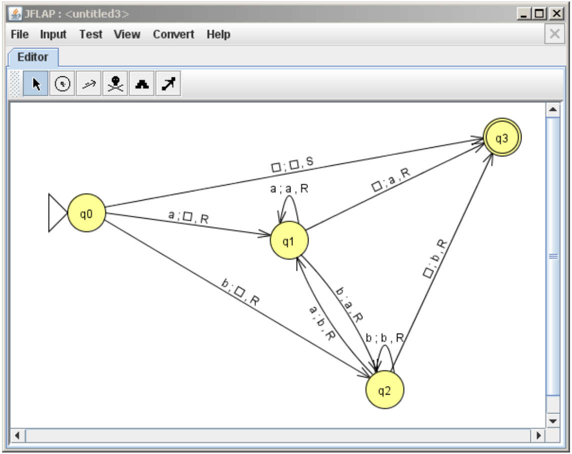
$$p,q \in K,\ a,b \in \Gamma,\ D \in \{L,R,S\}$$

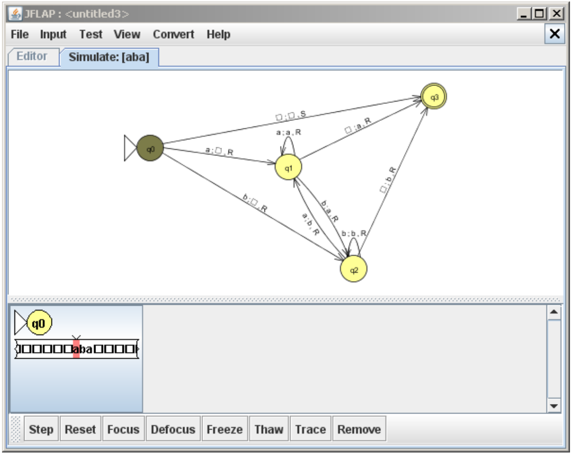


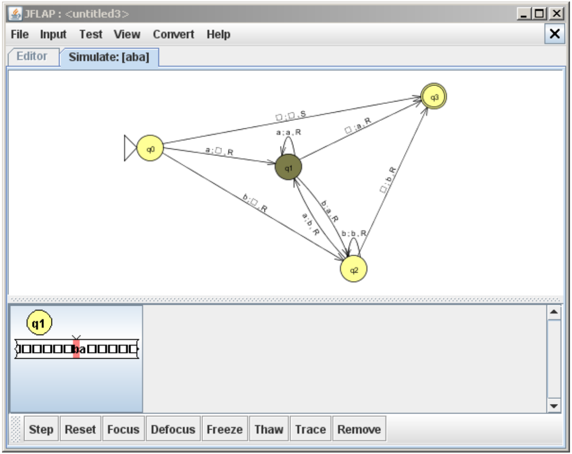
Beispiel:

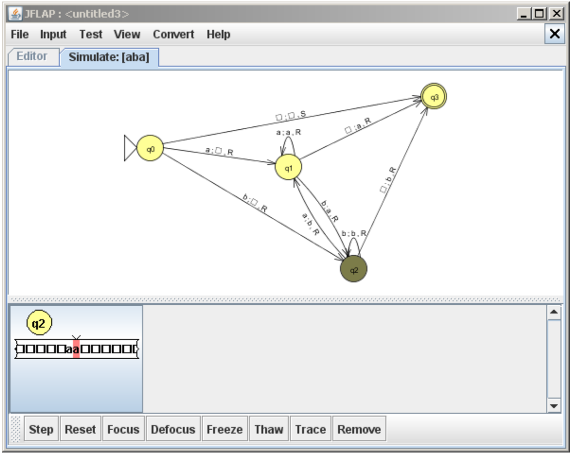


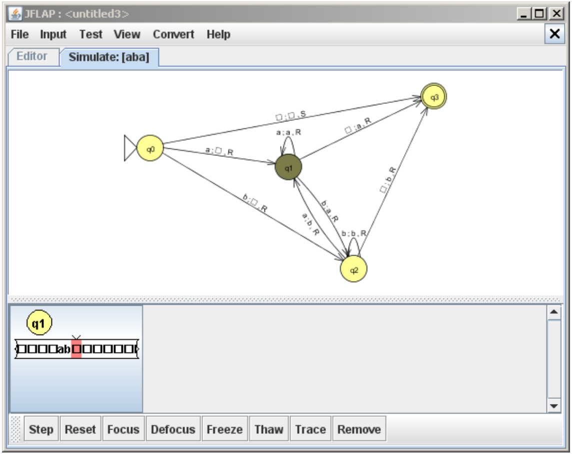
JFLAP kann Berechnungen von Turing-Maschinen simulieren:

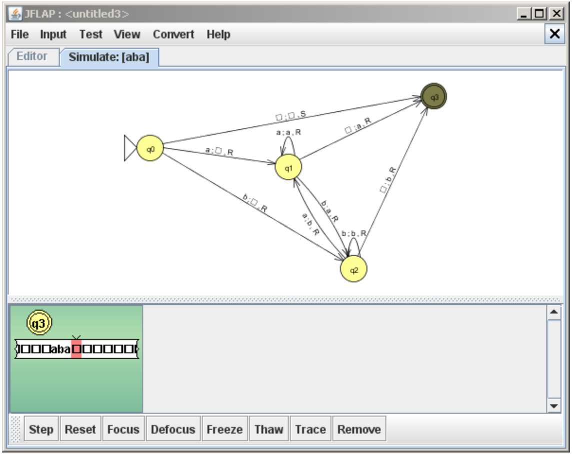



JFLAP : <untitled3>
File Input Test View Convert Help
Editor
Simulate: [aba]
q0
q1
q2
q3
Step Reset Focus Defocus Freeze Thaw Trace Remove

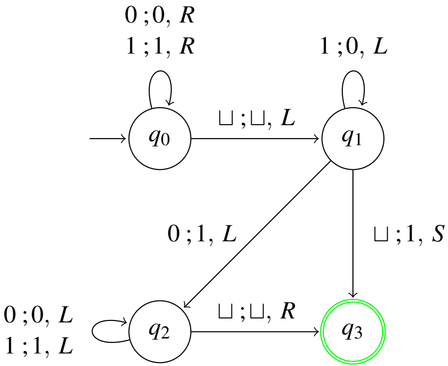

JFLAP : <untitled3>
File Input Test View Convert Help
Editor
Simulate: [aba]
q0
q1
q2
q3
□;□,S
a;a,R
□;a,R
a;□,R
b;□,R
□;b,R
b;a,R
a;b,R
b;b,R
Step Reset Focus Defocus Freeze Thaw Trace Remove


JFLAP : <untitled3>
File Input Test View Convert Help
Editor
Simulate: [aba]
q0
q1
q2
q3
□;□,S
a;a,R
a;□,R
□;a,R
b;□,R
b;a,R
a;b,R
b;b,R
□;b,R
q2
Step
Reset
Focus
Defocus
Freeze
Thaw
Trace
Remove


JFLAP : <untitled3>
File Input Test View Convert Help
Editor
Simulate: [aba]
q0
q1
q2
q3
a ; ☐ , R
a ; a , R
☐ ; ☐ , S
☐ ; a , R
b ; ☐ , R
b ; a , R
a ; b , R
b ; b , R
☐ ; b , R
q1
Step Reset Focus Defocus Freeze Thaw Trace Remove


JFLAP : <untitled3>
File Input Test View Convert Help
Editor
Simulate: [aba]
□;□,S
a;a,R
□;a,R
a;□,R
q0
q1
q2
q3
b;□,R
b;a,R
a;b,R
b;b,R
□;b,R
Step Reset Focus Defocus Freeze Thaw Trace Remove

Beispiel:



Falls die Maschine mit bin(k) für ein $k \in \mathbb{N}_0$ gestartet wird, ändert die Maschine den Bandinhalt zu bin$(k+1)$. Der Schreib-Lesekopf steht anschließend auf dem ersten Symbol von bin$(k+1)$.

Sei $M = (K, \Sigma, \Gamma, \delta, s, q_{\mathsf{accept}}, q_{\mathsf{reject}})$ eine Turing-Maschine.
Ein Tripel

$$(w_1, q, w_2)$$

aus $((\Gamma - \{\sqcup\})\Gamma^* \cup \{\varepsilon\}) \times K \times (\{\varepsilon\} \cup \Gamma^*(\Gamma - \{\sqcup\}))$ heißt *Konfiguration* von M.

Eine Konfiguration beschreibt einen Zustand von M und einen relevanten Teil des Bandes ohne Blanksymbole davor oder dahinter.

Eine Konfiguration (ε, s, w) mit $w \in \Sigma^*$ heißt *Startkonfiguration*.
Eine Konfiguration (w_1, q, w_2) mit $q \in \{q_{\mathsf{accept}}, q_{\mathsf{reject}}\}$ heißt *Haltekonfiguration*.

In Konfiguration (w_1, q, w_2) geschieht Folgendes:

- Falls $q \in \{q_{\text{accept}}, q_{\text{reject}}\}$, so hält M an.
- Falls $w_2 = aw_3$ und $\delta(q, a) = (p, \sigma, D)$, so ersetzt M das Symbol a durch σ und bewegt den Schreib-Lesekopf wie durch D vorgegeben, und geht in Zustand p über.
- Falls $w_2 = \varepsilon$ und $\delta(q, \sqcup) = (p, \sigma, D)$, so ersetzt M das Blanksymbol $\sqcup$ durch σ und bewegt den Schreib-Lesekopf wie durch D vorgegeben, und geht in Zustand p über.

Analog zu den anderen Automaten definieren wir nun eine Relation $\vdash_M$ auf den Konfigurationen von M.

Seien C und C' Konfigurationen von M. Dann gilt $C \vdash_M C'$ genau dann wenn M unter einer Anwendung von δ, d.h., in einem Berechnungsschritt, von C in C' übergeht.

Wie immer bezeichnet $\vdash_M^*$ die reflexive, transitive Hülle von $\vdash_M$.

Eine Folge von Konfigurationen $C_0, C_1, \ldots, C_n$, $n \geq 0$, mit

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots \vdash_M C_n$$

heißt eine *Berechnung* von M der *Länge* n. Wir schreiben $C_0 \vdash_M^n C_n$.

Sei $M = (K, \Sigma, \Gamma, \delta, s, q_{\mathsf{accept}}, q_{\mathsf{reject}})$ eine Turing-Maschine.

M *akzeptiert* $w \in \Sigma^*$ genau dann wenn es $u, v \in \Gamma^*$ gibt, so dass $(\varepsilon, s, w) \vdash_M^* (u, q_{\mathsf{accept}}, v)$.

M *verwirft* $w \in \Sigma^*$ genau dann wenn es $u, v \in \Gamma^*$ gibt, so dass $(\varepsilon, s, w) \vdash_M^* (u, q_{\mathsf{reject}}, v)$.

Die von der Turing-Maschine M *akzeptierte Sprache* ist $L(M) = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$.

Definition:
Eine Sprache L heißt rekursiv aufzählbar (oder auch semi-entscheidbar), falls es eine Turing-Maschine M gibt, so dass $L = L(M)$.

Eine Sprache L ist also rekursiv aufzählbar genau dann wenn es eine Turing-Maschine M gibt, so dass gilt:

$$w \in L \text{ g.d.w. } M \text{ akzeptiert } w$$

Falls $w \notin L$, so kann M das Wort w verwerfen oder nie eine Haltekonfiguration erreichen.

Satz:

Eine Sprache L über einem Alphabet Σ ist rekursiv aufzählbar genau dann wenn es eine Turing-Maschine M mit Startzustand s und einem ausgezeichneten Zustand p gibt, so dass

$$L = \{w \in \Sigma^* \mid (\varepsilon, s, \varepsilon) \vdash_M^* (\varepsilon, p, w)\}$$

Entscheidbare Sprachen

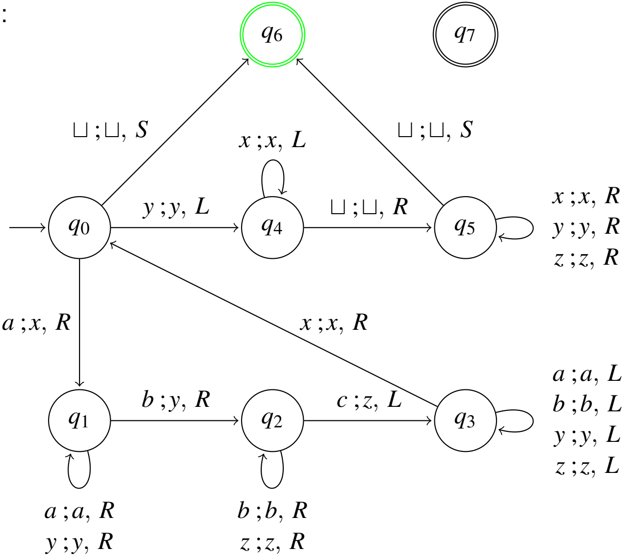
Sei M eine Turing-Maschine mit Eingabealphabet Σ.
M *entscheidet* eine Sprache $L \subseteq \Sigma^*$ genau dann wenn für alle $w \in \Sigma^*$ gilt:

- Falls $w \in L$, so hält M in q_{accept}.
- Falls $w \notin L$, so hält M in q_{reject}.

M hält also bei allen Eingabewörtern w aus Σ^*.

Definition:
Eine Sprache L heißt entscheidbar (oder auch rekursiv), falls es eine Turing-Maschine M gibt, die L entscheidet.

Beispiel:



q_6
q_7
$\sqcup\,;\sqcup,\ S$
$x\,;x,\ L$
$\sqcup\,;\sqcup,\ S$
q_0
$y\,;y,\ L$
q_4
$\sqcup\,;\sqcup,\ R$
q_5
$x\,;x,\ R$
$y\,;y,\ R$
$z\,;z,\ R$
$a\,;x,\ R$
$x\,;x,\ R$
q_1
$b\,;y,\ R$
q_2
$c\,;z,\ L$
q_3
$a\,;a,\ L$
$b\,;b,\ L$
$y\,;y,\ L$
$z\,;z,\ L$
$a\,;a,\ R$
$y\,;y,\ R$
$b\,;b,\ R$
$z\,;z,\ R$

Im Beispiel ist

$$\begin{aligned} K &= \{q_0, q_1, \ldots, q_7\} & s &= q_0 \\ \Sigma &= \{a, b, c\} & q_{\mathsf{accept}} &= q_6 \\ \Gamma &= \{a, b, c, x, y, z, \sqcup\} & q_{\mathsf{reject}} &= q_7 \end{aligned}$$

Zu δ: Im Diagramm fehlen einige Übergänge!

Zwei Varianten zur Komplettierung:

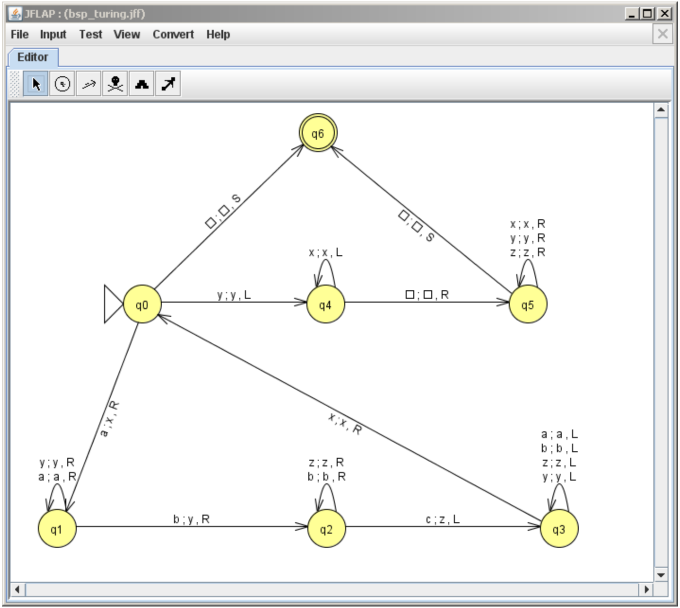
(1) Falls bei $q_i \in K - \{q_{\mathsf{accept}}, q_{\mathsf{reject}}\}$ ein Übergang unter $\sigma \in \Gamma$ fehlt, so gilt $\delta(q_i, \sigma) = (q_i, \sigma, S)$.

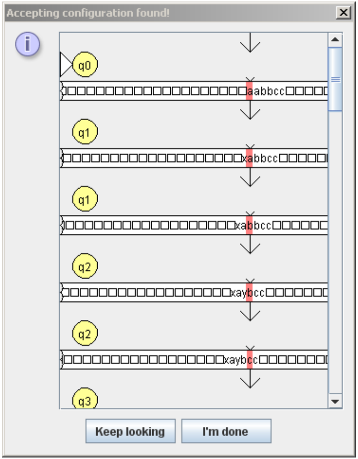
(2) Alle fehlenden Übergänge führen zu $q_7 = q_{\mathsf{reject}}$.

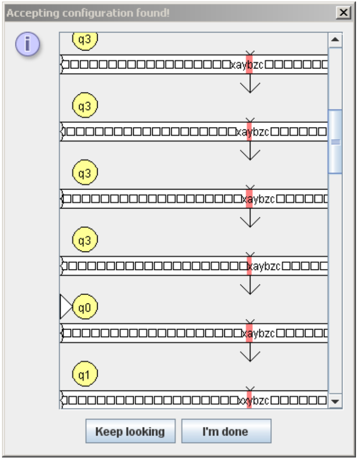
In beiden Varianten ist die von der Turing-Maschine akzeptierte Sprache die nicht kontextfreie Sprache

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

In Variante (2) entscheidet die Turing-Maschine die Sprache L.


JFLAP : (bsp_turing.jff)
File Input Test View Convert Help
Editor
q6
q0
q4
q5
q1
q2
q3
□ ; □ , S
□ ; □ , S
x ; x , R
y ; y , R
z ; z , R
x ; x , L
y ; y , L
□ ; □ , R
a ; x , R
x ; x , R
y ; y , R
a ; a , R
z ; z , R
b ; b , R
a ; a , L
b ; b , L
z ; z , L
y ; y , L
b ; y , R
c ; z , L


Accepting configuration found!
q0
aabbcc
q1
xabbcc
q1
xabbcc
q2
xaybcc
q2
xaybcc
q3
Keep looking
I'm done

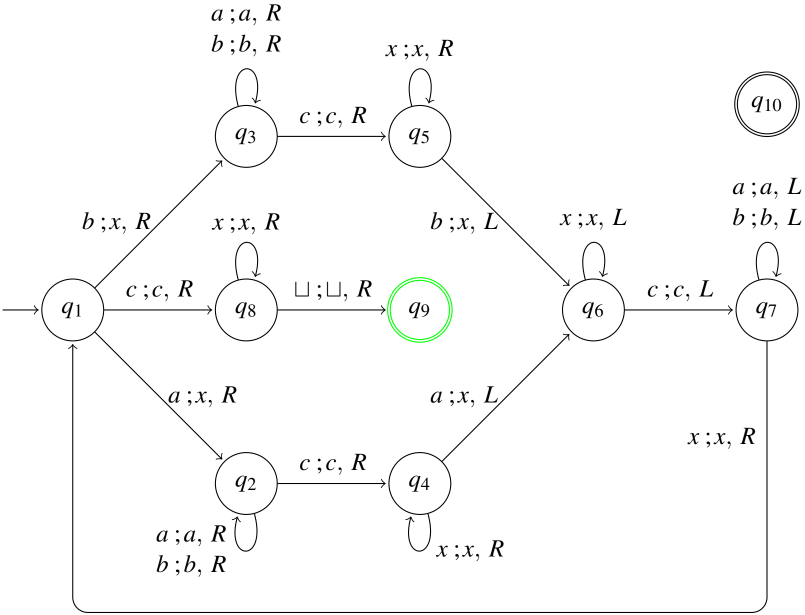

Accepting configuration found!
q3
xaybzc
q3
xaybzc
q3
xaybzc
q3
xaybzc
q0
xaybzc
q1
xxybzc
Keep looking
I'm done

Beispiel:

$$L = \{wcw \mid w \in \{a,b\}^*\}$$

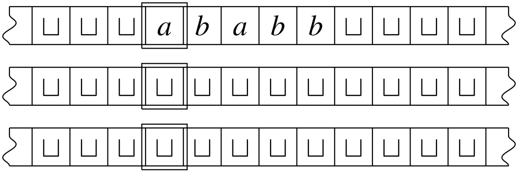
Informelle Beschreibung:
Springe zwischen korrespondierenden Positionen links und rechts von c hin und her und überprüfe, ob die Positionen die gleichen Symbole enthalten. Falls nicht oder falls die Symbole nicht aus $\{a,b\}$ sind, verwirf die Eingabe. Ersetze überprüfte Symbole durch ein x.

Falls alle Symbole links von c durch ein x ersetzt wurden, überprüfe, ob Symbole rechts von c übrig geblieben sind, also nicht durch x ersetzt wurden. Falls ja, verwirf die Eingabe, sonst akzeptiere sie.


a ;a, R
b ;b, R
x ;x, R
q10
c ;c, R
q3
q5
b ;x, R
x ;x, R
b ;x, L
x ;x, L
a ;a, L
b ;b, L
q1
c ;c, R
q8
⊔ ;⊔, R
q9
q6
c ;c, L
q7
a ;x, R
a ;x, L
x ;x, R
q2
c ;c, R
q4
a ;a, R
b ;b, R
x ;x, R

Mehrband-Turing-Maschinen

Eine k-*Band-Turing-Maschine* besitzt k verschiedene durchnummerierte beidseitig unendliche Bänder. Jedes Band besitzt einen eigenen Schreib-Lesekopf. Anfangs steht die Eingabe auf dem ersten Band, alle übrigen Bänder sind leer. Der Schreib-Lesekopf des ersten Bandes steht auf dem ersten Symbol der Eingabe. Beispiel $k = 3$:



Ebenso wie 1-Band-Turing-Maschinen wird eine k-Band-Turing-Maschine durch ein 7-Tupel $(K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ beschrieben.

δ beschreibt die Aktionen auf den k Bändern abhängig vom aktuellen Zustand und den Symbolen unter den k Schreib-Leseköpfen, ist also eine Funktion von $(K - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma^k$ nach $K \times (\Gamma \times \{L, R, S\})^k$.

Sei $M = (K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ eine k-Band-Turing-Maschine. Ein $(2k+1)$-Tupel

$$(q, w_{11}, w_{12}, w_{21}, w_{22}, \ldots, w_{k1}, w_{k2})$$

aus $K \times (((\Gamma - \{\sqcup\})\Gamma^* \cup \{\varepsilon\}) \times (\{\varepsilon\} \cup \Gamma^*(\Gamma - \{\sqcup\})))^k$ heißt *Konfiguration* von M.

Eine Konfiguration beschreibt einen Zustand von M und relevante Teile der k Bänder ohne Blanksymbole davor oder dahinter. Der j-te Schreib-Lesekopf steht auf dem ersten Symbol von w_{j2}.

Analog zu Turing-Maschinen mit nur einem Band definieren wir eine Relation $\vdash_M$ auf den Konfigurationen von M.

M *akzeptiert* $w \in \Sigma^*$ genau dann wenn es $u_1, v_1, \ldots, u_k, v_k \in \Gamma^*$ gibt, so dass $(s, \varepsilon, w, \varepsilon, \ldots, \varepsilon) \vdash_M^* (q_{\mathsf{accept}}, u_1, v_1, \ldots, u_k, v_k)$.

M *verwirft* $w \in \Sigma^*$ genau dann wenn es $u_1, v_1, \ldots, u_k, v_k \in \Gamma^*$ gibt, so dass $(s, \varepsilon, w, \varepsilon, \ldots, \varepsilon) \vdash_M^* (q_{\mathsf{reject}}, u_1, v_1, \ldots, u_k, v_k)$.

Die von der Turing-Maschine M *akzeptierte Sprache* ist $L(M) = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$.

Die k-Band-Turing-Maschine M *entscheidet* eine Sprache $L \subseteq \Sigma^*$ genau dann wenn für alle $w \in \Sigma^*$ gilt:

- Falls $w \in L$, so hält M in q_{accept}.
- Falls $w \notin L$, so hält M in q_{reject}.

k-Band-Turing-Maschinen sind nicht mächtiger als Standard-Turing-Maschinen mit nur einem Band in folgendem Sinn:

Satz:
Die Klasse der von k-Band-Turing-Maschinen akzeptierten Sprachen stimmt genau mit der Klasse der von Standard-Turing-Maschinen mit nur einem Band akzeptierten Sprachen überein.

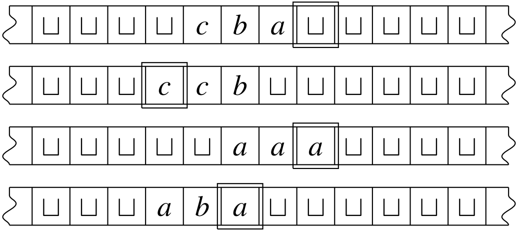
Satz:
Die Klasse der Sprachen, die von k-Band-Turing-Maschinen entschieden werden, ist genau die Klasse der Sprachen, die von Standard-Turing-Maschinen mit nur einem Band entschieden werden.

Beweisskizze:
Wir simulieren Berechnungen von k-Band-Turing-Maschinen mit geeigneten Standard-Turing-Maschinen.

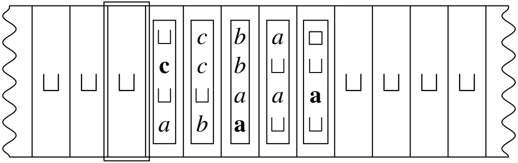
Sei $M = (K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ eine k-Band-Turing-Maschine. Wir simulieren auf einer Standard-Turing-Maschine k virtuelle Spuren, indem wir spezielle Symbole zum Bandalphabet hinzunehmen. Diese Symbole kodieren einen Stapel von k Symbolen aus Γ bzw. markierten Symbolen aus Γ. Die Markierungen geben die Positionen virtueller Schreib-Leseköpfe auf den virtuellen Spuren an. Wir erhalten so $(2|\Gamma|)^k$ neue zusätzliche Symbole.

Die Zustandsmenge der simulierenden Standard-Turing-Maschine umfasst K.

Simulierte 4-Band-Turing-Maschine:

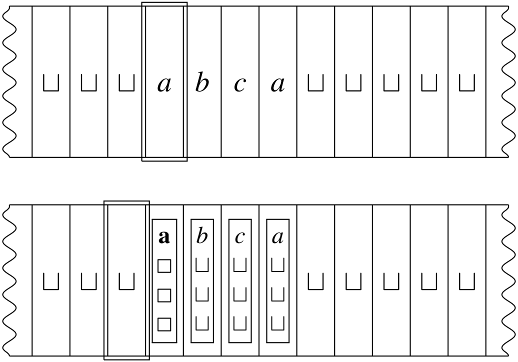


Simulierende Standard-Turing-Maschine:



Die simulierende Standard-Turing-Maschine arbeitet in Simulationsphasen.

Zunächst wird die Eingabe umgebaut:



Umbau der Eingabe: Symbole aus dem Eingabealphabet werden durch neue Symbole ersetzt: Steht auf dem Band $\sigma \in \Sigma$, so wird σ durch das neue Bandalphabetsymbol ersetzt, das folgende Bedeutung hat: Auf dem ersten Band der zu simulierenden Maschine steht an entsprechender Stelle σ, auf den übrigen Bändern an entsprechender Stelle $\sqcup$. Dabei wird berücksichtigt, dass der Schreib-Lesekopf auf dem ersten Symbol des Eingabewortes des ersten Bandes steht (und die Schreib-Leseköpfe der übrigen Bänder auf Blanksymbolen an entsprechender Position).

Der Schreib-Lesekopf der simulierenden Maschine steht danach auf dem Blanksymbol links von der umgebauten Eingabe.

Nun wird Schritt für Schritt simuliert:

(a) Aufsammeln der Symbole unter den virtuellen Schreib-Leseköpfe und Speichern dieser Symbole im Zustand.

(b) Eigentliche Zustandsänderung abhängig von den gelesenen Symbolen und dem vorherigen Zustand. Der neue Zustand kodiert die auszuführenden Schreib- und Kopfbewegeaktionen und den Zustand (der simulierten Maschine), der nach Ausführung dieser Aktionen erreicht wird.

(c) Ausführen der Schreib- und Kopfbewegeaktionen, die im Zustand gespeichert sind, auf den virtuellen Spuren. Dabei eventuell Ersetzen von $\sqcup$ durch ein geeignetes neues Symbol, das (evtl. markierten) Blanksymbolen auf k Spuren entspricht.

Nach der Simulation eines Schrittes der k-Band-Turing-Maschine steht der Schreib-Lesekopf der simulierenden Standard-Turing-Maschine stets auf dem Blanksymbol $\sqcup$, das unmittelbar links von dem Teil des Bandes liegt, das die neuen Symbole enthält.

Die simulierende Standard-Turing-Maschine erreicht q_{accept} bzw. q_{reject} bei Eingabe w genau dann wenn die simulierte k-Band-Turing-Maschine bei Eingabe w Zustand q_{accept} bzw. q_{reject} erreicht.

Die Simulation von Standard-Turing-Maschinen durch k-Band-Turing-Maschinen ist trivial. $\blacksquare$

Nichtdeterministische Turing-Maschinen

Definition:
Eine nichtdeterministische Turing-Maschine ist ein 7-Tupel $(K, \Sigma, \Gamma, \Delta, s, q_{\text{accept}}, q_{\text{reject}})$, *wobei gilt:*

- K *ist eine endliche Menge von Zuständen,*
- Σ *ist das Eingabealphabet,* $\sqcup \notin \Sigma$,
- Γ *ist das Bandalphabet,* $\Gamma \supseteq \Sigma$ *und* $\sqcup \in \Gamma$,
- $s \in K$ *ist der Startzustand,*
- $q_{\text{accept}} \in K$ *ist der akzeptierende Halteszustand,*
- $q_{\text{reject}} \in K$ *ist der verwerfende Halteszustand,* $q_{\text{reject}} \neq q_{\text{accept}}$,
- $\Delta \subseteq (K - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \times K \times \Gamma \times \{L, R, S\}$
  ist die Übergangsrelation.

Wie bei deterministischen Turing-Maschinen steht der Schreib-Lesekopf zu Beginn auf dem ersten Symbol der Eingabe bzw. auf dem Blanksymbol, falls die Eingabe das leere Wort ist.

Anders als deterministische Turing-Maschinen kann eine nichtdeterministische Turing-Maschine hängen bleiben, weil es keinen Übergang unter dem aktuellen Zustand und dem aktuellen Symbol unter dem Schreib-Lesekopf gibt.

Sei $M = (K, \Sigma, \Gamma, \Delta, s, q_{\mathsf{accept}}, q_{\mathsf{reject}})$ eine nichtdeterministische Turing-Maschine.

$(w_1, q, w_2) \in ((\Gamma - \{\sqcup\})\Gamma^* \cup \{\varepsilon\}) \times K \times (\{\varepsilon\} \cup \Gamma^*(\Gamma - \{\sqcup\}))$ nennen wir wie zuvor eine *Konfiguration* von M.

Analog zu deterministischen Turing-Maschinen definieren wir eine Relation $\vdash_M$ auf den Konfigurationen von M.

M *akzeptiert* $w \in \Sigma^*$ genau dann wenn es $u, v \in \Gamma^*$ gibt, so dass $(\varepsilon, s, w) \vdash_M^* (u, q_{\mathsf{accept}}, v)$.

Die von einer nichtdeterministischen Turing-Maschine M *akzeptierte Sprache* ist $L(M) = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$.

Eine nichtdeterministische Turing-Maschine M *entscheidet* eine Sprache $L \subseteq \Sigma^*$, falls M bei allen Eingaben $w \in \Sigma^*$ immer hält und L die von M akzeptierte Sprache ist.

Anmerkungen:

(1) Es gibt also ein $N \in \mathbb{N}$, das nur von M und w abhängt, so dass es keine Konfiguration C gibt mit $(\varepsilon, s, w) \vdash_M^N C$.

(2) Falls $w \notin L$, so erreicht M also stets q_{reject} nach endlich vielen Schritten.

Nichtdeterministische Turing-Maschinen sind nicht mächtiger als deterministische Turing-Maschinen in folgendem Sinn:

Satz:
Die Klasse der von nichtdeterministischen Turing-Maschinen akzeptierten Sprachen stimmt genau mit der Klasse der von deterministischen Turing-Maschinen akzeptierten Sprachen überein.

Satz:
Die Klasse der Sprachen, die von nichtdeterministischen Turing-Maschinen entschieden werden, ist genau die Klasse der Sprachen, die von deterministischen Turing-Maschinen entschieden werden.

Beweisskizze:
Wir zeigen, falls eine nichtdeterministische Turing-Maschine M eine Sprache akzeptiert bzw. entscheidet, so gibt es auch eine deterministische Turing-Maschine M', die die Sprache akzeptiert bzw. entscheidet.

Die Beweisidee ist, alle möglichen Berechnungen durchzuprobieren.

Sei $M = (K, \Sigma, \Gamma, \Delta, s, q_{\mathsf{accept}}, q_{\mathsf{reject}})$ eine nichtdeterministische Turing-Maschine. Dann existiert $m \in \mathbb{N}$, so dass für alle Paare (q, a) aus $K \times \Gamma$ gilt $|\{(p, b, D) \mid (q, a, p, b, D) \in \Delta\}| \leq m$.

Falls es für $(q,a) \in (K - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma$ keinen Übergang geben sollte, fügen wir (q,a,q,a,S) zu Δ hinzu. Das ändert am Akzeptanzverhalten von M nichts!

Sei nun Σ_1 Menge von m Symbolen.
Für jedes Paar $(q,a) \in (K - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma$ ordnen wir jedem $\sigma \in \Sigma_1$ einen Übergang aus $U_{q,a} = \{(p,b,D) \mid (q,a,p,b,D) \in \Delta\}$ zu, so dass jedes Element aus $U_{q,a}$ mindestens einem Symbol zugeordnet ist.

Nun kodiert jedes Wort aus Σ_1^d eine Berechung bestehend aus d Schritten.

Wir probieren mit einer 3-Band Turing-Maschine alle Berechnungen bestehend aus d Schritten für wachsendes d aus:

Band 1: Eingabe w
Band 2: bearbeitete Kopie von w
Band 3: $\sigma_1\sigma_2\ldots\sigma_d \in \Sigma_1^*$.

Das Symbol auf der j-ten Position von Band 3 bestimmt, welche der m möglichen Aktionen auf Band 2 ausgeführt wird. Wird auf Band 3 ein $\sqcup$ erreicht, so wird die aktuelle deterministische Simulation gestoppt, ein neues Wort auf Band 3 generiert und die zum neu erzeugten Wort gehörige Simulation gestartet. Auf Band 3 werden der Länge nach alle Wörter aus Σ_1^* generiert. $\blacksquare$

Grammatiken und Turing-Maschinen

Beispiel: Sei $G = (V, \Sigma, R, S)$ mit $V = \{S, A, D, \rhd, \heartsuit\}$, $\Sigma = \{a\}$ und

$$
\begin{aligned}
R = \{ \quad S &\rightarrow \rhd A \heartsuit, \\
\rhd &\rightarrow \rhd D, \\
D\heartsuit &\rightarrow \heartsuit, \\
DA &\rightarrow AAD, \\
\rhd &\rightarrow \varepsilon, \\
\heartsuit &\rightarrow \varepsilon, \\
A &\rightarrow a \quad \}
\end{aligned}
$$

$$L(G) = \{a^{2^n} \mid n \geq 0\}$$