Durchschnittsleerheitsproblem (DLP)

- Gegeben: L, L'.
- Es ist zu entscheiden, ob $L \cap L' = \emptyset$

Komplement von DLP (DNLP)

- Gegeben: L, L'.
- Es ist zu entscheiden, ob $L \cap L' \neq \emptyset$

Nichtentscheidbarkeit von DLP

Theorem
DNLP nicht entscheidbar $\Longrightarrow$ DLP nicht entscheidbar.
(Verallgemeinerbar auf beliebige Entscheidbarkeitsprobleme)

Theorem
DLP ist ist für kontextfreie Sprachen nicht entscheidbar.
Beweisidee: Reduktion des Postschen Korrepondenzproblems auf DNLP

Postsches Korrespondenzproblem (PCP)

- $PCP = ((u_1, \ldots, u_n), (v_1, \ldots, u_n))$ mit $u_i, v_i \in T^*$
- PCP lösbar, wenn $i_1 \cdots i_k$ mit $k \geq 1$ existiert, so dass

$$u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}.$$

Theorem

Lösbarkeit des Postschen Korrespondenzproblem ist nicht entscheidbar (für $\emptyset \neq T \neq \{a\}$).

Typ	Bezeichnung	Automaten	Durchschnittsleerheitsproblem
0	allgemein	Turing-Maschinen	-
1	kontext-sensitiv, monoton	linear beschränkte Automaten	-
2	kontextfrei	Kellerautomaten	-
3	regulär, rechtslinear	endliche Automaten	+

Entscheidbarkeit

Formulierung von Entscheidungsproblemen als Sprachen

$$dp\colon A^* \to BOOL \quad \rightsquigarrow \quad L_{dp} = \{w \in A^* \mid dp(w) = T\}$$

Beispiel

$$L_{WP(G)} = L(G)$$

G: Chomsky-Grammatik.

Formulierung von Sprachen als Entscheidungsprobleme

$$L \subseteq A^* \quad \rightsquigarrow \quad dp_L \colon A^* \to BOOL \text{ mit}$$

$$dp_L(w) = \begin{cases} T, & \text{falls } w \in L \\ F & \text{sonst} \end{cases}$$

Bemerkung: dp_L ist das Wortproblem von L.

Beispiel

$$dp_{L(G)} = WP(G)$$

G: Chomsky-Grammatik .

Entscheidbare Sprache

Eine Sprache $L \subseteq A^*$ ist entscheidbar, falls dp_L berechenbar ist (z.B. durch eine CE-S-Spezifikation, eine Turing-Maschine oder ein WHILE-Programm).

Beispiel: $L(G)$, wobei $G = (N, T, P, S)$ eine monotone Grammatik ist.

Semi-entscheidbare Sprache

Eine Sprache $L \subseteq A^*$ ist semi-entscheidbar, falls die partielle Funktion $dp'_L\colon A^* \to BOOL$ mit

$$dp'_L(w) = \begin{cases} T, & \text{falls } w \in L \\ \text{undefiniert} & \text{sonst} \end{cases}$$

berechenbar ist.

Beispiel:

$$L_{PCP} = \{\ ((u_1, \ldots, u_n), (v_1, \ldots, v_n)) \mid u_i, v_i \in T^*, \\ \exists i_1, \ldots, i_k\ (k \geq 1) : u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}\}$$

Entscheidbare und semi-entscheidbare Probleme

Ein Problem dp ist entscheidbar, falls es berechenbar ist.

Beispiel: Wortproblem für monotone Grammatiken

Ein Problem dp ist semi-entscheidbar, falls L_{dp} semi-entscheidbar ist.

Beispiel: PCP

Komplement einer Sprache

Das Komplement einer Sprache $L \subseteq A^*$ ist definiert durch

$$\bar{L} = A^* \setminus L.$$

Theorem

1. L entscheidbar $\Longrightarrow$ $\bar{L}$ entscheidbar.
2. L und $\bar{L}$ semi-entscheidbar $\Longrightarrow$ L entscheidbar.

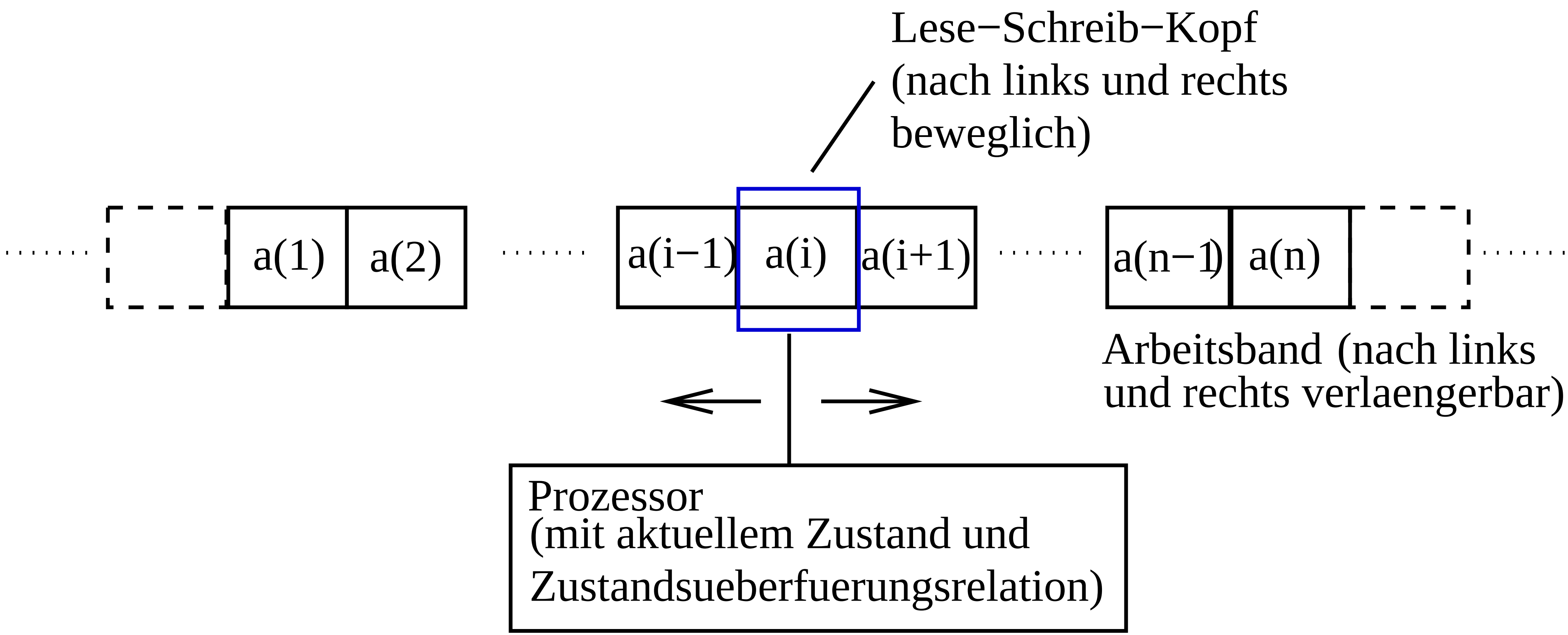
Turing-Maschine

- Von Alan Turing in den 30er Jahren dieses Jahrhunderts eingeführt
- Eines der ältesten Berechenbarkeitsmodelle
  Idee: den mechanischen Anteil des Rechnens mit Bleistift und Radiergummi auf Papier formal fassen.
- Grundlage für zahlreiche Beweise in der Berechenbarkeits- und Komplexitätstheorie

Bestandteile

- Prozessor (mit aktuellem Zustand und Zustandsüberführungrelation)
- Arbeitsband (nach links und rechts verlängerbar)
- Leseschreibkopf (nach links und rechts beweglich)

Graphische Veranschaulichung

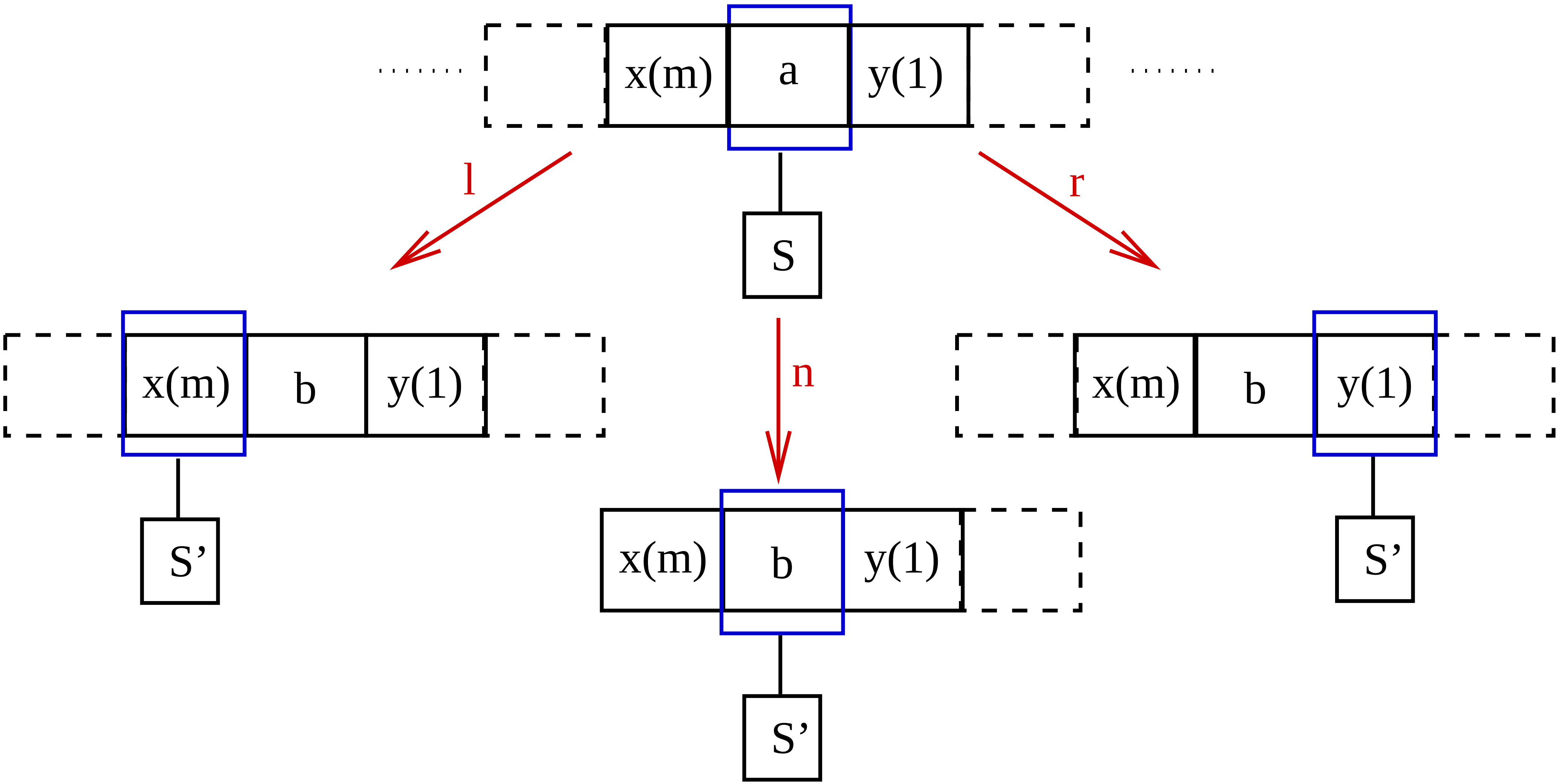


Arbeitsweise

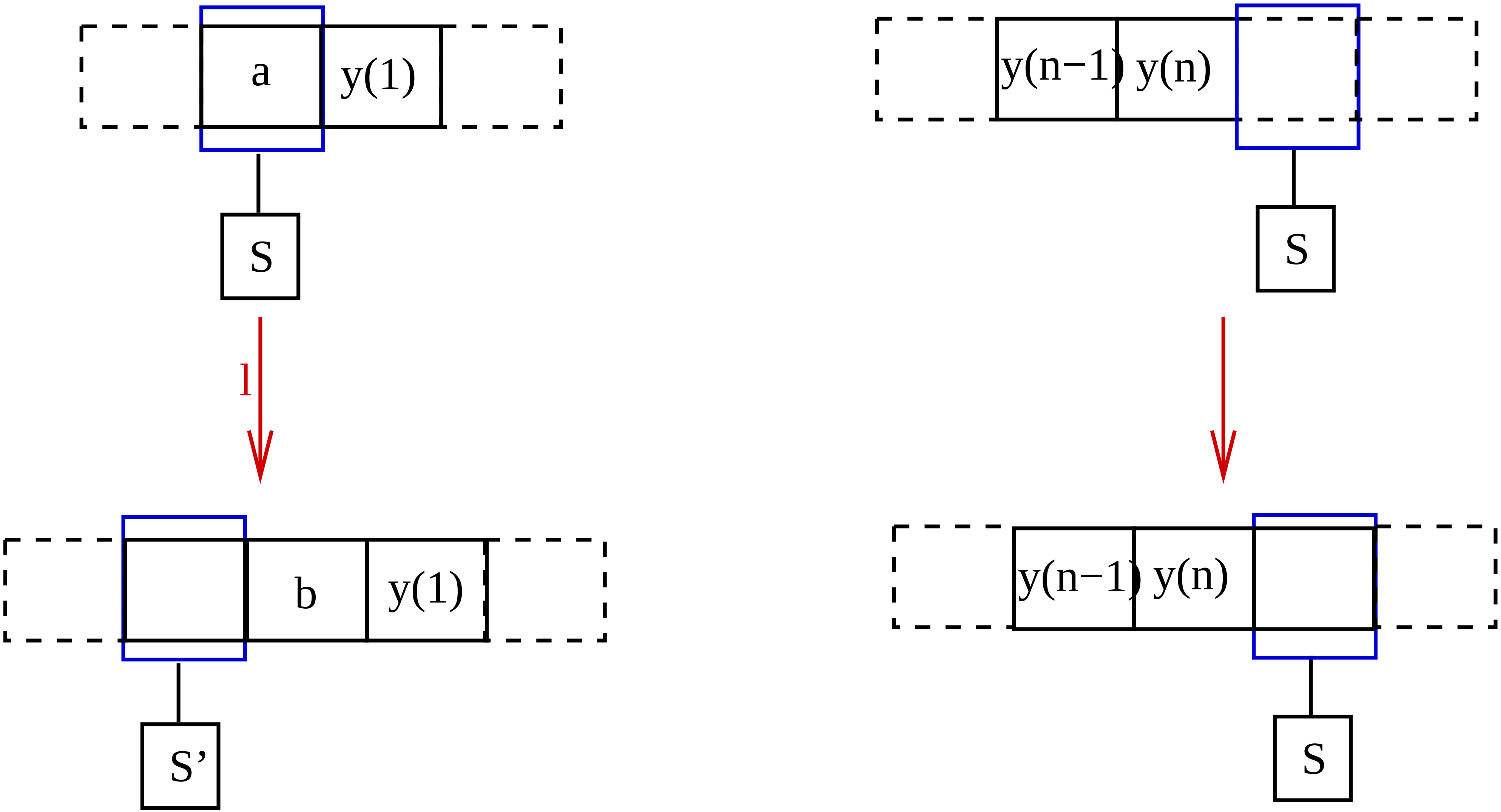
Lesen von a im aktuellen Zustand s bewirkt

- Schreiben von b,
- neuen Zustand s' und
- Kopfbewegung laut Zustandsüberführung.

Graphische Veranschaulichung (1)



Graphische Veranschaulichung (2)



Turing-Maschine: Formale Definition

Turing-Maschine $TM = (S, A, d, s_0, F)$ mit

- S: endliche Menge von Zuständen,
- A: endliches Alphabet,
- d: Zustandsüberführungsrelation mit $d(s, a) \subseteq S \times (A \cup \{\square\}) \times \{l, r, n\}$ für alle $s \in S$, $a \in A \cup \{\square\}$
- $s_0 \in S$: Anfangszustand
- $F \subseteq S$: Menge von Endzuständen

$\square$ steht für leeres Feld ($\square \notin A$)

Konfigurationen

- Sei $A' = A \cup \{\Box\}$

$\boxed{\text{Konfiguration: } usv \text{ mit } s \in S,\ u, v \in A'^*}$

$\boxed{\text{Anfangskonfiguration: } \lambda s_0 w \text{ mit } w \in A^*}$

$\boxed{\text{Endkonfiguration: } us'v \text{ mit } s' \in F,\ u, v \in A'^*}$

Folgekonfiguration

Folgekonfiguration

Für alle $s, s' \in S$, $u, v \in A'^*$ und $a, b, c \in A'$:

$$usav \longmapsto us'bv, \qquad \text{falls } (s', b, n) \in d(s, a)$$

$$usav \longmapsto ubs'v, \qquad \text{falls } (s', b, r) \in d(s, a)$$

$$\left.\begin{array}{l} ucsav \longmapsto us'cbv \\ \lambda sav \longmapsto s'\Box bv \end{array}\right\} \quad \text{falls } (s', b, l) \in d(s, a)$$

$$us\lambda \longmapsto us\Box$$

Konfigurationsfolge

$$con = con_0 \vdash con_1 \vdash \cdots \vdash con_n = con'$$

Dafür kurz: $con \overset{n}{\vdash} con'$ oder $con \overset{*}{\vdash} con'$

Erkannte Sprache

Sei $TM = (S, A, B, d, s_0, F)$ eine Turing-Maschine.
Dann ist $L(TM)$ die von TM erkannte Sprache mit

$$L(TM) = \{w \in A^* \mid s_0 w \vdash^* usv, u, v \in A'^*, s \in F\}$$