Dreifach logarithmischen Platzbedarf gibt es nicht

Sommerakademie Rot an der Rot — AG 1
Wieviel Platz brauchen Algorithmen wirklich?

Tobias Zech

Physikalisches Institut
Uni Freiburg

10. August 2010

Ausblick

1. Beispiele
2. Werkzeuge
   - Turingmaschinen für L
   - Pumpen
   - Endlichkeit $\Rightarrow$ Periodizität
3. GAP-Theorem
   - Einweg-DTM $\geq \log\ n$
   - Zweiwege-DTM $\geq \log \log n$

Akzeptiert in L

L=DSPACE[log n]

- Zähler

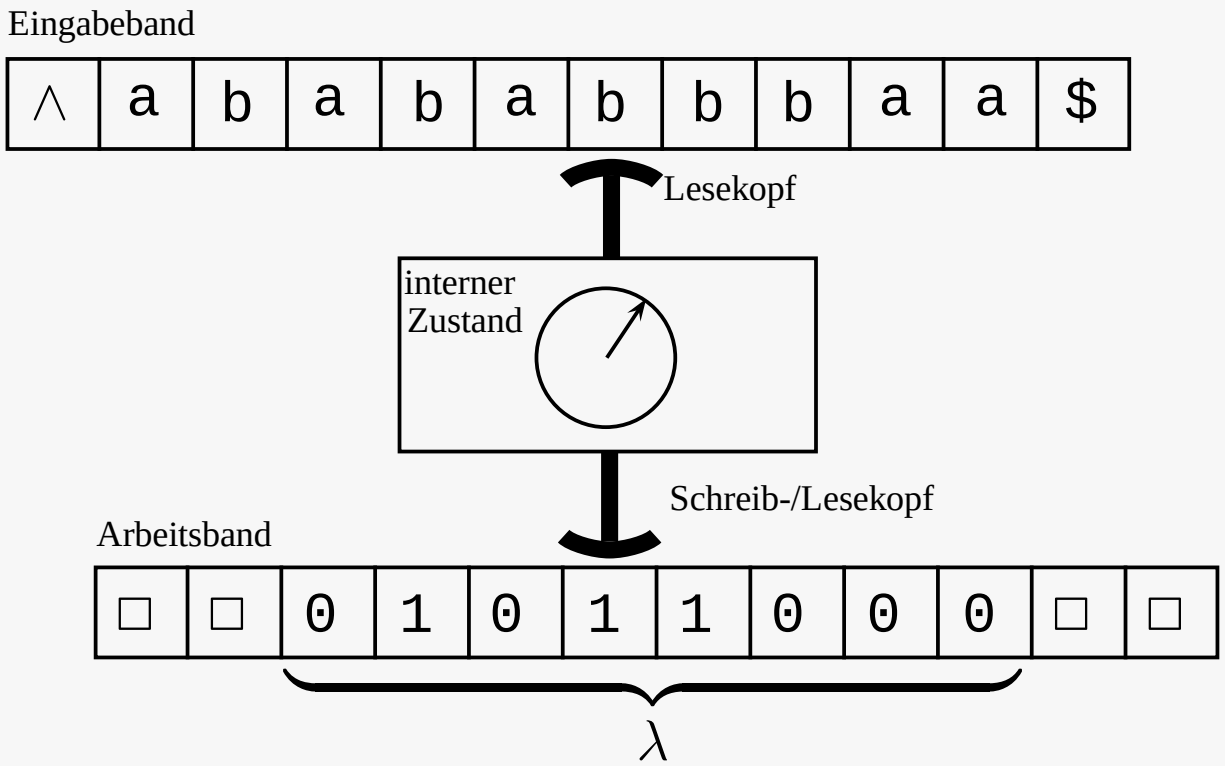
Waren genausoviel Romanistinnen wie Physiker auf der Physiker-Romanisten-Party?
$\{0^n1^n | n \geq 1\}$

DSPACE[log log n]

- kein Zähler

Hat jemand gezählt, ob mehr Physiker als Romanistinnen da waren?
$\{\text{bin}(1); \text{bin}(2); \ldots; \text{bin}(n) | n \geq 1\}$

Turingmaschinen für L=DSPACE[log n]



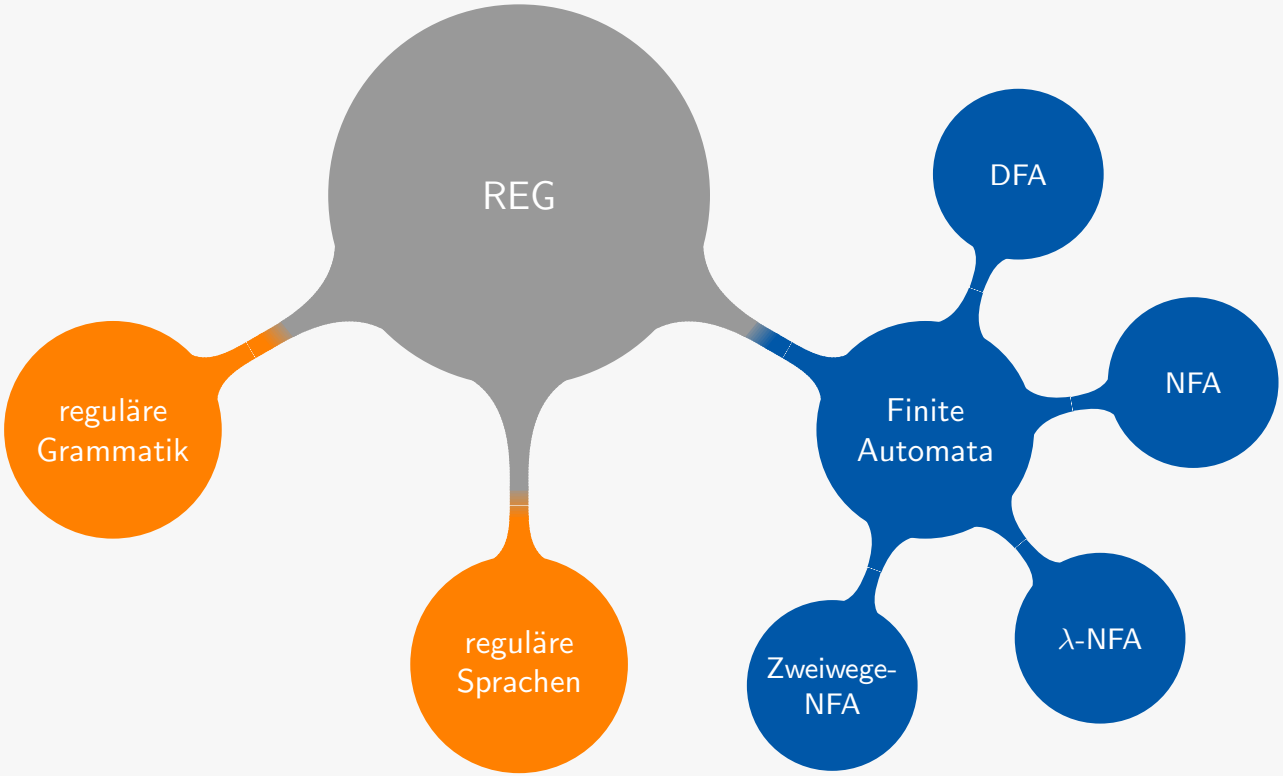
Bänder

- Eingabe: Read-only
- Verarbeitung: unbeschrieben

. . . ,weil . . .

1 Band $\Rightarrow$ DSPACE[n]

Wiederholung: REG = SPACE(0)



Pumping

Zurück zu endlichen Automaten DFA
Länge des Arbeitsbands = konstant/null

Pumping-Lemma

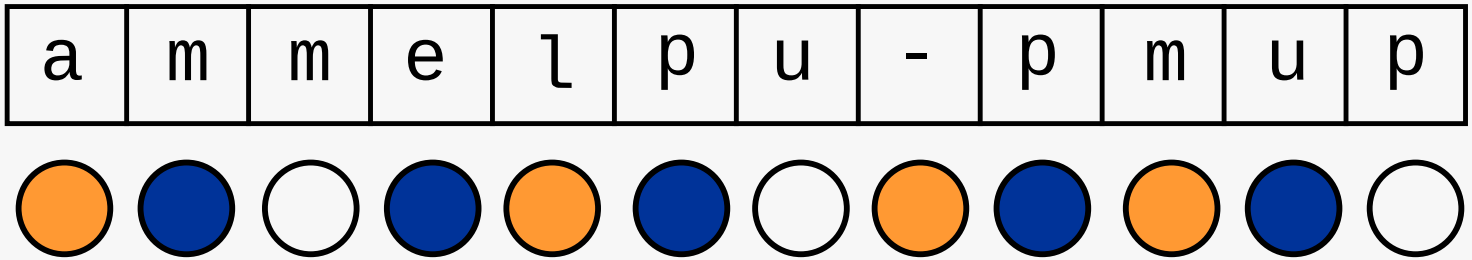
L=L(DFA)

- ab Wortlänge $n = |w|$
- existiert Zerlegung: $w = x\ y\ z$ für alle Worte
- sodass aufgepumptes Wort: $x\ y^i\ z \in L$ für alle i

folgt aus: endlich vielen *internen* Zustände

Aufpumpen/Abpumpen

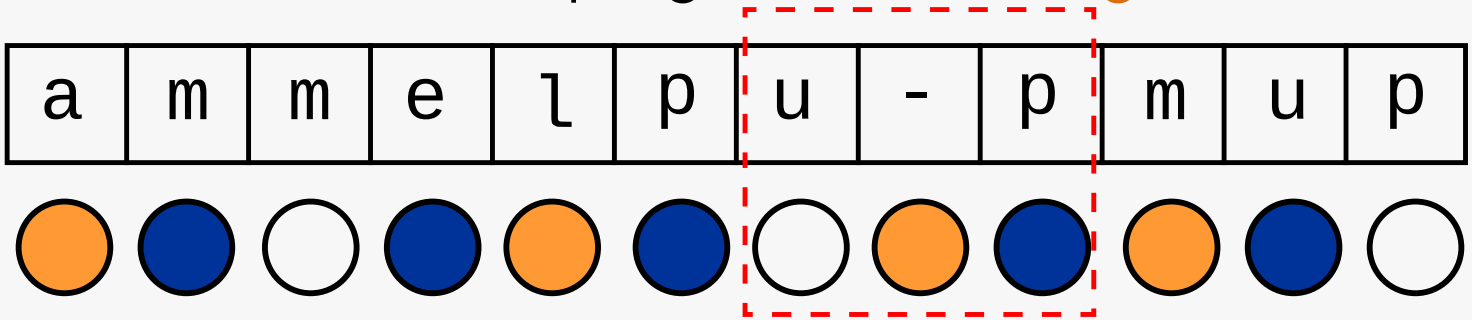
Auf der Suche nach dem Abfluss: Dem kürzestem Wort



Aufpumpen/Abpumpen

Auf der Suche nach dem Abfluss: Dem kürzestem Wort

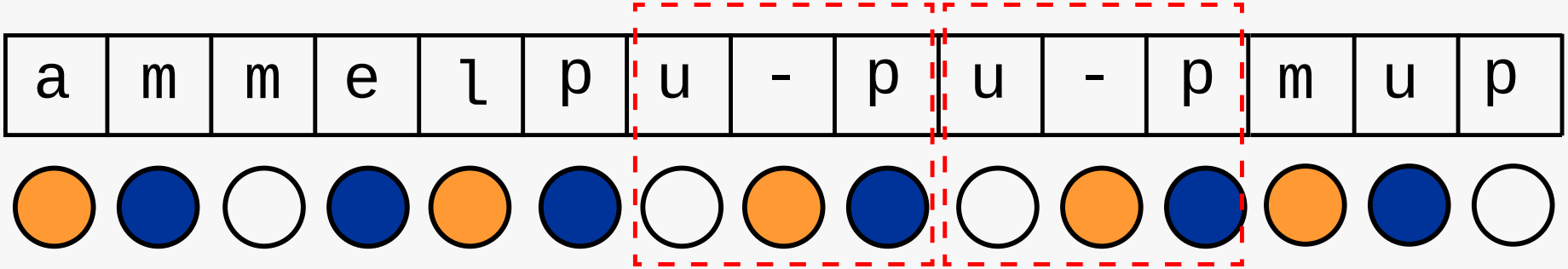
Ausnutzen der Pump-Eigenschaft: Verlängern



Aufpumpen/Abpumpen

Auf der Suche nach dem Abfluss: Dem kürzestem Wort

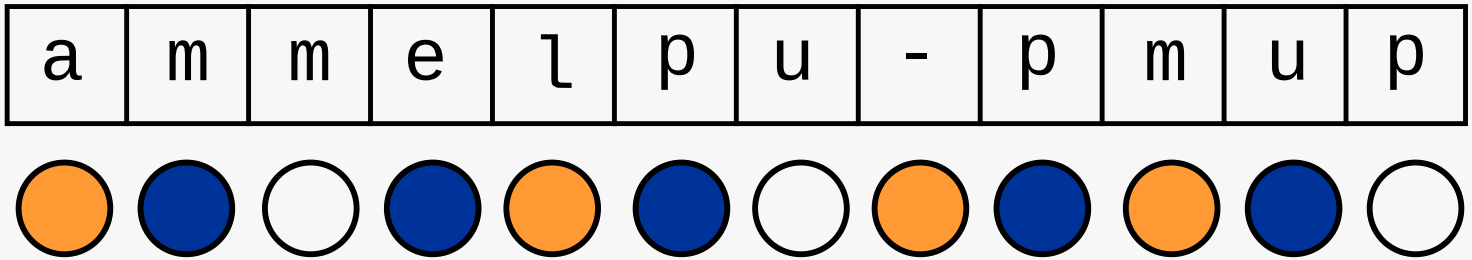
Ausnutzen der Pump-Eigenschaft: Verlängern



Aufpumpen/Abpumpen

Auf der Suche nach dem Abfluss: Dem kürzestem Wort

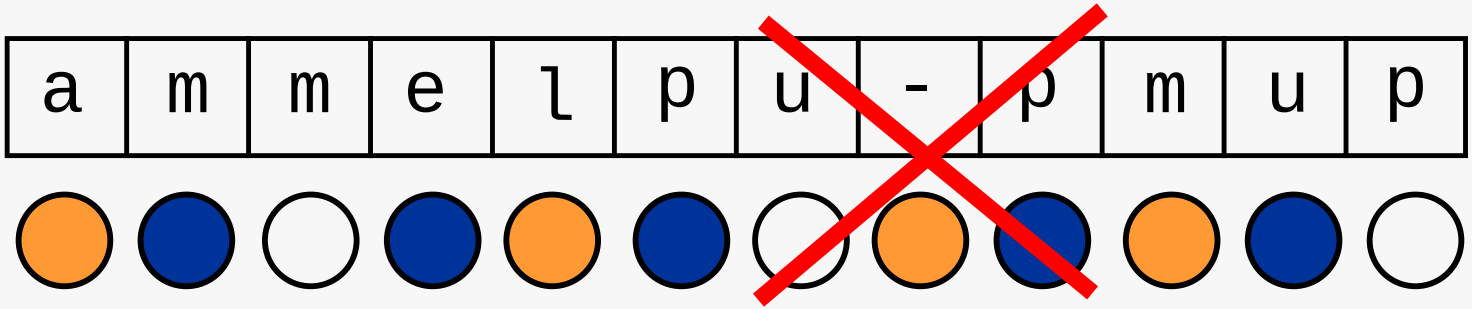
Abpumpen: Verkürzen



Aufpumpen/Abpumpen

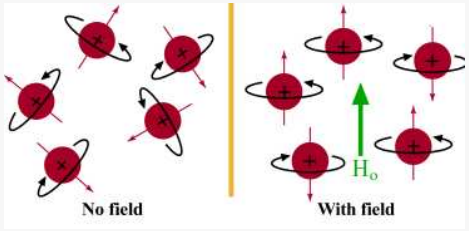
Auf der Suche nach dem Abfluss: Dem kürzestem Wort

Abpumpen: Verkürzen



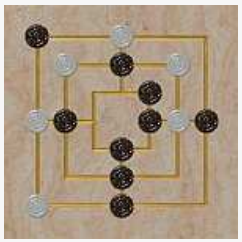
Endlichkeit ⇒ Periodizität

Kernspin



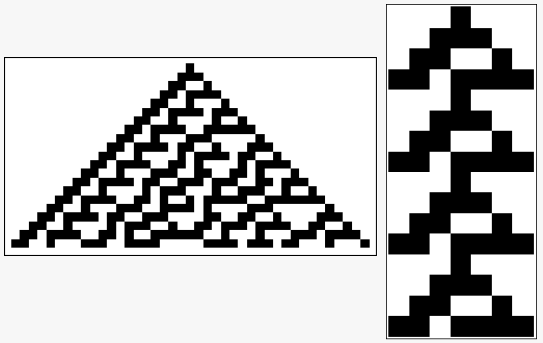
- Spin up↑/down↓

Mühle



- Ecke •/ ∘ /+

Zelluläre Automaten

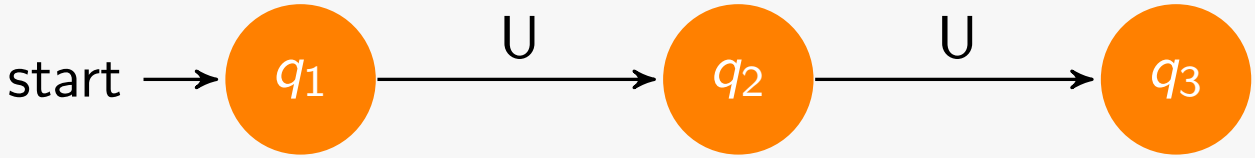


- endliches Grid

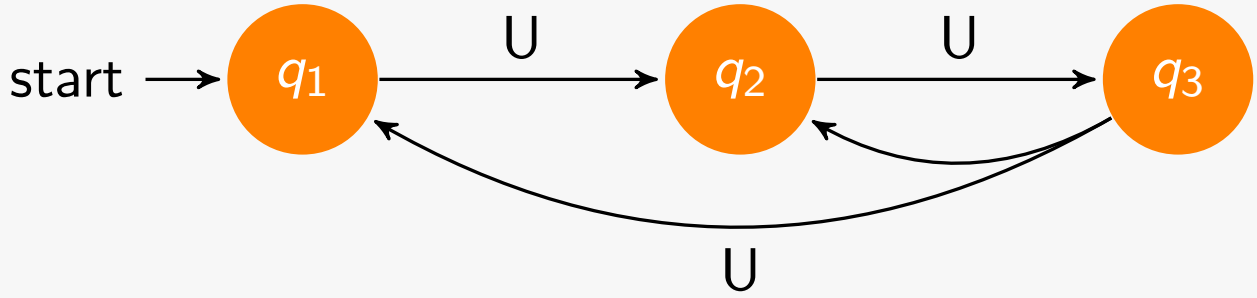
endlich viele *System*-Zuständen ∧ deterministischer Zustandsübergang

⇒ Periodizität

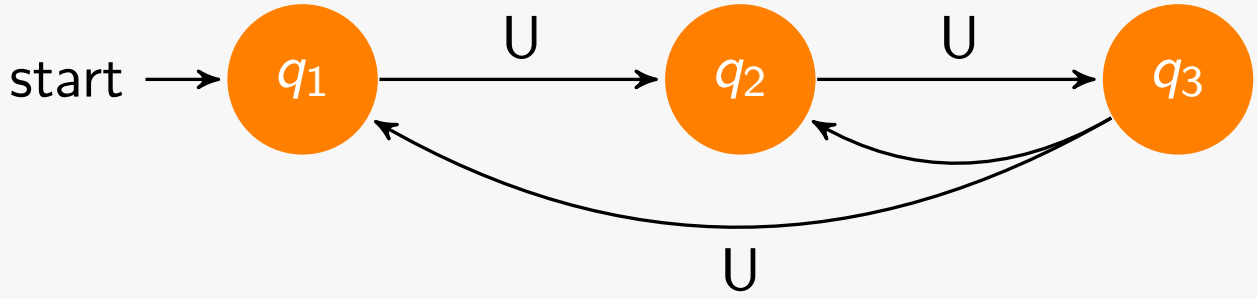
Periodendauer



Periodendauer

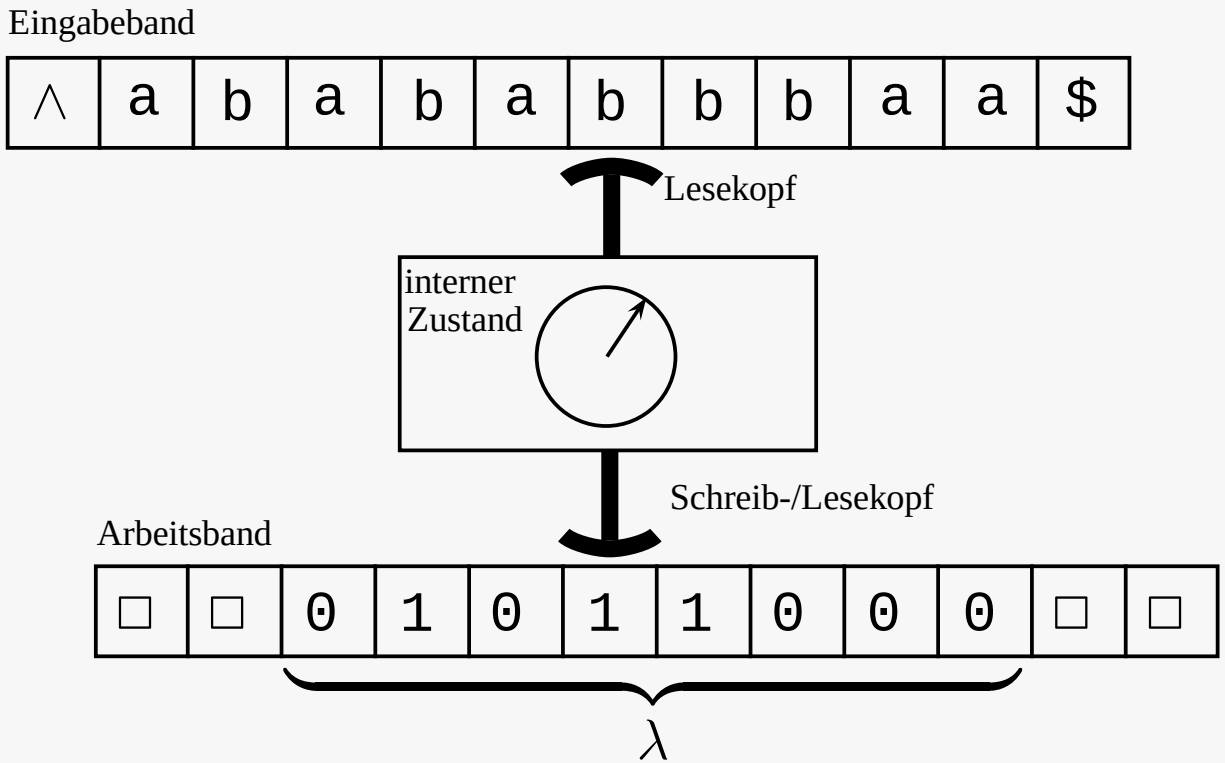


Periodendauer



Anzahl der *System*-Zustände beschränkt die maximale Periode

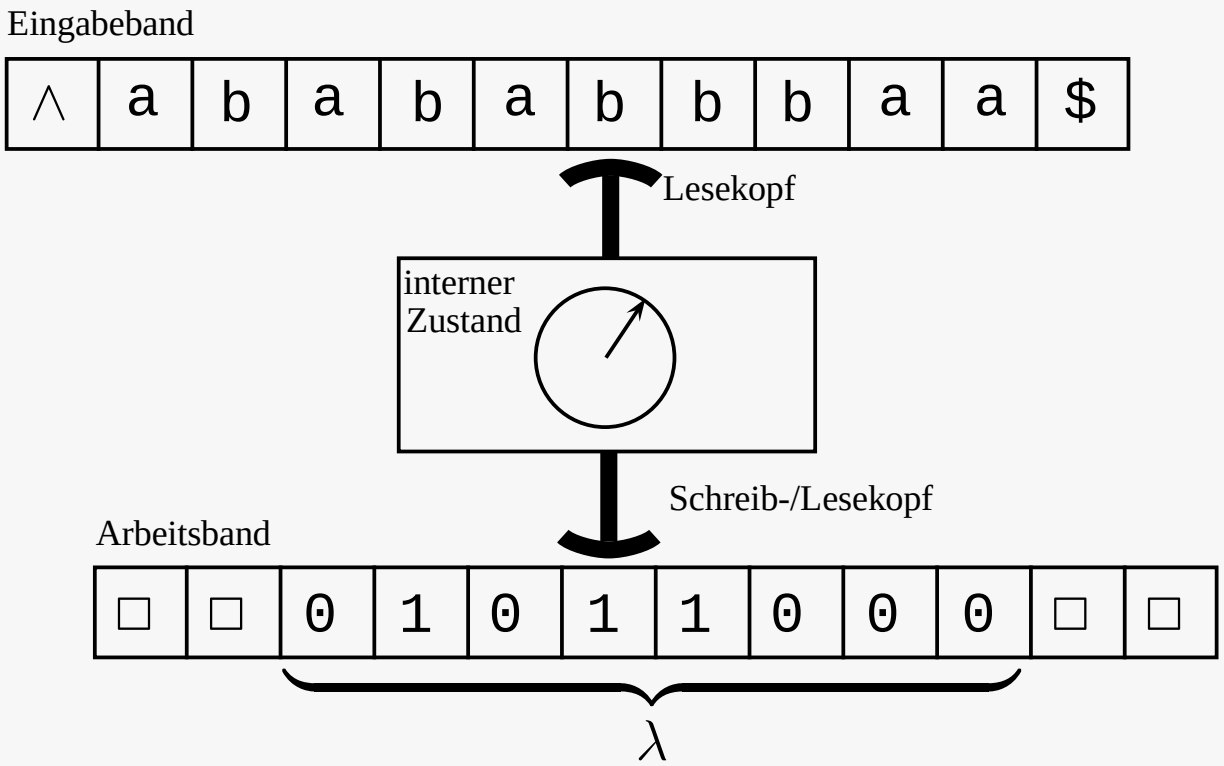
DTM mit Einweg-Eingabeband



Lesekopf des Eingabebandes = Einbahnband: Nach rechts (/ Warten)

- Lesereihnfolge = Zeichenfolge
- Kein Symbol wird doppelt gelesen

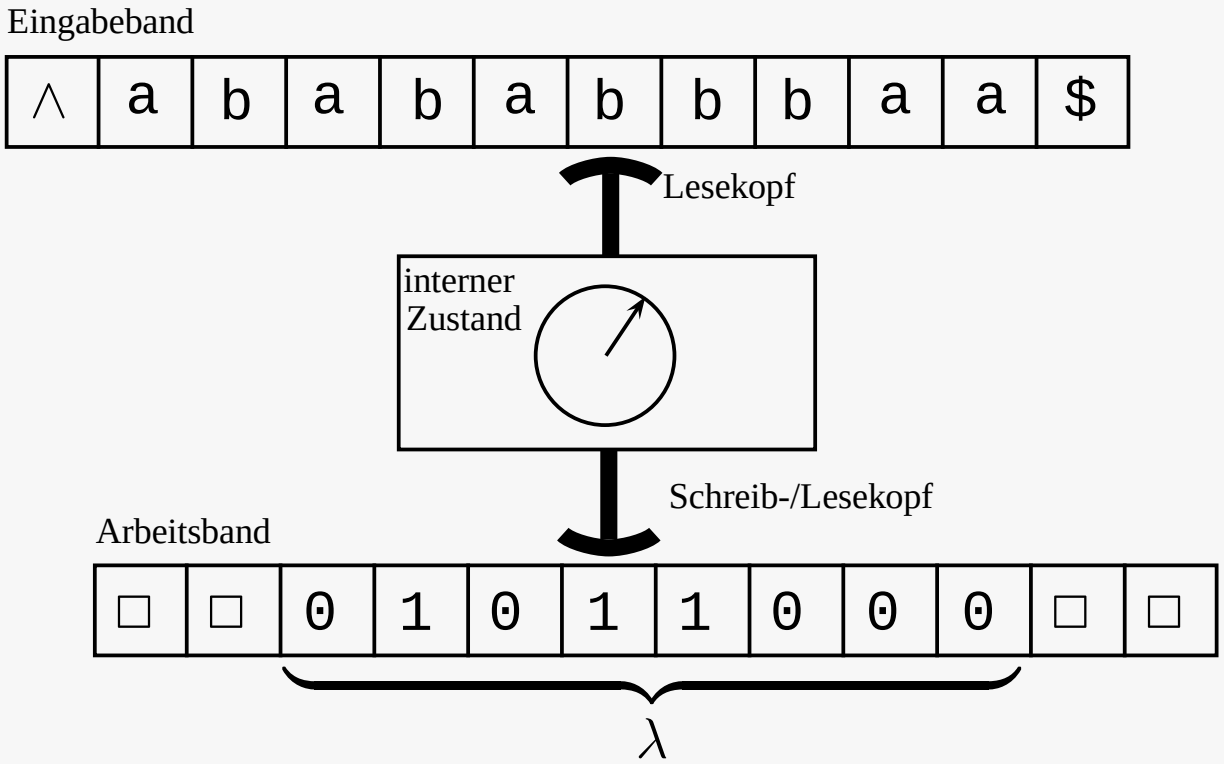
Speicher-Konfigurationen



Speicher-Konfiguration :=

(interner Zustand , String auf Arbeitsband, Position Schreib-/Lesekopf)

Speicher-Konfigurationen



Speicher-Konfiguration :=
(interner Zustand , String auf Arbeitsband, Position Schreib-/Lesekopf)

System-Zustand :=
Speicher-Konfiguration + Zeichen auf Eingabeband

Platzbedarf

Sei $L :=$ nicht-reguläre Sprache, erkannt von Einweg-DTM M

- Gesucht:
  maximaler Platzbedarf bei gegebener Eingabelänge; $\lambda(n) \leq \lambda$
- Equivalent:
  minimale Wortlänge bei gegebenem Platzbedarf; $n \leq n(\lambda)$

Vorteil: Fester Platz $\Rightarrow$ endlich viele *System*-Zustände

$$\begin{aligned} \text{minimale Wortlänge} &= \text{maximale Periodendauer} \\ &\leq \text{Menge aller } \textit{System}\text{-Zustände} \end{aligned}$$

Das kürzeste Wort

Wähle w, kürzestes Wort mit Platzbedarf genau λ, Länge $|w|$

$$w = x_1 \dots x_r \dots x_s \dots x_{|w|} \quad \in L$$

Abpumpen

- $x_r = x_s$
- Speicher bei $x_r =: \quad c_{x_r} = c_{x_s} \quad :=$ Speicher bei x_s

$\Rightarrow \quad w' = x_1 \dots x_r x_{s+1} \dots x_{|w|} \in L$

$$|w'| < |w| \Rightarrow \lambda(w') < \lambda$$

$$w' = x_1 \dots x_r x_{s+1} \dots x_{|w|} \in L \qquad \wedge \qquad \lambda(w') < \lambda$$

Übergang

- $x_r \rightarrow_{w'} x_{s+1}$ mit $c_{x_r} = c_{x_s} \wedge x_r = x_s$

$\Rightarrow$ $x_r \rightarrow_{w'} x_{s+1} \quad \equiv \quad x_s \rightarrow_w x_{s+1}$

- *System*-Zustände von $w' \subset$ *System*-Zustände von w

$\Rightarrow$ Haltekonfiguration gleich für w und w'

↯ w hält nicht für Platz $< \lambda$

$\Rightarrow$ gleiches Eingabezeichen bedeutet
unterschiedliche Speicher-Konfiguration

Zustände zählen

$$x_r = x_s \quad \Rightarrow \quad c_{x_r} \neq c_{x_s}$$

$$\Rightarrow\ |w| =: n \leq |\text{Eingabealphabet}|\ |\text{Speicher-Konfig. mit Platz } \leq \lambda|$$

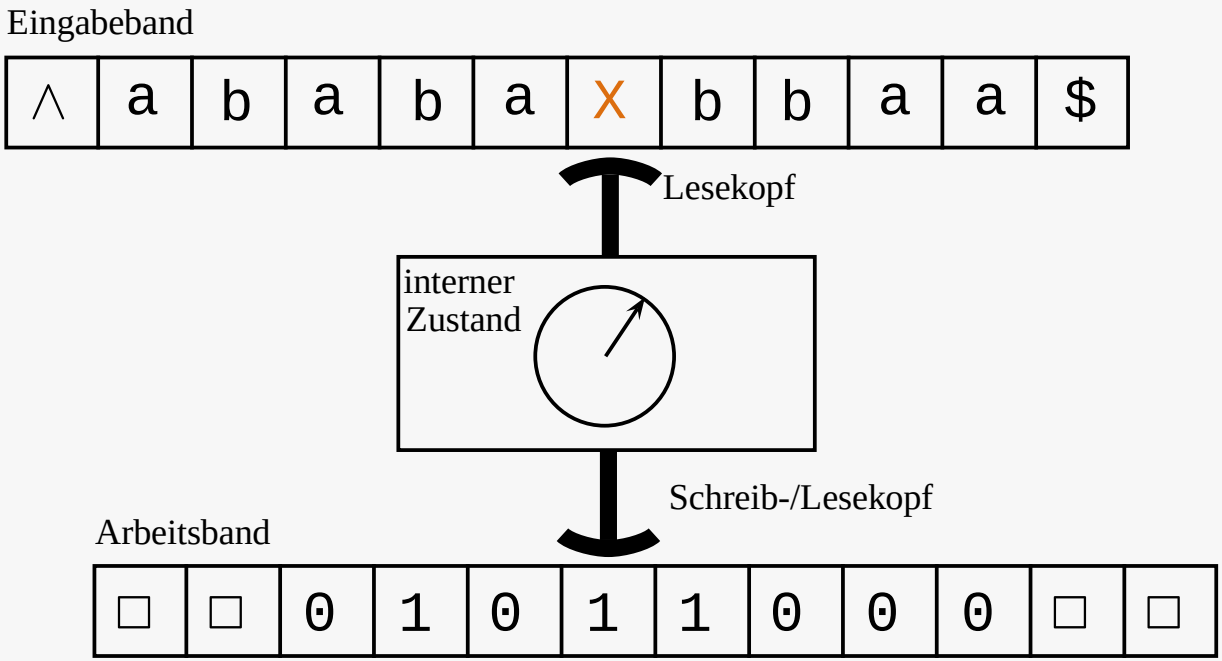
$$\text{Speicher-Konfiguration mit Platzbedarf genau } i := (\ \text{interner}\ \underset{q}{\text{Zustand}}\ ,\ \underbrace{\text{String auf Arbeitsband}}_{k^i},\ \text{Position}\ \underset{i}{\text{Schreib-/Lesekopf}})$$

$$\text{Speicher-Konfigurationen mit Platzbedarf weniger/gleich } \lambda = \sum_{i=1}^{\lambda} q\ k^i\ i \leq p^\lambda$$

Platzbedarf von Einweg-DTM

$$n \lesssim p^\lambda \quad \Rightarrow \quad \lambda \geq \log n$$

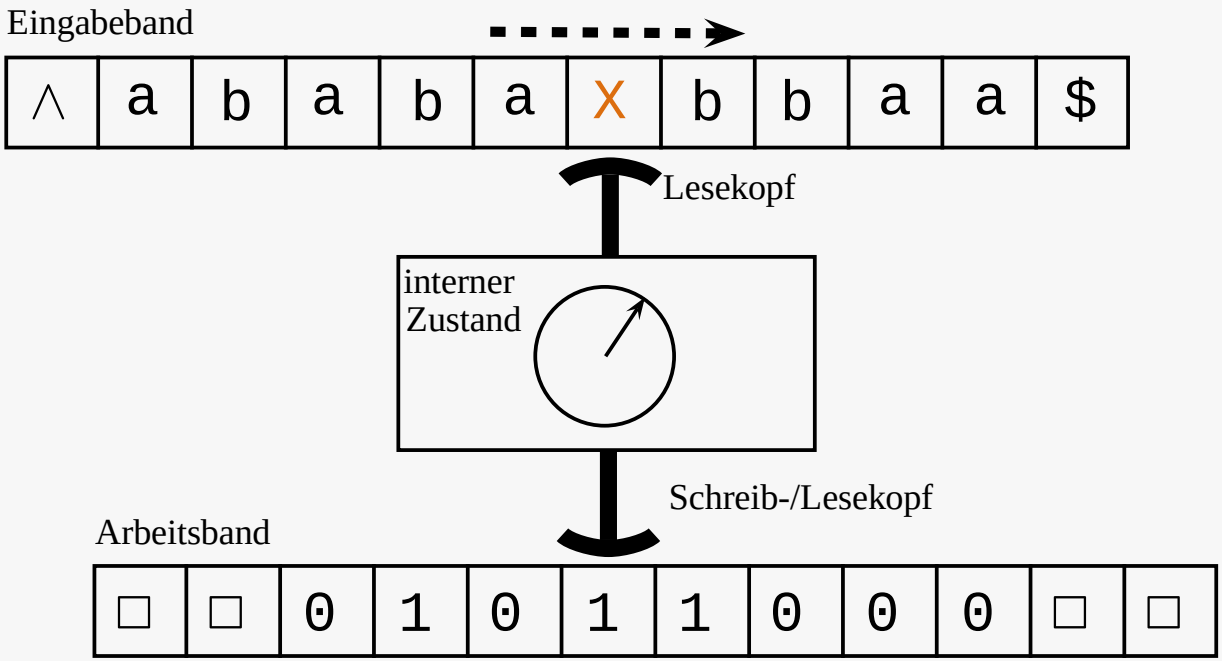
Zweiwege-DTM



Lesekopf des Eingabebands = Rechts/Warten/Links

Man sieht sich immer zweimal.

Zweiwege-DTM



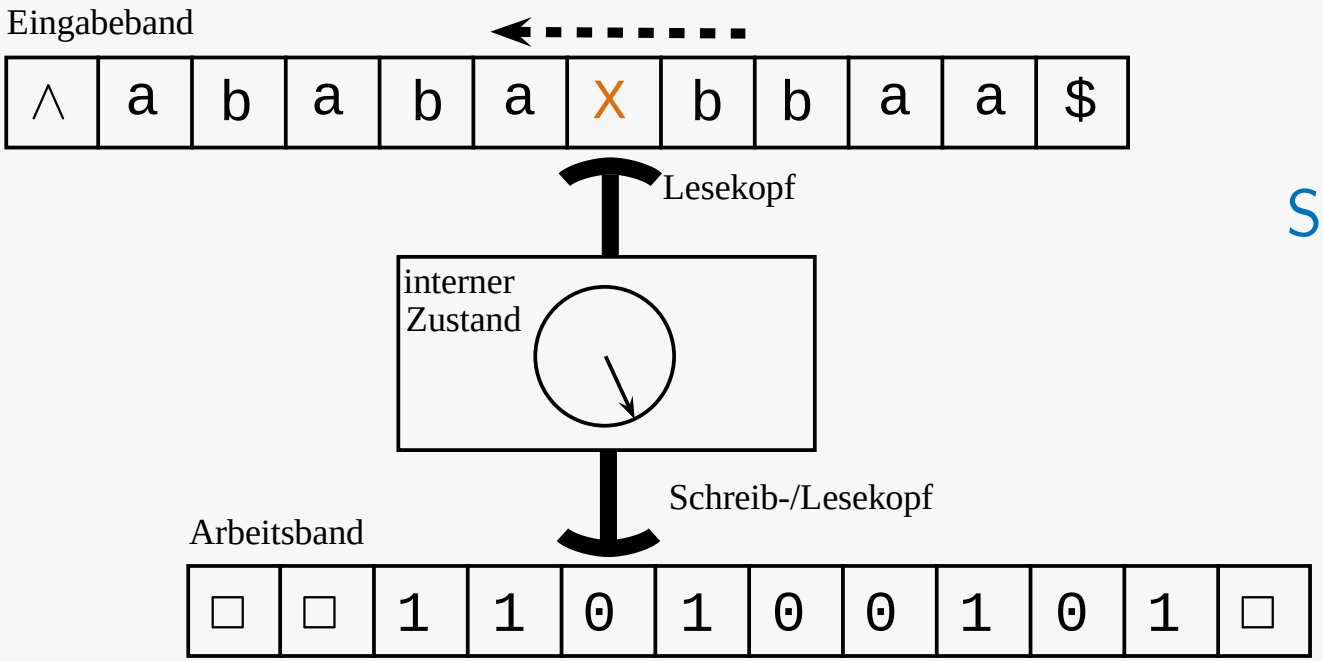
Speicher bei X

(15° 01011000 6)

Lesekopf des Eingabebands = Rechts/Warten/Links

Man sieht sich immer zweimal.

Zweiwege-DTM



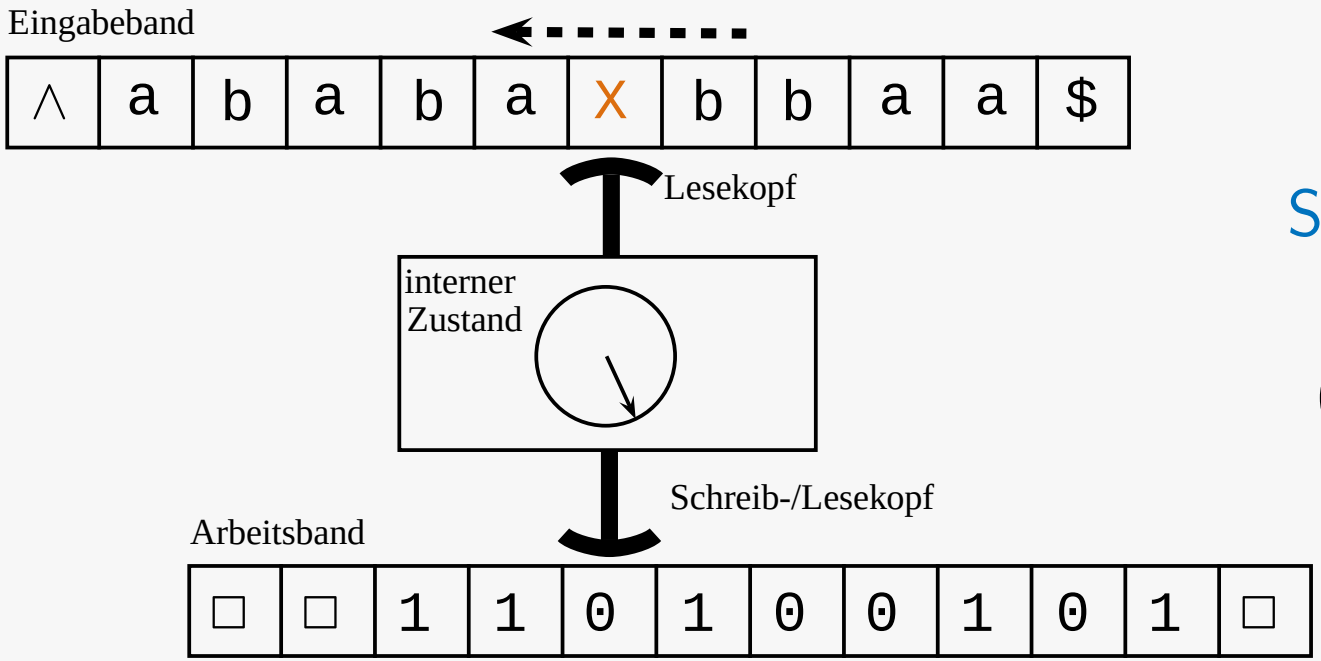
Speicher bei X

(15° 01011000 6)

Lesekopf des Eingabebands = Rechts/Warten/Links

Man sieht sich immer zweimal.

Zweiwege-DTM



Speicher bei X

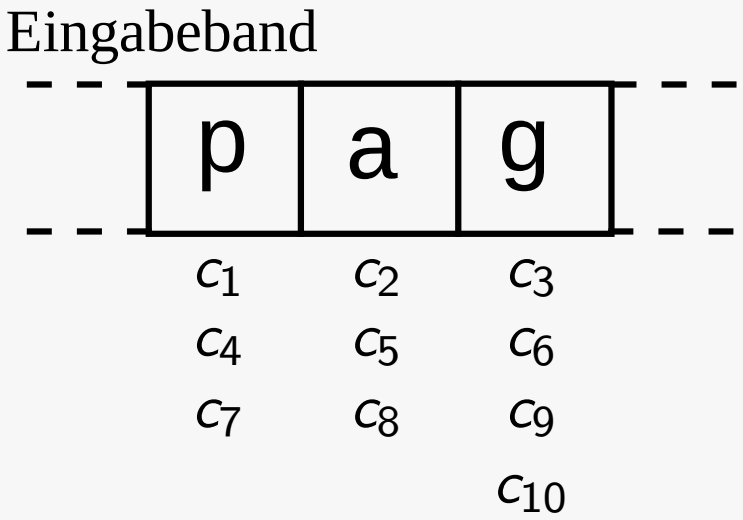
$$\begin{array}{ccc} (\ 15^\circ & 01011000 & 6) \\ (165^\circ & 110100101 & 5) \\ \vdots & \vdots & \vdots \end{array}$$

Lesekopf des Eingabebands = Rechts/Warten/Links

Man sieht sich immer zweimal.

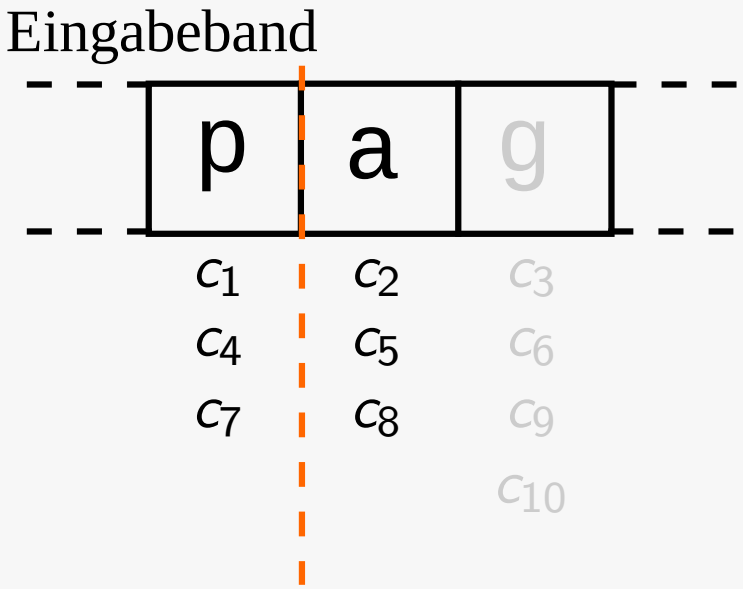
System-Zustand $\neq$ Speicher-Konfig. + Eingabe

$c :=$ Speicher-Konfiguration



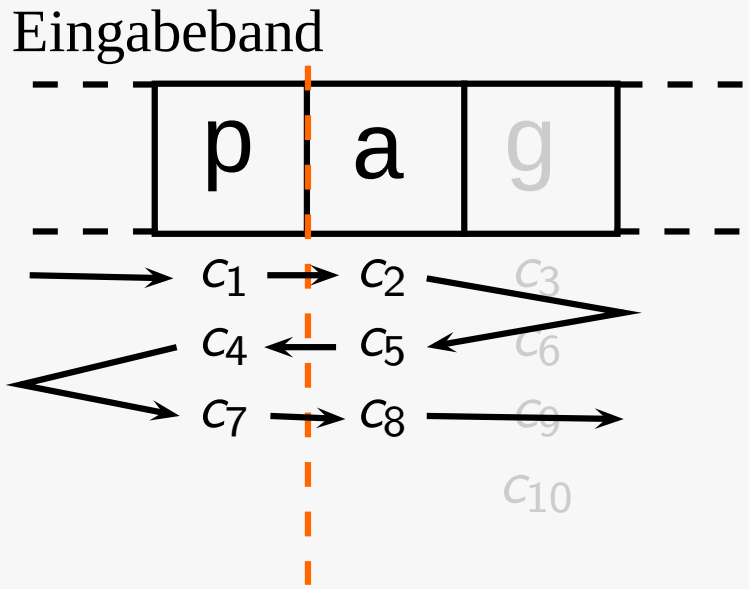
Teilmenge der Menge aller Speicher-Konfigurationen

$c :=$ Speicher-Konfiguration



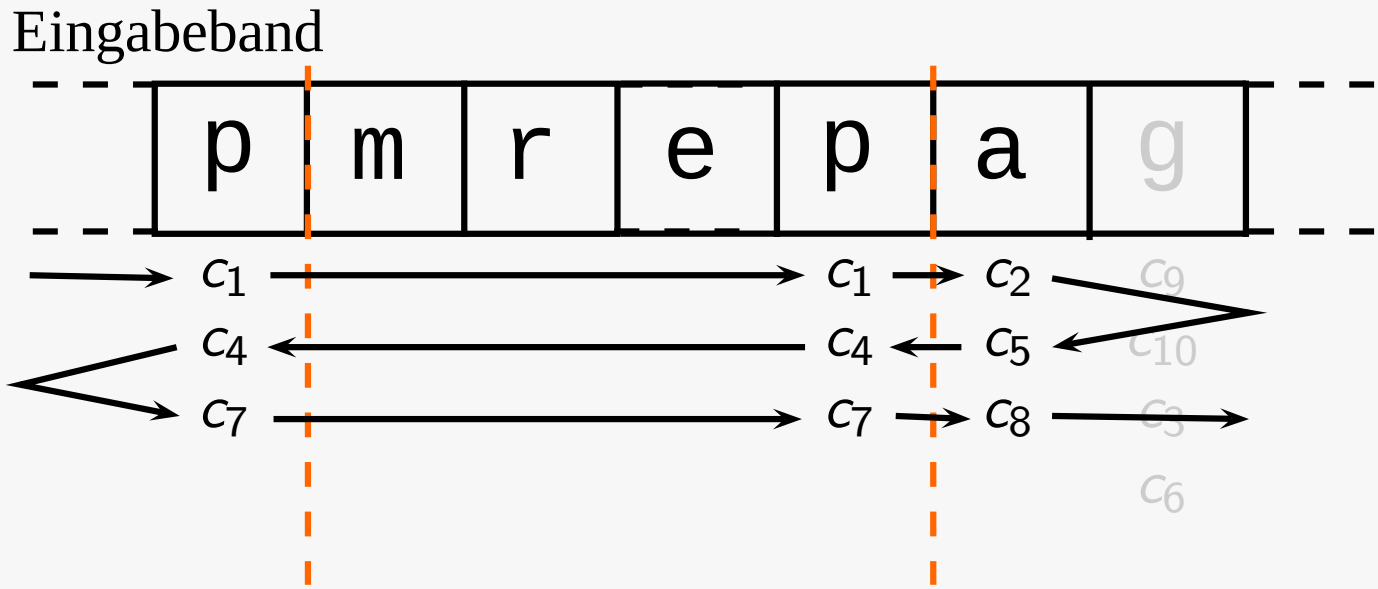
Teilmenge der Menge aller Speicher-Konfigurationen

$c :=$ Speicher-Konfiguration



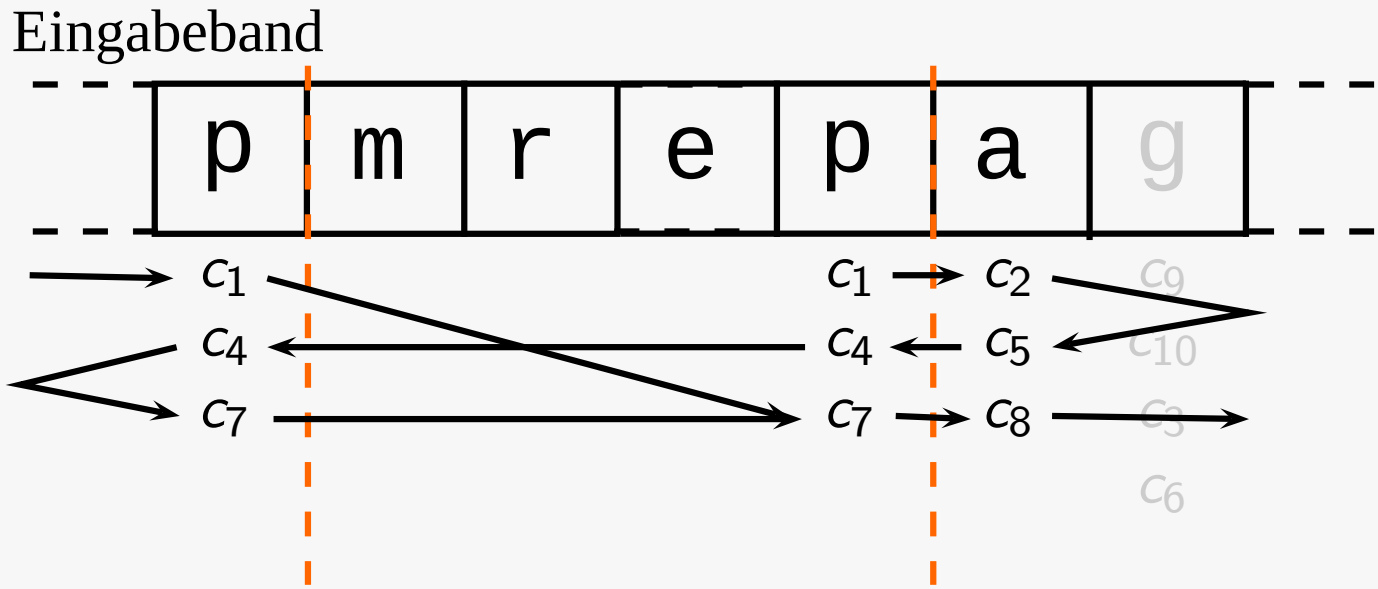
Teilmenge der Menge aller Speicher-Konfigurationen

$c :=$ Speicher-Konfiguration



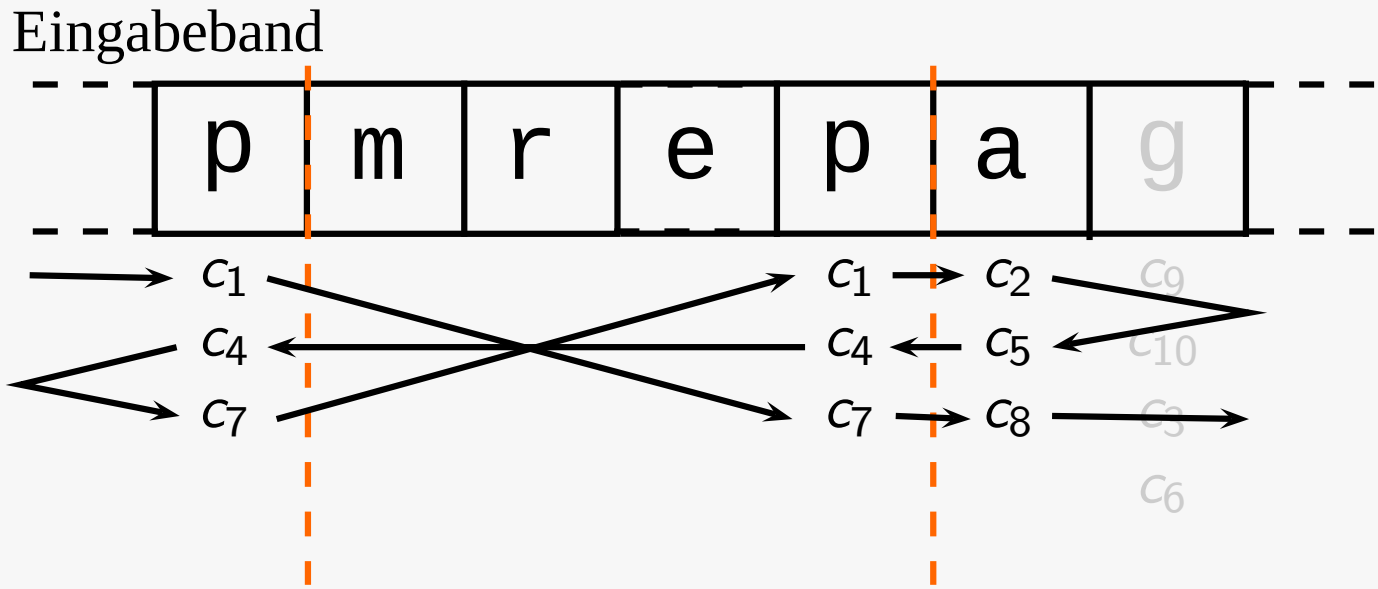
Teilmenge der Menge aller Speicher-Konfigurationen

$c :=$ Speicher-Konfiguration



Teilmenge der Menge aller Speicher-Konfigurationen

$c :=$ Speicher-Konfiguration



Teilmenge der Menge aller Speicher-Konfigurationen

Gleiches Spiel wie für Einweg-DTM

System-Zustände = Mengen von Speicher-Konfig. + Eingabe

Wie bisher:

- Suche kürzestes Wort mit Platzbedarf λ
- Zeige, dass gleiche *System*-Zustände nicht auftreten können
- Finde obere Schranke:

$$\begin{aligned} n &\leq |\text{Eingabealphabet}|\ |\text{Potenzmenge aller Speicher-Konfig.}| \\ &\leq |\text{Eingabealphabet}|\qquad 2^{p^\lambda} \end{aligned}$$

$$\lambda \geq \log\log n$$

Rigoros

Was bisher geschah...

- endlich viele *System*-Zustände $\Rightarrow$ Periodizität
- Einweg-DTM: Speicher-Konfigurationen $\Rightarrow \lambda \geq \log n$
- Zweiwege-DTM: Mehrere Speicher-Konfig. $\Rightarrow \lambda \geq \log\log n$

Rigoros

Was bisher geschah...

- endlich viele *System*-Zustände $\Rightarrow$ Periodizität
- Einweg-DTM: Speicher-Konfigurationen $\Rightarrow \lambda \geq \log n$
- Zweiwege-DTM: Mehrere Speicher-Konfig. $\Rightarrow \lambda \geq \log\log n$

...zum Weiterlesen

Formaler Zugang über Äquivalenzrelation/klassen:

Andrzej Szepietowski, Turing Machines with Sublogarithmic Space, LNCS Band 843, Springer, 1994. Kapitel 5.1, (Google Books)

Original, sehr anschaulich:

R.E. Stearns et al., Hierarchies of Memory Limited Computations, FOCS '65: Proceedings of the 6th Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1965)