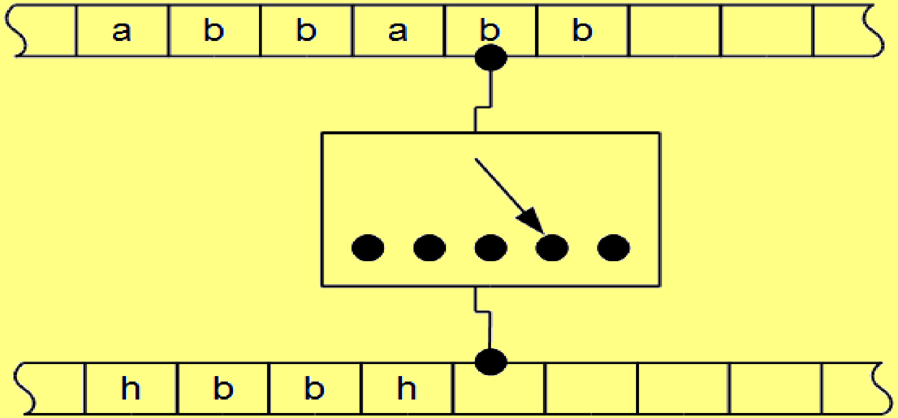
Sprachen und Automaten

Tino Hempel

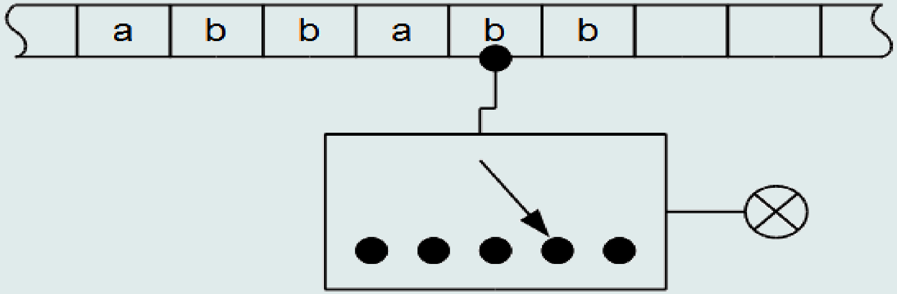
Mealy-Automat vs. Akzeptor

Automat mit Ausgabe



Das abstrakte Modell des endlichen Automaten mit Ausgaben erfasst die wesentlichen Aspekte eines Computers (EVA-Prinzip).

Automat ohne Ausgabe (deterministischer Akzeptor)



Die Sprache eines Automaten L(A) ist die Menge aller von ihm akzeptierten Wörter über das Eingabealphabet X.

Mealy-Automat vs. Akzeptor

$A = (X, Y, Z, \delta, \lambda, q_0)$

$A = (X, Z, \delta, q_0, Z_E)$

- X ... Eingabealphabet (nichtleere, endliche Menge)
- Z ... Zustandsmenge (nichtleere, endliche Menge)
- $\delta: X \times Z \rightarrow Z$ … Überführungsfunktion, welche jedem Paar (Eingabezeichen, Zustand) einen Folgezustand zuordnet
- $q_0 \in Z$ ist der Anfangszustand

- Y ... Ausgabealphabet (nichtleere, endliche Menge)
- $\lambda: X \times Z \rightarrow Y$ … Ausgabefunktion, welche jedem Paar (Eingabezeichen, Zustand) ein Ausgabezeichen zuordnet

- $Z_E \subseteq Z$ ... Endzustandsmenge

Mealy-Automat vs. Akzeptor

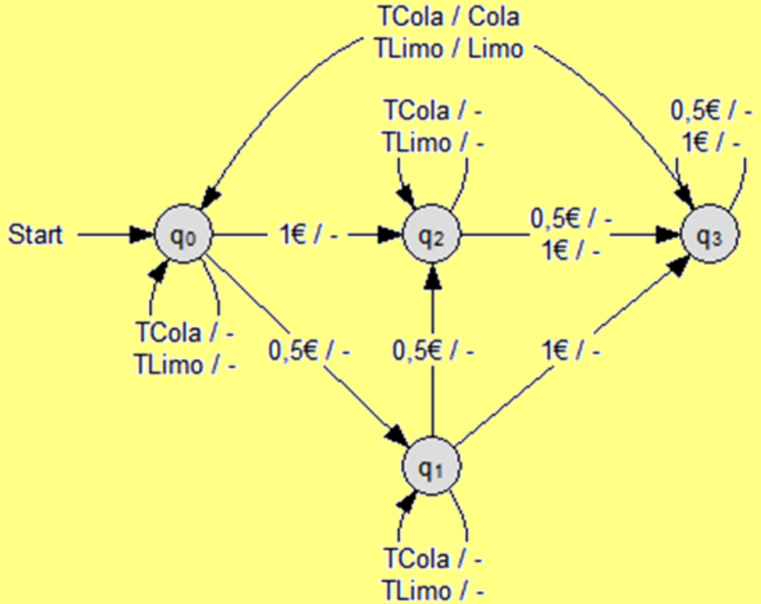
Bestimmen Sie die Mengen

X = { }, Z = { },

Y = { }

Beschreiben Sie die Bedienung.

Geben Sie den Preis der Getränke an.



Bestimmen Sie die Mengen

X = { }, Z = { },

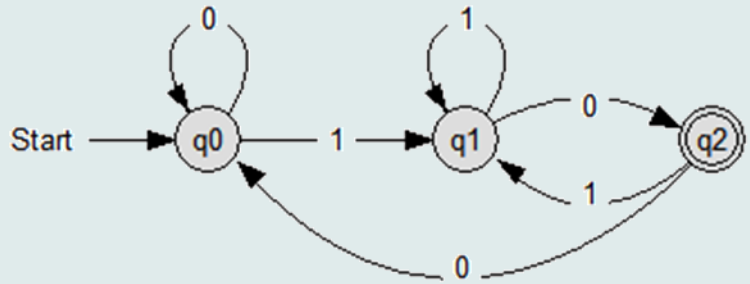
Z_E = { }

Setzen Sie $\in$ oder $\notin$ korrekt :

w_1= 0110 L(A), w_2= 1100 L(A)

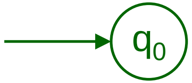
w_3= 01101 L(A), w_4= 1110 L(A)

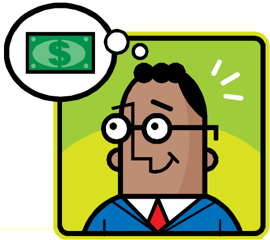
Bestimmen Sie die Sprache des Automaten L(A) = { }



Die 1-Millionen-Euro-Aufgabe:

Entwickeln Sie einen Akzeptor A, der die Sprache $L(A) = \{a^n b^n \mid n \in \mathbf{N}, n > 0\} = \{ab, aabb, aaabbb, \ldots\}$ mit $X = \{a, b\}$ erkennt.





Grenzen des Akzeptor

Es ist **nicht möglich** einen Akzeptor A zu konstruieren, der die Sprache $L(A) = \{a^n b^n \mid n \in \mathbf{N}, n > 0\}$ mit $X = \{a, b\}$ erkennt, da der Automat **unendlich viele Zustände** benötigen würde. Dies steht im Widerspruch zur Definition des Akzeptors!

Es gibt also (mindestens eine) Sprache(n), für die man keinen Akzeptor bauen kann.

Ist die Sprache $L = \{a^n b^n \mid n \in \mathbf{N}, n > 0\}$ denn so wichtig?

Grenzen des Akzeptor

Ein Rechner muss solche Sprachen erkennen können, z. B. für korrekt geklammerte Ausdrücke in mathematischen Termen.

Beispiel: Klammersprache ()

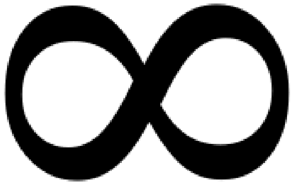
Die Benutzung von Klammern ist nur dann korrekt, wenn

- die Anzahl „(“ ist gleich der Anzahl „)“ und
- in jedem Anfangsstück $w_1, \ldots, w_i$ $(i \leq n)$ die Anzahl „(“ nicht kleiner als die Anzahl „)“ ist.

Kellerautomat

Problembeschreibung

Akzeptor kann sich nicht unendlich viele Informationen merken



Problemlösung

Erweiterung des Akzeptors um eine Speichereinheit, dem sog. **Kellerspeicher**

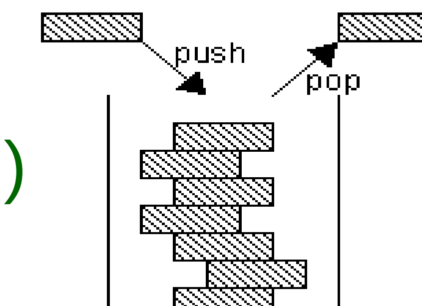
Kellerautomat

Kellerspeicher (LIFO – last in first out)

- unendlich groß
- wächst nach oben
- Kellerboden: durch Symbol # markiert

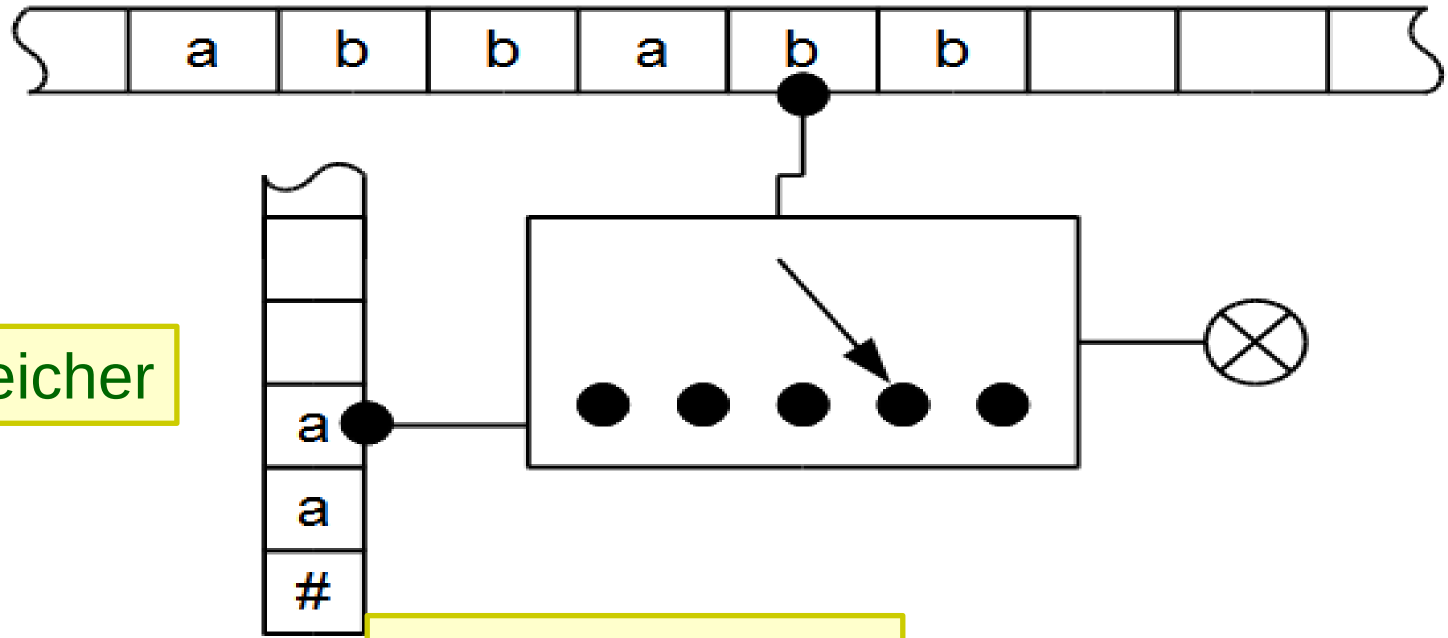
Kellerbefehle

- **Entnehmen** des obersten Zeichens (pop)
- **Schreiben eines Zeichens** (push Zeichen)
- keine Operation (nop)



Kellerautomat

Aufbau



Kellerspeicher

Kellerleerzeichen

Kellerautomat

Ein (nichtdeterministischer) **Kellerautomat** KA = (X, Z, Γ, δ, q_0, k_0, Z_E) wird definiert durch:

- X … Eingabealphabet (nichtleere, endliche Menge)
- Z … Zustandsmenge (nichtleere, endliche Menge)
- Γ … Kelleralphabet (nichtleere, endliche Menge)
- δ … eine (nicht eindeutige oder nicht vollständig definierte) Überführungsfunktion, welche jedem Tupel (Eingabezeichen, Kellerzeichen, Zustand) ein Kelleroperation und einen Folgezustand zuordnet
- $q_0 \in Z$ ist der Anfangszustand
- $k_0 = \#$, $k_0 \in \Gamma$ ist das Kellervorbelegungszeichen
- $Z_E \subseteq Z$ ist die Menge der Endzustände

Kellerautomat

Akzeptieren eines Wortes

Ein Wort wird nur dann akzeptiert, wenn

- ein Endzustand erreicht wurde,
- die Eingabe abgearbeitet wurde und
- der Keller leer ist.

Darstellung des Automatengraph

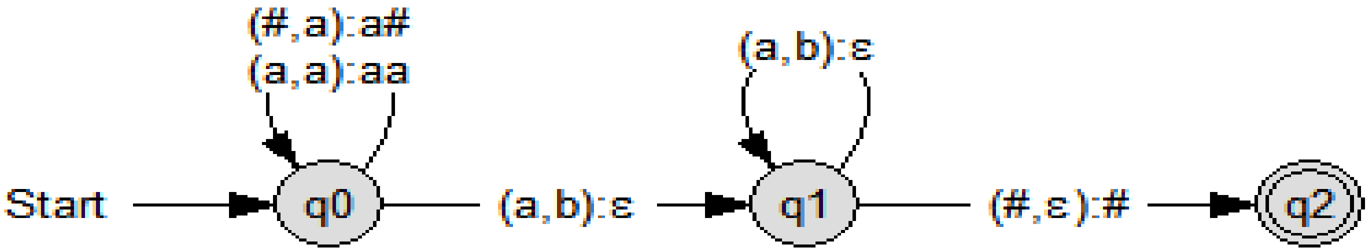
Kellerautomat

Beispiel für einen Kellerautomaten

Sprache $L(A) = \{a^n b^n \mid n \in N, n > 0\}$ mit $X = \{a, b\}$

Einkellern aller a

Auskellern



Übergang, wenn ein b kommt

Übergang zum Endzustand, falls Eingabe und Keller leer

ACHTUNG: Jeder Blick in den Keller löscht das oberste Zeichen

ε als Eingabezeichen … **keine** Eingabe

ε als Kelleroperation … nichts schreiben

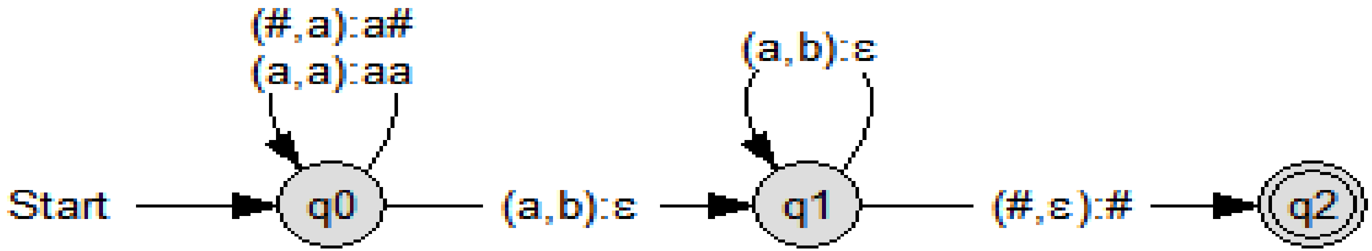
Kellerautomat

Probleme in der Simulation mit AutoEdit

Jeder Blick in den Keller löscht das oberste Zeichen

Lösung:

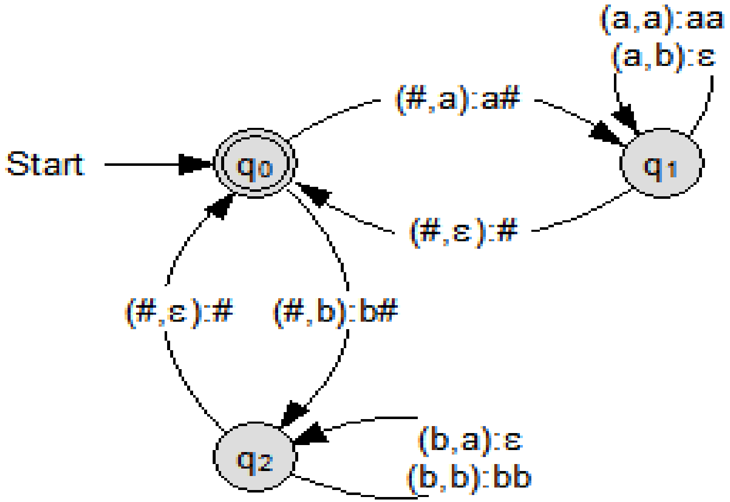
- Push: Schreiben des neuen und gelesenen Zeichens
- Pop: keine Veränderung des Kellers (Epsilon)
- Nop: Schreiben des gelesenen Zeichens



Übungen

Untersuchen Sie den folgenden Kellerautomaten KA.

1. Bestimmen Sie X, Z, Z_E, Γ.
2. Untersuchen Sie die Arbeitsweise von KA für die Wörter w_1 = aabb, w_2 = baba und w_3 = aba
3. Bestimmen Sie die Sprache des Automaten.



Rückfragen