Formale Sprachen

Eine formale Sprache beschreibt die Syntax, jedoch nicht die Semantik

- Σ Alphabet (zugelassene Zeichen)
- ω Wort, $\boxed{\omega \in \Sigma^*}$
- L formale Sprache, $\boxed{L \subset \Sigma^*}$

Automaten I

DEA = deterministischer endlicher Automat → Akzeptor: überprüft, ob ein Wort zur Sprache gehört

alle Übergänge eindeutig

- Σ Eingabealphabet
- Q Zustände, $\boxed{q1, q2, \ldots}$
- s Startzustand, $\boxed{s \in Q}$ → Verarbeitung beginnt hier
- F Endzustände, $\boxed{F \subset Q}$ → Wenn wir nach vollständiger Verarbeitung der Eingabe hier sind, ist die Eingabe akzeptiert
- $\delta: Q \times \Sigma \to Q$ Übergangsfunktion, dargestellt als Zustandsübergangsgraph

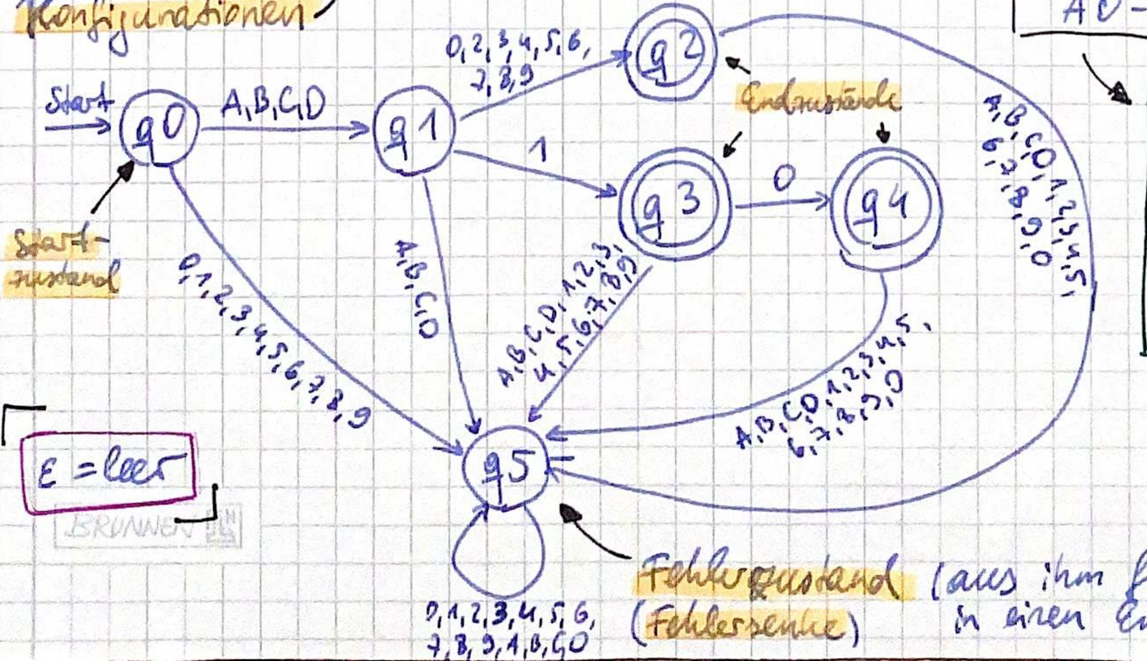
Zustand   Rest der Eingabe

$(q0, A4) \rightarrow$
$\rightarrow (q1, 4) \rightarrow$
$\rightarrow (q2, \varepsilon)$

Konfigurationen

(Papierformate A0 – A10, B1 – B10, ...)

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D\}$
$Q = \{q1, q2, q3, q4, q5, q0\}$
$s = q0$
$F = \{q2, q3, q4\}$



ε = leer

Fehlerzustand (aus ihm führt kein Weg in einen Endzustand)
(Fehlersenke)

Automaten II

DKA = deterministischer Kellerautomat

Keller = Stack

- Σ Eingabealphabet
- Q Zustände
- s Startzustand, $s \in Q$
- F Endzustände, $F \subset Q$
- K / Γ Kelleralphabet mit Kellervorbelegungszeichen #, $\# \in \Gamma$
- δ Übergangsfunktion, ordnet einer Kombination aus Zustand, Eingabezeichen u. Kellerzeichen einen Nachfolgezustand und eine Kelleroperation zu

same wie bei DEA

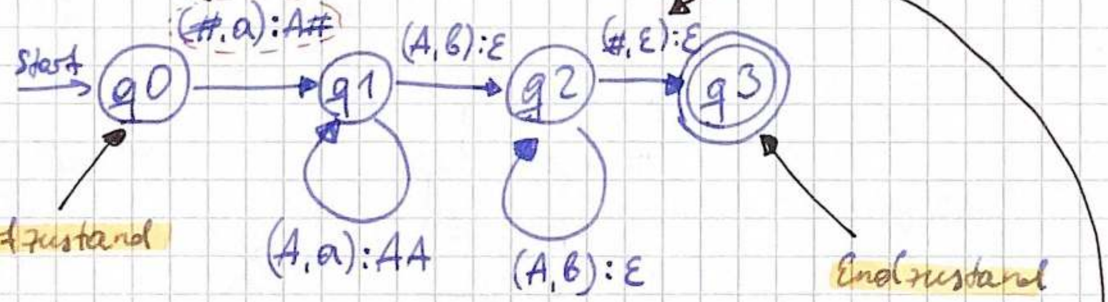
das Einzige, was zu Beginn auf dem Stack liegt

n-mal „a"
n-mal „b"

nicht Potenz!

Wörter der Form $a^n b^n$

$\Sigma = \{a, b\}$
$Q = \{q0, q1, q2, q3\}$
$s = q0$
$F = \{q3\}$
$\Gamma = \{\#, A, B\}$



lies:
„wenn auf dem Keller # liegt
und das Eingabezeichen a ist,
dann leg auf den Keller # und dann A"

Konfigurationen bestehen hier aus Zustand, Eingabe und aktueller Kellerbelegung

ε-Übergang:
nur Kellerzeichen wird berücksichtigt, Eingabezeichen kann beliebig sein

nicht eingezeichnete Übergänge führen hier (automatisch) zu einem Fehler

→ müssen nicht eingezeichnet werden

müssen aber beim DEA eingezeichnet werden oder vermerkt werden !

Automaten III

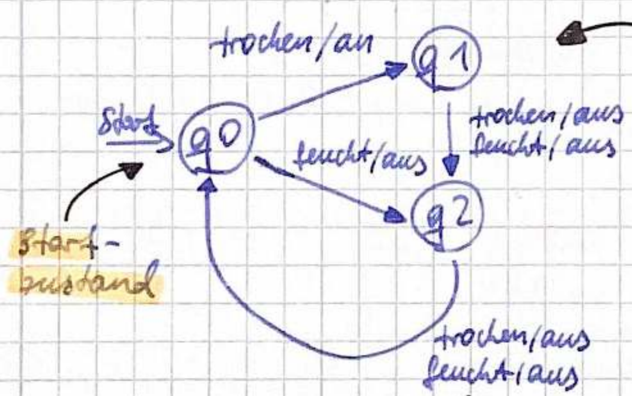
Mealy-Automat

→ „Aktor" (Bewässerungsanlage, Getränkeautomat)

→ Transduktor (Codierung)

- $\boxed{\Sigma}$ Eingabealphabet
- $\boxed{\Omega}$ Ausgabealphabet
- $\boxed{Q}$ Zustände
- $\boxed{s}$ Startzustand, $\boxed{s \in Q}$
- kein Endzustand
- $\boxed{\delta : Q \times \Sigma \rightarrow Q}$ Übergangsfunktion (Zustand + Eingabe → Zustand)
- $\boxed{\lambda : Q \times \Sigma \rightarrow \Omega}$ Ausgabefunktion (Zustand + Eingabe → Ausgabe)

Bewässerungsanlage, Eingabe vom Sensor (feucht/trocken), Ausgabe an die Pumpe (an = „an für 1s", aus = „aus für 1min")



$\Sigma = \{„feucht", „trocken"\}$
$\Omega = \{„an", „aus"\}$
$Q = \{q0, q1, q2\}$
$s = q0$

Notation des Übergangs:
Eingabe/Ausgabe

Grammatiken

Eine Grammatik erzeugt eine Sprache, ein zugehöriger Automat erkennt diese

- V / N Nichtterminalsymbole (Variablen) → Großbuchstaben
- Σ / T Terminalsymbole (Alphabet) → Kleinbuchstaben, Zahlen, ...
- P Produktionsregeln
- S Startsymbol $\boxed{S \in N}$

Chomsky-Hierarchie	Grammatiktypen	
Typ 1	kontextsensitiv	auf der linken Seite der Produktionsregeln höchstens ein Nicht-Terminal, eins oder mehr Terminale, die den Kontext vergeben $\boxed{aBdc \rightarrow agdc}$
Typ 2	kontextfrei	auf der linken Seite der Produktionsregeln genau ein Nicht-Terminal $\boxed{A \rightarrow dBA}$
Typ 3	regulär	auf der rechten Seite höchstens ein Nicht-Terminal $A \rightarrow dB$ → rechts → rechtsregulär $A \rightarrow Bd$ → links → linksregulär

Papiertypen
! a1, a2, ..., b1, ...

$N = \{S, B, Z\}$
$T = \{a, b, c, d, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$
$S = S$
P:
- $S \rightarrow BZ$
- $B \rightarrow a \mid b \mid c \mid d$
- $Z \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 10$

kontextfrei
(mehrere Nicht-Terminale)

Beziehung von Automaten zu Grammatiken

Zu einem DEA gehört immer eine reguläre Grammatik dazu

DEA		reguläre Grammatik
Σ	$\hat{=}$	T
Q	$\hat{=}$	N
s	$\hat{=}$	S
F	$\hat{=}$	$F \rightarrow \varepsilon$ (Überführung des Zustandes in ein leeres Terminal) s.u.
δ	$\hat{=}$	P
$A \xrightarrow{a,b,c} B$	$\hat{=}$	$A \rightarrow aB \mid bB \mid cB$
F (Endzustand)	$\hat{=}$	$F \rightarrow \varepsilon$ leer, sonst fände unser Wort kein Ende
A mit Schleife n, m	$\hat{=}$	$A \rightarrow nA \mid mA$