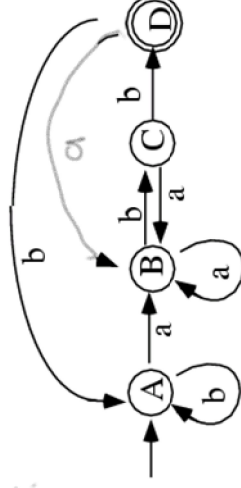


Systematische Scanner-Implementierung

Anmerkung

- Zur Implementierung eines Scannergenerators benötigt man einen Parser für reguläre Ausdrücke
- Implementierung wird einfacher, wenn man nur einen Endzustand zulässt (ggf. neuen Endzustand und ϵ -Übergänge von den ursprünglichen Endzuständen in den neuen einführen)
- Man kann aus dem NEA durch Teilmengen-Konstruktion einen DEA ableiten (siehe THI)
- Es gibt andere Verfahren, die direkt einen DEA konstruieren (hier nicht behandelt)

DEA A_1 :



mpten 44

descriptore Sprache;

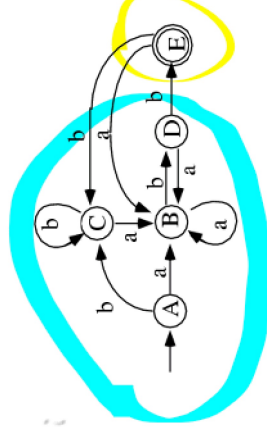
$$L((a|b)^*abb)$$

Systematische Scanner-Implementierung

Satz (neu!)

Zu jedem (nichtdeterministischen) endlichen Akzeptor existiert ein äquivalenter deterministischer endlicher Akzeptor mit minimaler Zustandsmenge. Bis auf die Zustandsbenennung ist der minimale Akzeptor eindeutig bestimmt.
(ohne Beweis)

DEA A_2 :



Systematische Scanner-Implementierung

Algorithmus:

Die Zustandsmenge von A_0 wird mit S bezeichnet. Jeder Zustand $z \in S$ repräsentiert eine Klasse äquivalenter Zustände aus der Zustandsmenge S des Ausgangsautomaten A .

Die Äquivalenzklasseneinteilung wird sukzessive durch Verfeinerung berechnet:

1. (Initialisierung der Äquivalenzklassen) $S := \{F, S \setminus F\}$
(Für $w \neq \epsilon$ sind alle Endzustände und alle Nicht-Endzustände unterscheidbar)
2. Verfeinerung der Äquivalenzklassen (siehe nächste Seite)
3. Solange neue Klasseneinteilung feiner, wiederhole Verfeinerung, falls Klasseneinteilung sich nicht ändert, beende Verfeinerung

Systematische Scanner-Implementierung

Algorithmus:

Verfeinerung der Äquivalenzklassen:

Für jeden Zustand $z \in S$ verteile die Zustände in der Klasse z so auf Teilklassen, dass zwei Zustände des Ausgangsautomaten $s_1, s_2 \in S$ genau dann in der gleichen Teilklass z sind:

wenn für alle Eingabesymbole $x \in A$ gilt:

$\text{next}(s_1, x)$ und $\text{next}(s_2, x)$ sind in der gleichen Klasse von S

Führe die Aufteilung der Zustände in Teilklassen so durch, dass möglichst wenig Teilklassen entstehen

Ersetze die Zustände S durch die neu gebildeten Teilklassen.

Minimiere Anzahl der Zustände für A_2 :

1. $A_1 = \{E\}$, $A_2 = \{A, B, C, D\}$

$\text{next}(D, b) = E \in A_1 \rightarrow D$ abgespalten

$\text{next}(A, b) = C \in A_2$

$\text{next}(B, b) = D \in A_2$ gleiches Verhalten

$\text{next}(C, b) = C \in A_2$

$\text{next}(D, a) = B \in A_2$

$\text{next}(A, a) = B \in A_2$ gleiches Verhalten

$\text{next}(B, a) = B \in A_2$

$\text{next}(C, a) = B \in A_2$

2. Äquivalenzklasseneinteilung (Verfeinerung)

$A_1 = \{E\}$, $A_2 = \{A, B, C\}$, $A_3 = \{D\}$

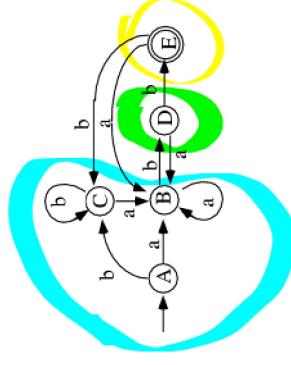
alle Zustände aus A_2 behalten

sich bei Eingabe a gleich

$\text{next}(B, b) = D \in A_3 \rightarrow$ abgespalten!

$\text{next}(A, b) = C \in A_2$

$\text{next}(C, b) = C \in A_2$



3. Äquivalenzklasseneinteilung: (Vorfierung)

$$A_1 = \{E\}, A_2 = \{A, C\}, A_3 = \{D\}, A_4 = \{B\}$$

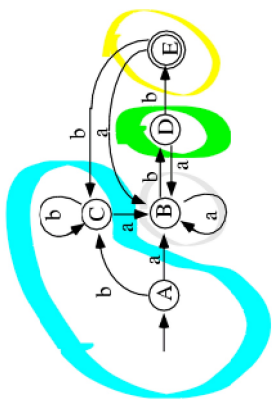
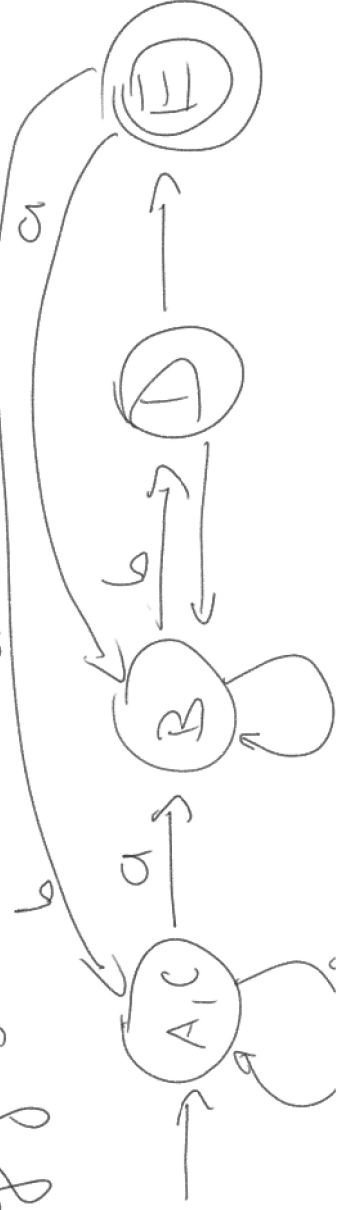
$$\text{next}(A, a) = B \in A_4$$

$$\text{next}(C, a) = B \in A_4$$

$$\text{next}(A, b) = C \in A_2$$

$$\text{next}(C, b) = C \in A_2$$

$\Rightarrow$ Zerlegung ist stabil wenn DEA hat 4 Zustände:



)

)



DEF A_1 :

