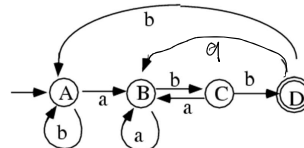


Systematische Scanner-Implementierung

Satz (neu)

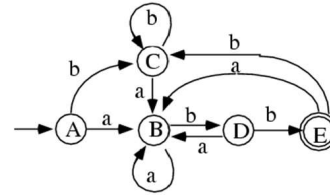
Zu jedem (nichtdeterministischen) endlichen Akzeptor existiert ein äquivalenter deterministischer endlicher Akzeptor mit minimaler Zustandsmenge. Bis auf die Zustandsbenennung ist der minimale Akzeptor eindeutig bestimmt.
(ohne Beweis)

DEA A_1



akzeptierte Sprache: $(a|b)^* a|bb$

DEA A_2 :



akzeptierte Sprache

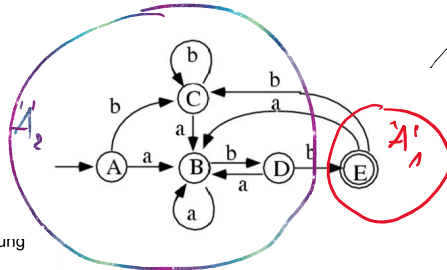
$(a|b)^* a|bb$

Minimiere die Anzahl der Zustände für A_2 :

Systematische Scanner-Implementierung

Algorithmus:

Die Zustandsmenge von A_1 wird mit S bezeichnet. Jeder Zustand $z \in S$ repräsentiert eine Klasse äquivalenter Zustände aus der Zustandsmenge S des Ausgangsautomaten A .
Die Äquivalenzklassenbildung wird sukzessive durch Verfeinerung berechnet:
1. (Initialisierung der Äquivalenzklassen) $S = \{F, S \setminus F\}$
(Für $w \in S$ sind alle Endzustände und alle Nicht-Endzustände unterscheidbar)
2. Verfeinerung der Äquivalenzklassen (siehe nächste Seite)
3. Solange neue Klassenentstellung findet, wiederhole Verfeinerung, falls Klassenentstellung sich nicht ändert, beende Verfeinerung
4. Zustandsübergänge sind durch die Übergänge der Repräsentanten definiert



1. $A'_1 = \{E\}$

$A'_2 = \{A, B, C, D\}$

$\text{next}(A, a) = B \in A'_2$

$\text{next}(B, a) = B \in A'_2$

$\text{next}(C, a) = B \in A'_2$

$\text{next}(D, a) = B \in A'_2$

$\text{next}(A, b) = C \in A'_2$

$\text{next}(B, b) = D \in A'_2$

$\text{next}(C, b) = C \in A'_2$

$\text{next}(D, b) = E \in A'_1$

gleiches Verhalten

gleiches Verhalten

$\rightarrow D$ abspalten

Systematische Scanner-Implementierung

Algorithmus:

Verfeinerung der Äquivalenzklassen:

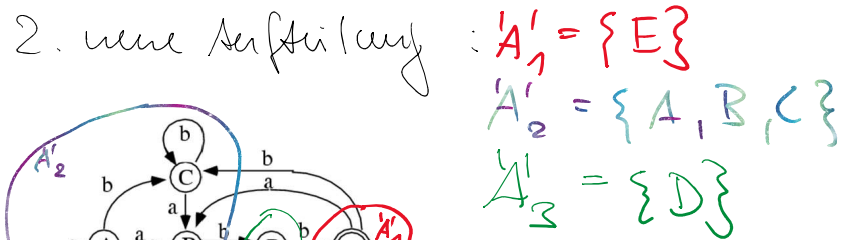
Für jeden Zustand $z \in S$ verteile die Zustände in der Klasse z so auf Teilklassen, dass zwei Zustände des Ausgangsautomaten $s_1, s_2 \in S$ genau dann in der gleichen Teilklass z sind:

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.



Alle Zustände verhalten sich bei Eingabe a gleich, jedoch nicht bei b:

$$\text{next}(A, b) = C \in A'_2$$

$$\text{next}(B, b) = D \in A'_3 \rightarrow \text{Abspalten}$$

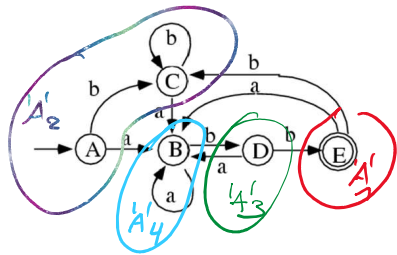
$$\text{next}(C, b) = C \in A'_2$$

3. neue Verfeinerung: $A'_1 = \{E\}$

$$A'_2 = \{A, C\}$$

$$A'_3 = \{D\}$$

$$A'_4 = \{B\}$$



$$\text{next}(A, a) = B \in A'_4$$

$$\text{next}(C, a) = B \in A'_4$$

$$\text{next}(A, b) = C \in A'_2$$

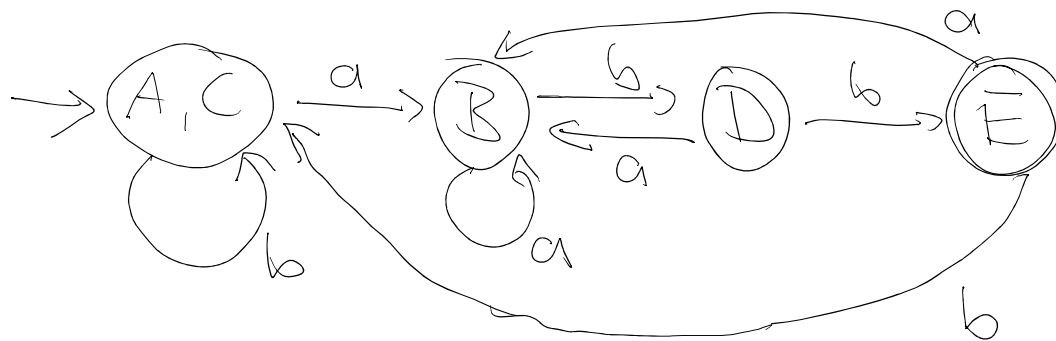
$$\text{next}(C, b) = C \in A'_2$$

Zerlegung ist stabil, kein weitere Verfeinerung nötig:
 neues DFA hat folgende Zustände:

Systematische Scanner-Implementierung

Implementierung des minimalen DEAs:

- Zustände durchnummerieren
- Ende der Eingabe-Zeichen beachten
- Eingabe Akzeptieren:
 - Automat ist in einem Endzustand
 - Eingabe ist vollständig gelesen (Ende der Eingabe ist aktuelles Zeichen)



entspricht A_1
bis auf Zustandsnamen

