Finite-State Technology

Teil IV:

Automaten (2. Teil)

Definition eines ε-NEA

- Ein ε-NEA ist ein Quintupel $A = (Q, \Sigma, \delta, q0, F)$, wobei

  Q = eine endliche Menge von Zuständen

  Σ = eine endliche Menge von Eingabesymbolen

  $q0$ = der Anfangszustand

  F = eine Menge von Endzuständen (akzeptierende Zustände), wobei $F \subseteq Q$

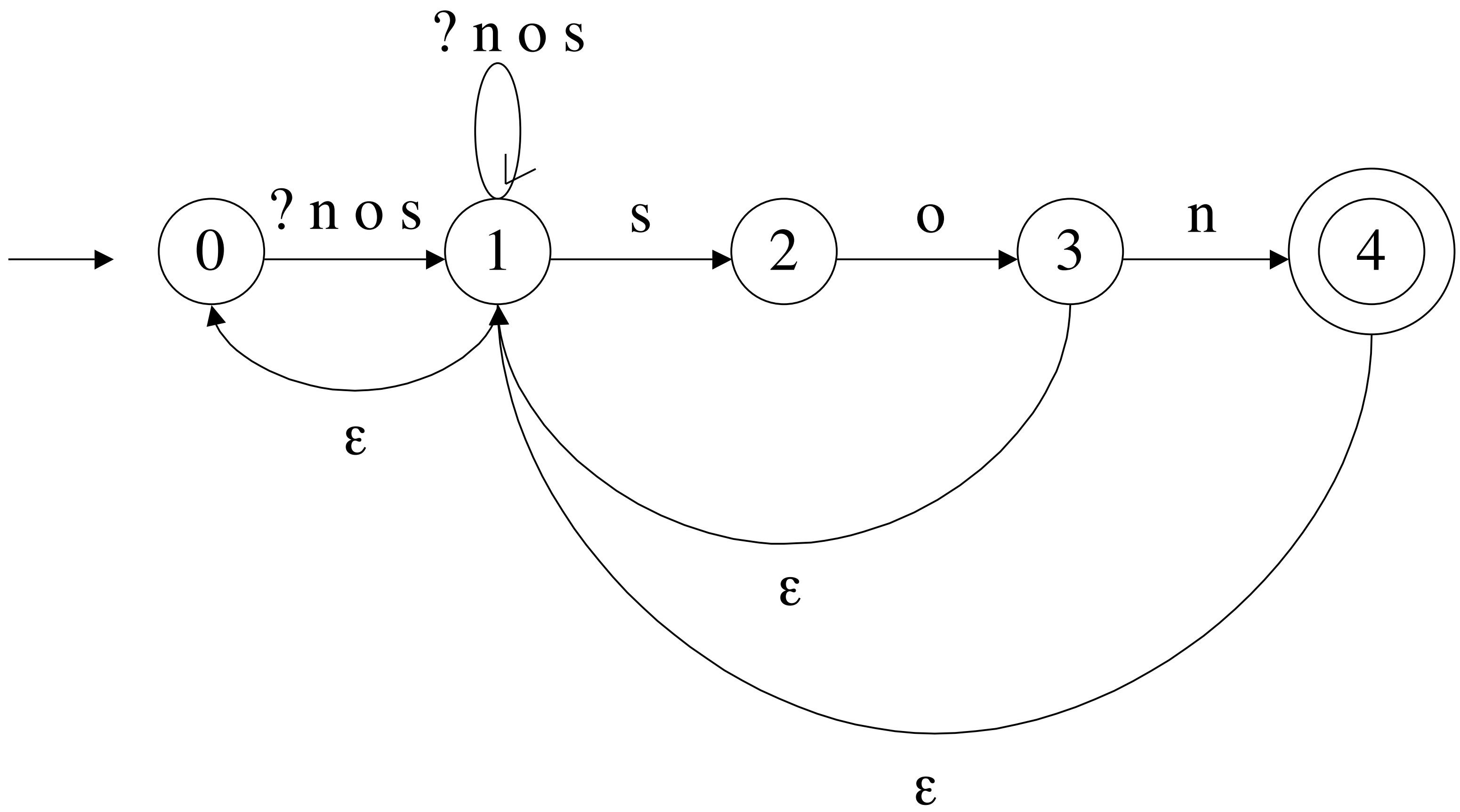
  δ = die Übergangsfunktion nimmt einen Zustand und ein Eingabesymbol aus $\Sigma \cup \{\varepsilon\}$ als Argument und gibt eine Teilmenge von Q zurück, also $Q \times (\Sigma \cup \{\varepsilon\}) \times 2^Q$

Beispiel ε-NEA: Übergangstabelle für δ

Input →	?	s	o	n	ε
State↓					
→ S0	{S1}	{S1}	{S1}	{S1}	∅
S1	{S1}	{S1, S2}	{S1}	{S1}	{S0}
S2	∅	{S3}	∅	∅	∅
S3	∅	∅	{S4}	∅	{S1}
*S4	∅	∅	∅	∅	{S1}

Frage: Wie sieht das Übergangsdiagramm aus?

Beispiel ε-NEA: Übergangsdiagramm



ε-Hülle

- Zum Analysieren von Eingaben braucht man die ε-Hülle.
- Definition von ε-Hülle:

  Basis: Zustand q ist in ε-Hülle(q) enthalten

  Induktionsschritt: Wenn der Zustand p in ε-Hülle(q) enthalten ist und es einen Übergang vom Zustand p zum Zustand r mit der Beschriftung ε gibt, dann ist r in ε-Hülle(q) enthalten

- Im vorigen Beispiel: ε-Hülle(S0) = {S0}; ε-Hülle (S1) = {S1, S2}

Erweiterte Übergangsfunktion

- Basis: $\delta\text{^}(q, \varepsilon) = \varepsilon\text{-Hülle}(q)$.
- Induktionsschritt: Angenommen, w ist eine Zeichenreihe xa, wobei a das letzte Symbol von w ist. Außerdem:

  $\delta\text{^}(q, x) = \{p_1, p_2, ..., p_k\}$

  $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, ..., r_m\}$

- Dann gilt: $\delta\text{^}(q, w) = \bigcup_{j=1}^{m} \varepsilon\text{-Hülle}\{r_j\}$.

Beispiel: Analysiere “larsson”

- Basis:

  $\delta\hat{}\ (S0, \varepsilon) = \varepsilon\text{-Hülle}(S0) = \{S0\}$

Beispiel: Analysiere “larsson”

- $\delta(S0, l) = \{S1\}$
- $\delta\char`^(S0, l) = \varepsilon\text{-Hülle}(S1) = \{S0, S1\}$
- $\delta(S0, a) \cup \delta(S1, a) = \{S1\} \cup \{S1\} = \{S1\}$
- $\delta\char`^(S0, la) = \varepsilon\text{-Hülle}(S1) = \{S0, S1\}$
- $\delta(S0, r) \cup \delta(S1, r) = \{S1\} \cup \{S1\} = \{S1\}$
- $\delta\char`^(S0, lar) = \varepsilon\text{-Hülle}(S1) = \{S0, S1\}$
- $\delta(S0, s) \cup \delta(S1, s) = \{S1\} \cup \{S1, S2\} = \{S1, S2\}$
- $\delta\char`^(S0, lars) = \varepsilon\text{-Hülle}(S1) \cup \varepsilon\text{-Hülle}(S2) = \{S0, S1\} \cup \{S2\} = \{S0, S1, S2\}$
- $\delta(S0, s) \cup \delta(S1, s) \cup \delta(S2, s) = \{S1\} \cup \{S1, S2\} \cup \emptyset = \{S1, S2\}$
- $\delta\char`^(S0, larss) = \varepsilon\text{-Hülle}(S1) \cup \varepsilon\text{-Hülle}(S2) = \{S0, S1\} \cup \{S2\} = \{S0, S1, S2\}$
- $\delta(S0, o) \cup \delta(S1, o) \cup \delta(S2, o) = \{S1\} \cup \{S1\} \cup \{S3\} = \{S1, S3\}$
- $\delta\char`^(S0, larsso) = \varepsilon\text{-Hülle}(S1) \cup \varepsilon\text{-Hülle}(S3) = \{S0, S1\} \cup \{S1, S3\} = \{S0, S1, S3\}$
- $\delta(S0, n) \cup \delta(S1, n) \cup \delta(S3, n) = \{S1\} \cup \{S1\} \cup \{S4\} = \{S1, S4\}$
- $\delta\char`^(S0, larsson) = \varepsilon\text{-Hülle}(S1) \cup \varepsilon\text{-Hülle}(S4) = \{S0, S1\} \cup \{S1, S4\} = \{S0, S1, S4\}$

Beispiel: Analysiere “larsson”

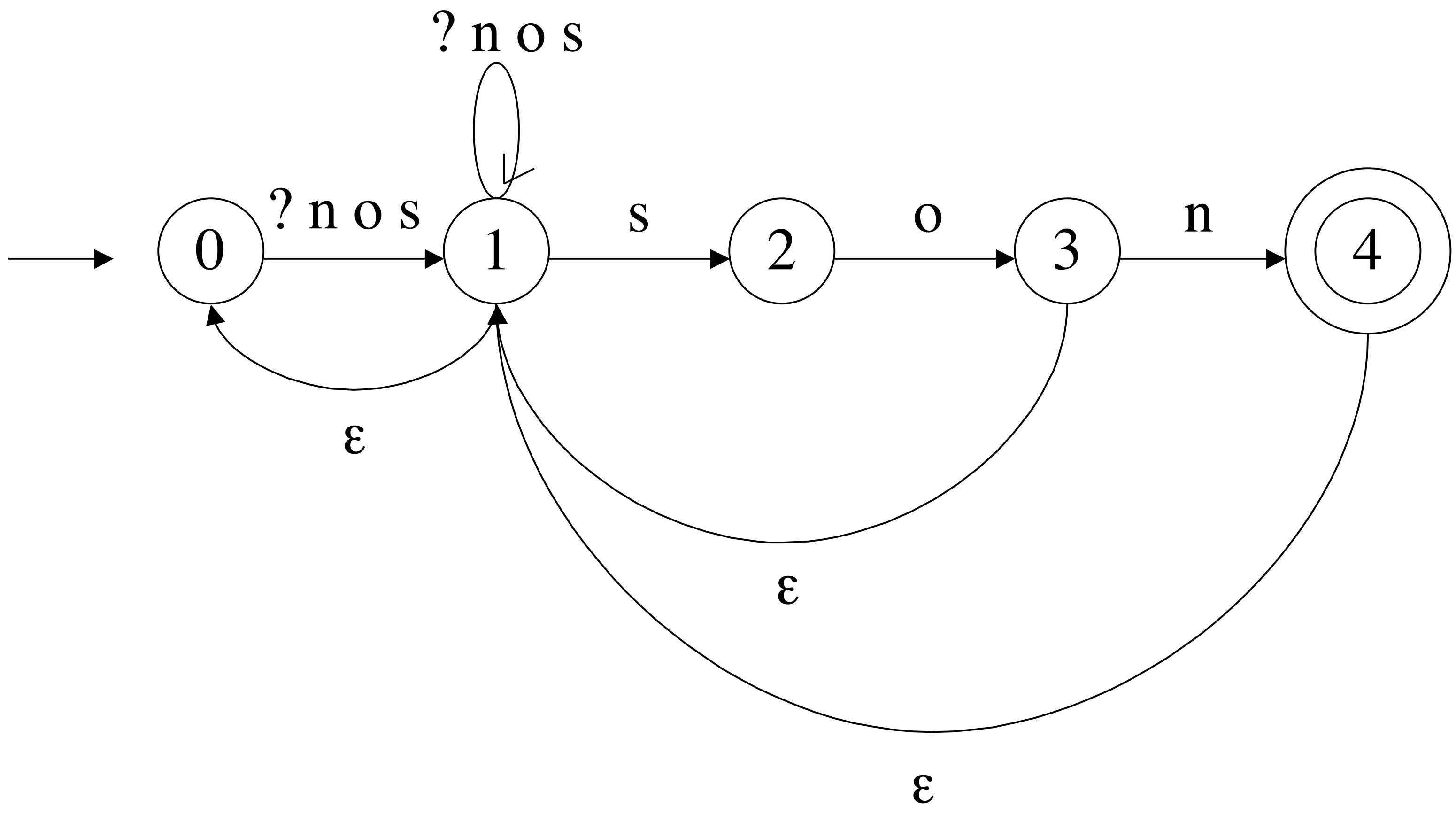
- Die Zeichenreihe wird akzeptiert, wenn $\{S4\} \subseteq F$ zutrifft.

Eliminierung von ε-Übergängen

Informelle Prozedur:

1. Bilde die ε-Hülle des Startzustandes.
2. Finde für jeden Zustand aus dieser ε-Hülle und für jedes mögliche Symbol den Folgezustand.
3. Bilde von diesen Folgezuständen die ε-Hülle.
4. Diese ε-Hüllen bilden die neuen Zustände des epsilonfreien NEA.
5. Gehe zu Schritt 2 (falls noch nicht der akzeptierende Zustand erreicht ist).

Beispiel. ε-NEA -> NEA



Beispiel: ε-NEA -> NEA

1. ε-Hülle des Startzustandes = ε-Hülle(S0) = {S0}.
   S0 ist also der **erste** Zustand des ε-freien NEA.
2. Für S0 finde jeden Folgezustand für jedes Symbol „?, s, o, n“.
   S0:? -> S1; S0:s -> S1; S0:o -> S1; S0:n -> S1
3. Bilde die ε-Hülle von S1 = {S0, S1}.
4. Dies ist der **zweite** Zustand des ε-freien NEA.
5. Gehe zu Schritt 2: Finde für S0 und S1 für jedes Symbol den Folgezustand.

Beispiel: ε-NEA -> NEA

S0, S1:? -> S1 -> ε-Hülle = {S0, S1} (bekannt)

S0, S1:s -> S1, S2 -> ε-Hülle = {S0, S1, S2}

S0, S1:o -> S1 -> ε-Hülle = {S0, S1} (bekannt)

S0, S1:n -> S1 -> ε-Hülle = {S0, S1} (bekannt)

{S0, S1, S2} ist der **dritte** Zustand des ε-freien NEA.

Gehe zu Schritt 2: Finde für {S0, S1, S2} und {S0, S1} für jedes Symbol den Folgezustand.

Beispiel: ε-NEA -> NEA

S0, S1, S2:? -> S1 -> ε-Hülle = {S0, S1} (bekannt)

S0, S1, S2:s -> S1, S2 -> ε-Hülle = {S0, S1, S2} (bekannt)

S0, S1, S2:o -> S1, S3 -> ε-Hülle = {S0, S1, S3}

S0, S1 ,S2:n -> S1 -> ε-Hülle = {S0, S1} (bekannt)

{S0, S1, S3} ist der **vierte** Zustand des ε-freien NEA.

Gehe zu Schritt 2 : Finde für {S0, S1, S2}, {S0, S1} und {S0, S1, S3} für jedes Symbol den Folgezustand.

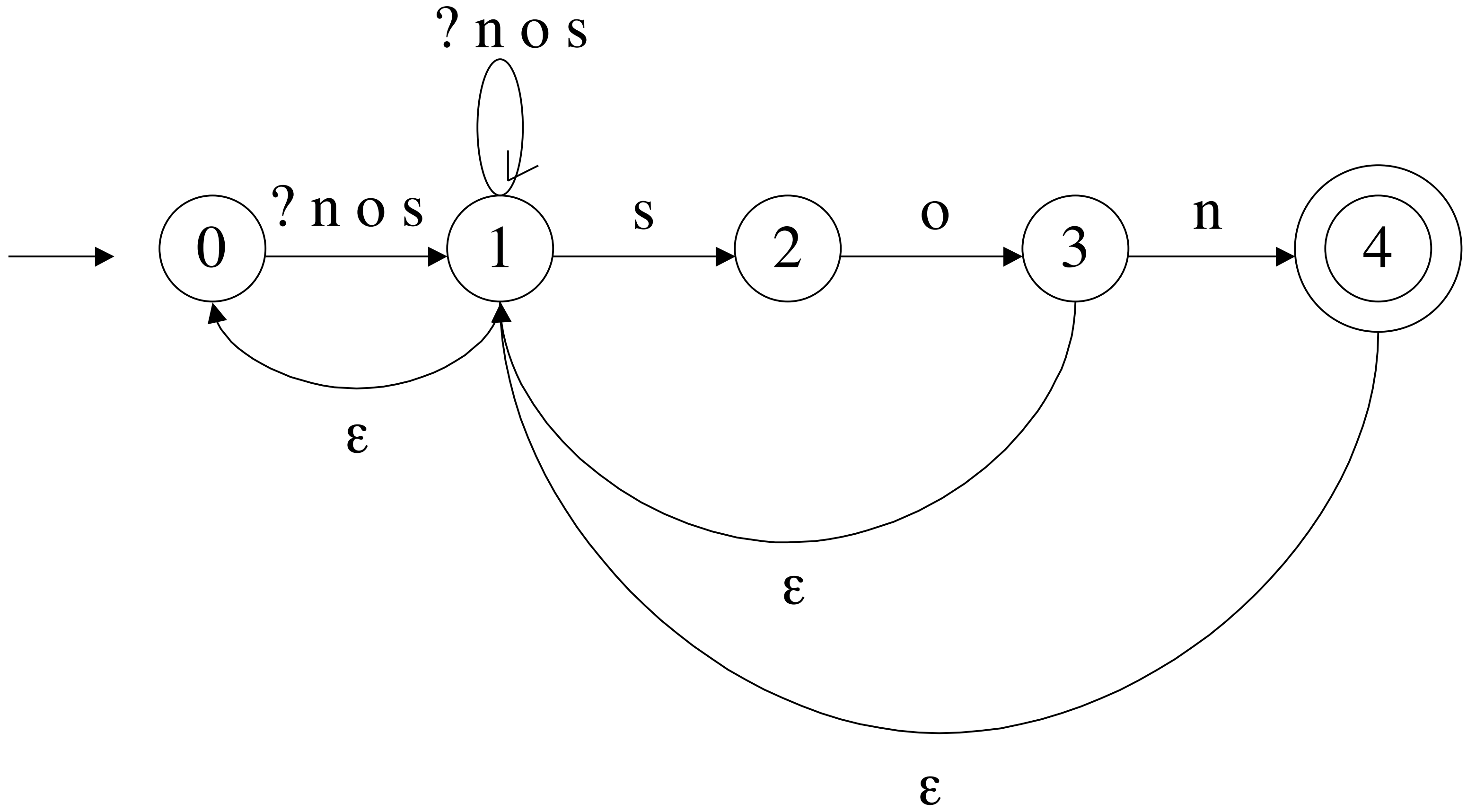
...

Beispiel: ε-NEA -> NEA

	?	s	o	n
-> 0	0,1	0,1	0,1	0,1
0,1	0,1	0,1,2	0,1	0,1
0,1	0,1	0,1,2	0,1	0,1
0,1,2	0,1	0,1,2	0,1,3	0,1
0,1	0,1	0,1,2	0,1	0,1
0,1,2	0,1	0,1,2	0,1,3	0,1
0,1,3	0,1	0,1,2	0,1	0,1,4
0,1	0,1	0,1,2	0,1	0,1
0,1,2	0,1	0,1,2	0,1,3	0,1
0,1,3	0,1	0,1,2	0,1	0,1,4
0,1,4	0,1	0,1,2	0,1	0,1

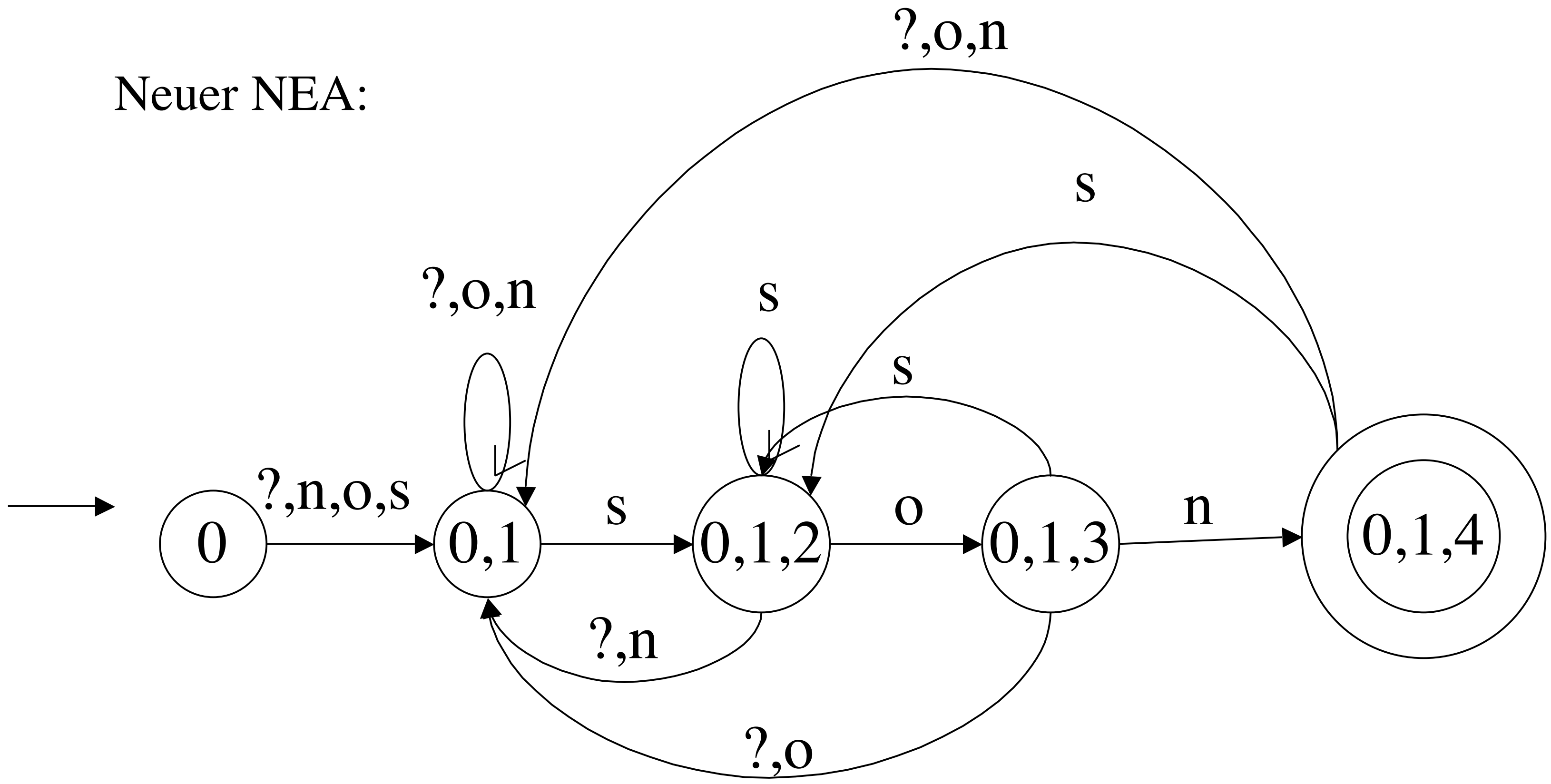
Beispiel: ε-NEA -> NEA

alter ε-NEA:



Beispiel: ε-NEA -> NEA

Neuer NEA:



Äquivalenz von RE’s und Automaten

- Wir sind jetzt in der Lage, RE‘s in Automaten umzuwandeln.
- Für jede Sprache L=L(R) für einen regulären Ausdruck R gilt: L = L(E) für einen ε-NEA E.

Äquivalenz von RE's und Automaten

- Wiederholung: Einfache reguläre Ausdrücke definieren reguläre Sprachen. Sie werden induktiv gebildet aus der Basis und drei Operationen:

  ε, $\emptyset$ sind reguläre Ausdrücke. $L(\varepsilon) = \{\varepsilon\}$, $L(\emptyset) = \emptyset$.

  Variablen (meist Großbuchstaben) repräsentieren ein Sprache.

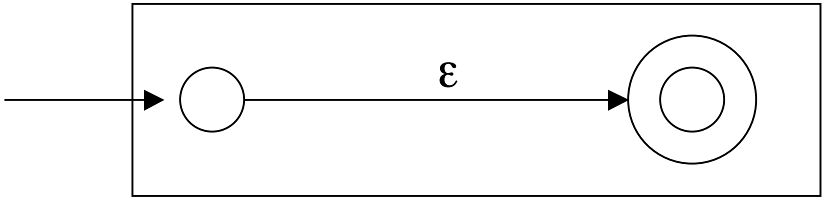
  Wenn a ein beliebiges Symbol ist, dann ist a ein RE: $L(a) = \{a\}$.

  E+F ist ein regulärer Ausdruck: $L(E + F) = L(E) \cup L(F)$.

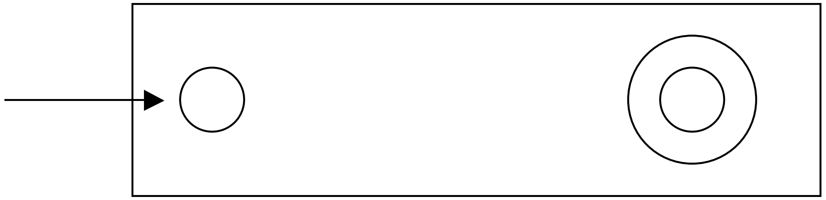
  EF ist ein regulärer Ausdruck: $L(EF) = L(E)L(F)$.

  Wenn E ein RE ist, dann ist E* ein RE: $L(E^*) = (L(E))^*$.

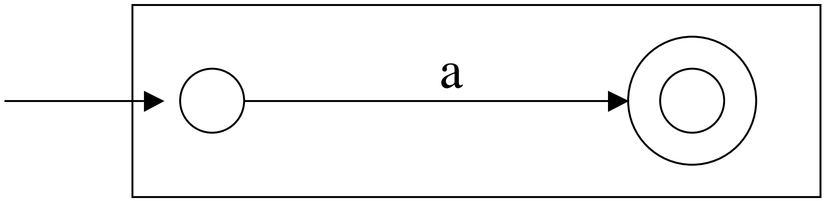
Äquivalenz von RE's und Automaten



L(ε)

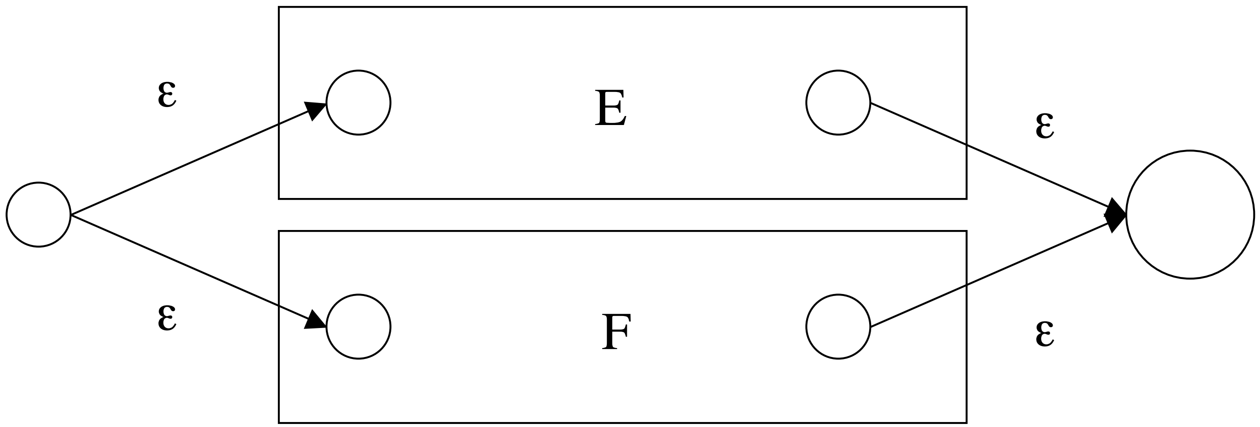


L(∅)



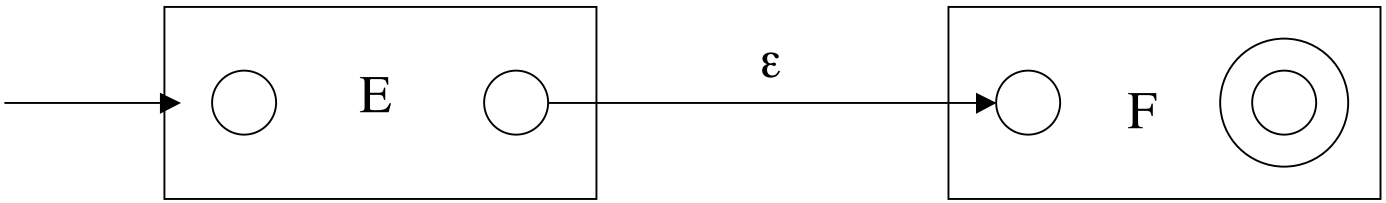
L(a)

Äquivalenz von RE's und Automaten



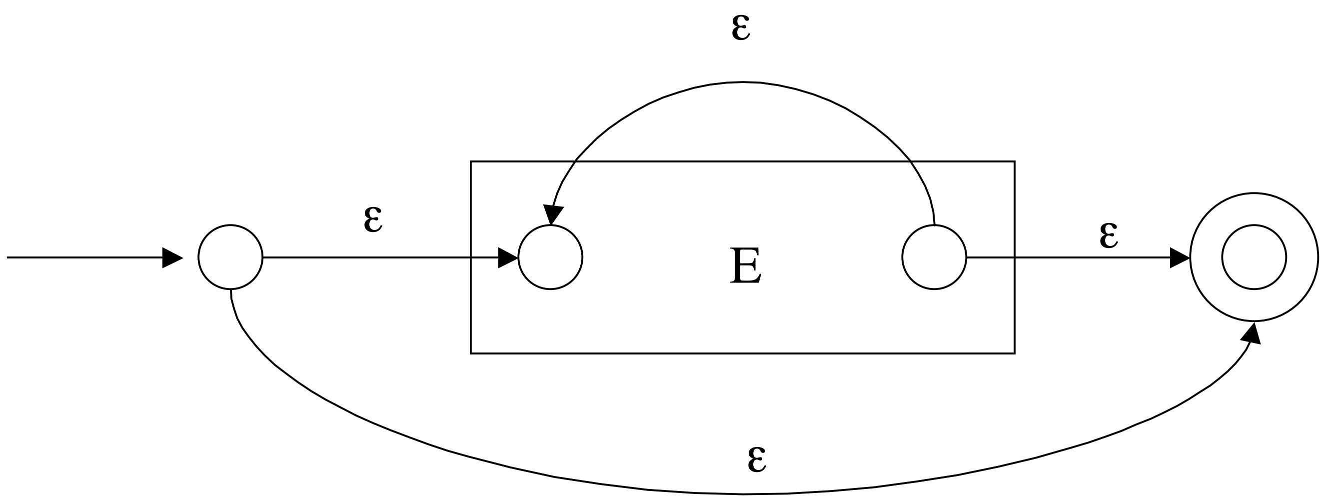
Beispiel Vereinigung: L(E +F) = L(E) ∪ L(F)

Äquivalenz von RE's und Automaten



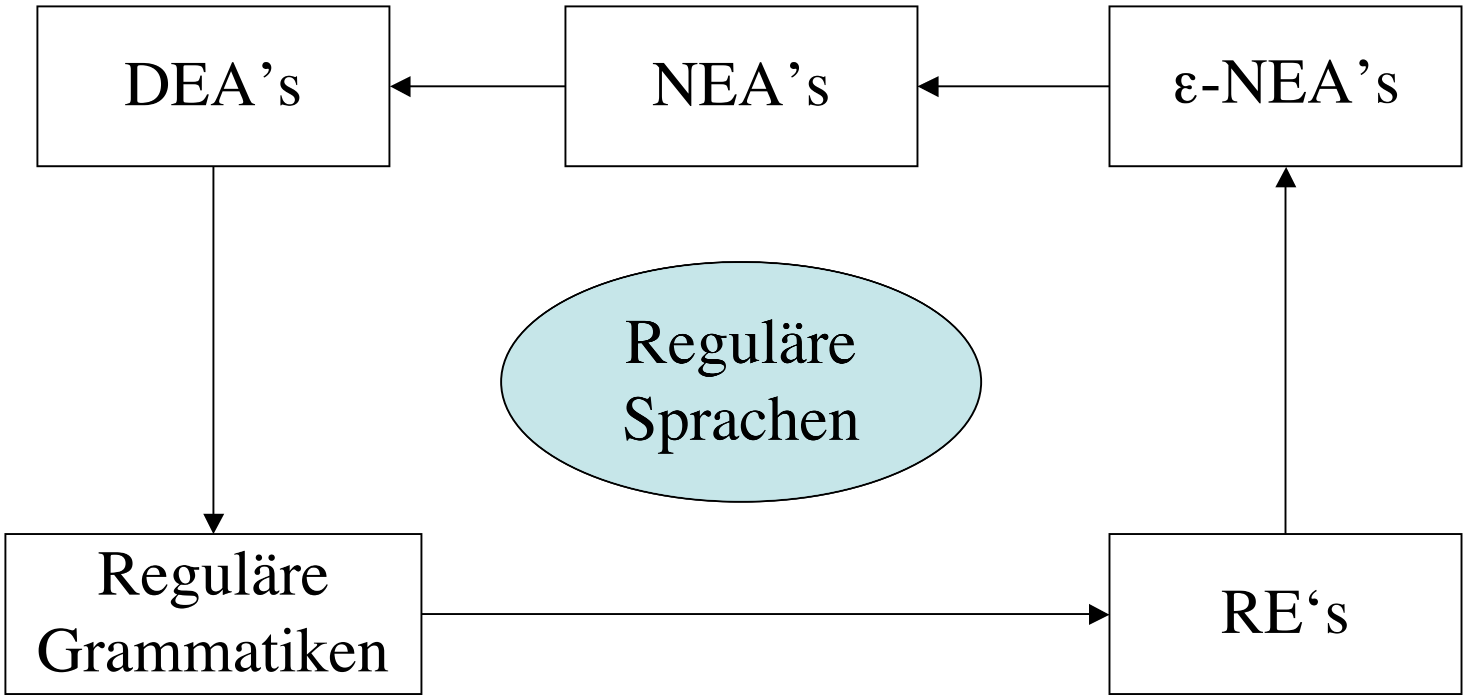
Beispiel Konkatenation: L(EF) = L(E)L(F).

Äquivalenz von RE's und Automaten



Beispiel Kleene-Iteration: $L(E)^* = (L(E))^*$

Notationen für reguläre Sprachen



Äquivalenz von Zuständen

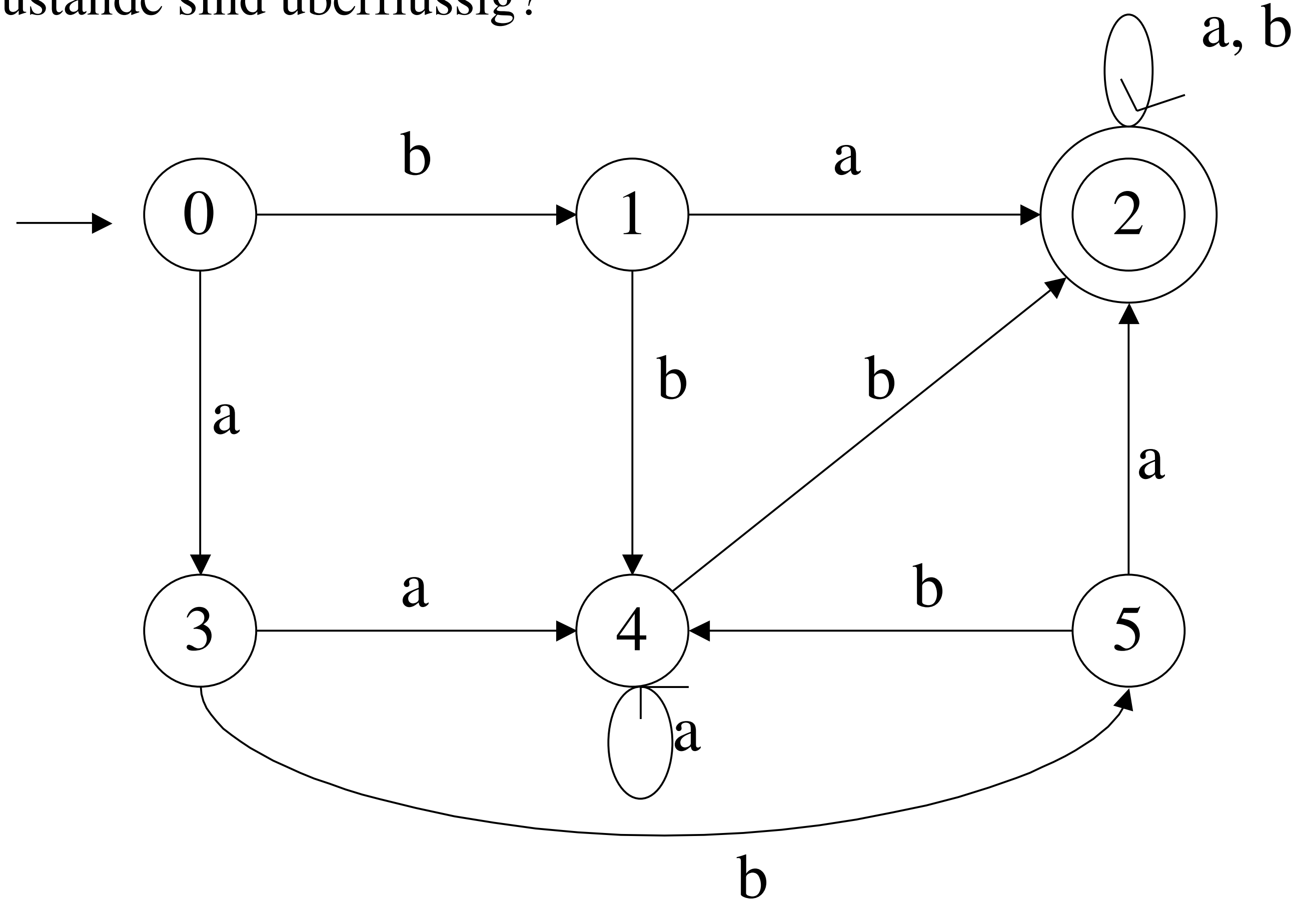
- Die Äquivalenz von Zuständen ist nötig für die Minimierung von Zuständen.
- Definition der Äquivalenz von Zuständen:

  Zwei Zustände p und q sind äquivalent, wenn für alle Zeichenreihen w gilt: $\hat{\delta}(p, w)$ ist ein akzeptierender Zustand genau dann, wenn $\hat{\delta}(q, w)$ auch ein akzeptierender Zustand ist.

- Mit anderen Worten: Wenn p ein akzeptierender Zustand ist und q nicht, dann ist das Paar {p, q} unterscheidbar.
- Dies kann man mit dem *table-filling*-Algortihmus veranschaulichen.

Äquivalenz von Zuständen: Beispiel

Welche Zustände sind überflüssig?



Äquivalenz von Zuständen: Beispiel

- Prozedur (informell) zum Ausfüllen der Äquivalenztabelle:

1. Notiere jedes Paar, das den akzeptierenden Zustand und einen nichtakzeptierenden Zustand beinhaltet.
2. Für die übriggebliebenen Paare: Teste jedes Paar mit jeder möglichen Zeichenreihe. Notiere es in der Tabelle, sobald eine der Zeichenreihen von einem Zustand zum akzeptierenden Zustand führt, zum anderen aber nicht.
3. Die leerstehenden Felder in der Tabelle geben die äquivalenten Zustände an.

Äquivalenz von Zuständen: Schritt 1

1					
2	X	X			
3			X		
4			X		
5			X		
	0	1	2	3	4

Äquivalenz von Zuständen: Schritt 2

1	a				
2	X	X			
3	ab	a	X		
4	b	b	X	b	
5	a		X	a	a
	0	1	2	3	4

Äquivalenz von Zuständen: Schritt 2

1	X				
2	X	X			
3	X	X	X		
4	X	X	X	X	
5	X		X	X	X
	0	1	2	3	4

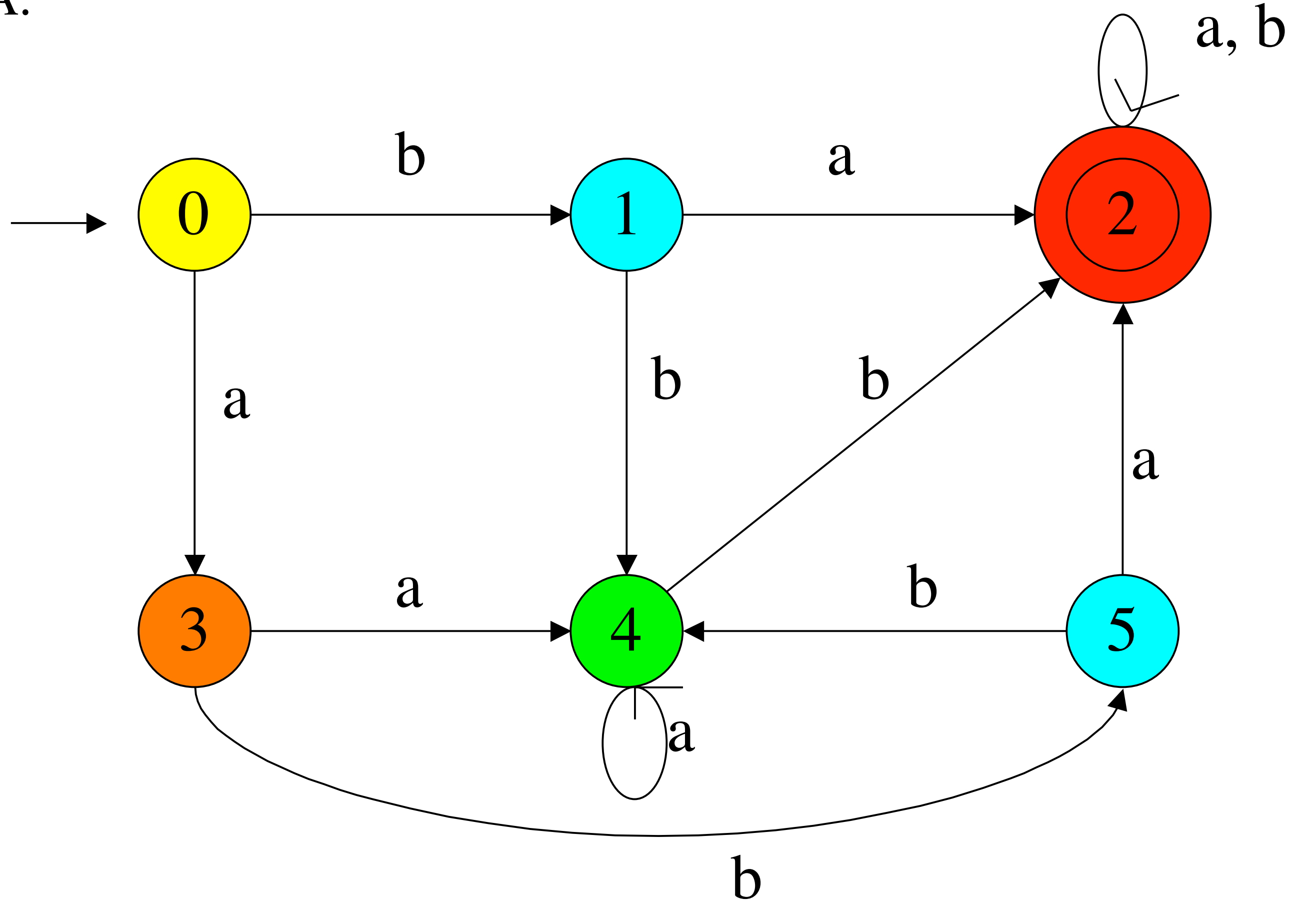
Minimierung von Zuständen: Beispiel

- Mit der Gruppierung von äquivalenten Zuständen in Blöcke ist der erste Schritt zur Minimierung von Zuständen getan.
- Die äquivalenten Zustände sind: {1, 5}. Diese bilden die ersten Blöcke.
- Alle anderen Zustände bilden eigene Blöcke: {0}, {2}, {3}, {4}.
- Der neue minimierte Automat hat also nur noch fünf Zustände:

  {0}, {1, 5}, {2}, {3}, {4}.

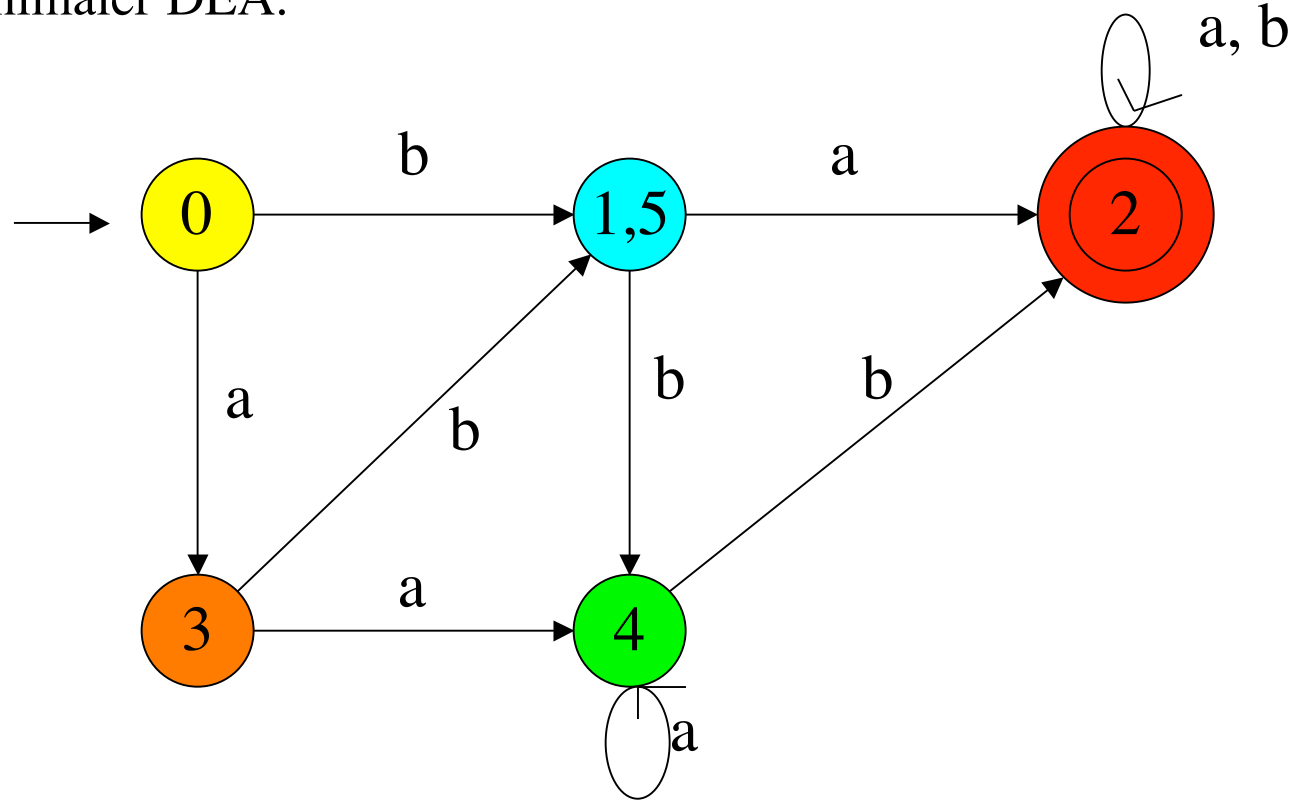
Äquivalenz von Zuständen: Beispiel

Alter DEA.



Äquivalenz von Zuständen: Beispiel

Neuer minimaler DEA.



Äquivalenz von Automaten

- Für jeden DEA gibt es einen äquivalenten DEA, der so wenige Zustände besitzt, wie kein anderer DEA, der dieselbe Sprache akzeptiert.
- Prozedur (informell):

  a) Nimm zwei Automaten und forme sie in DEA's um.

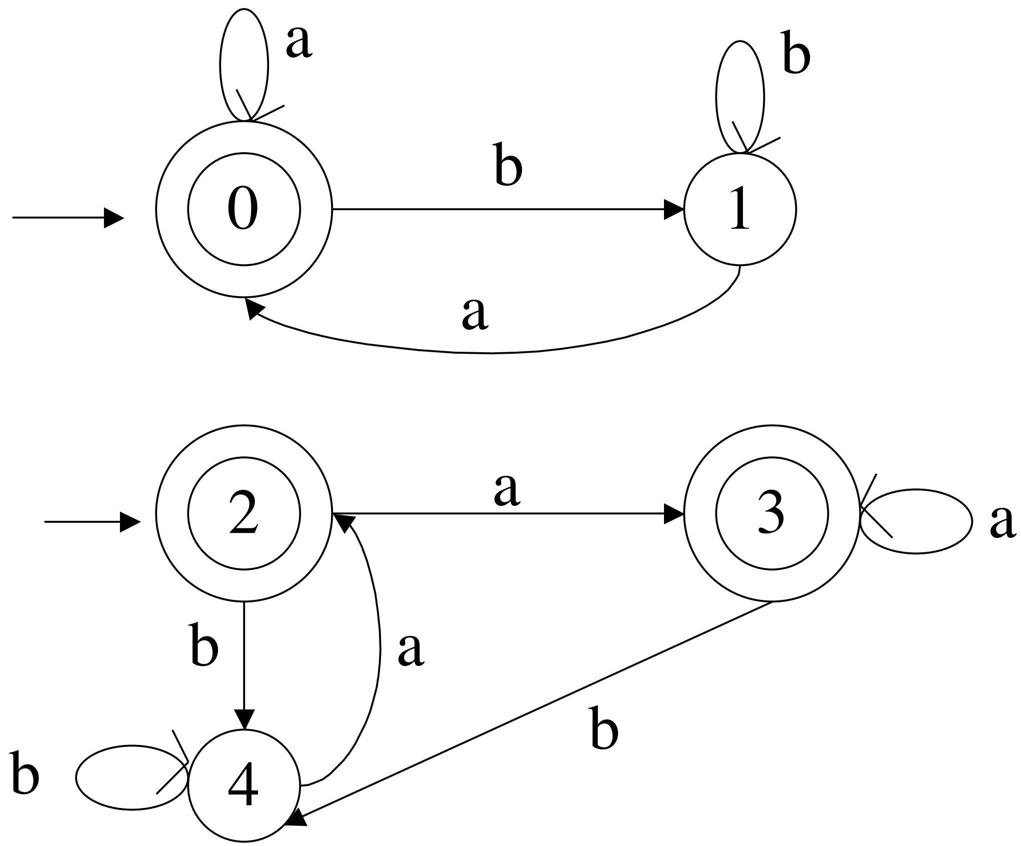
  b) Bilde die Vereinigung beider DEA's in einen einzelnen DEA.

  c) Bilde eine Äquivalenztabelle für diesen erweiterten DEA.

  d) Wenn die Startzustände der beiden kleineren Automaten in dem erweiterten Automaten äquivalent sind, sind beide Automaten auch äquivalent.

- Beispiel: Sind folgende Automaten äquivalent? Welche Sprache definieren sie?

Äquivalenz von Automaten: Beispiel



Äquivalenz von Automaten: Beispiel

1	X			
→2		X		
→3		X	X	
4	X		X	X
	→0	1	2	3

Äquivalenz von Automaten: Beispiel

- Die äquivalenten Zustände können in Blöcke gruppiert werden:
  {0, 2, 3}, {1, 4}.
- Die Anfangszustände der beiden Automaten waren 0, 2 und 3. Da sie äquivalent sind, sind auch die Automaten äquivalent.

Zusammenfassung

- Wir haben gesehen, wie

  a) RE’s in ε-NEA‘s umgewandelt werden
  b) ε-NEA‘s in NEA‘s umgewandelt werden
  c) NEA‘s in DEA‘s umgewandelt werden
  d) DEA‘s minimiert werden.

- Die Übergangstabellen von DEA’s können nun für einen Algorithmus zur Simulation von Automaten benutzt werden.

Literatur

- Beesley/Karttunen (2003:II-III)
- Jurafsky/Martin 2000:2.2
- Partee et al. 1993:17.1
- Roche/Schabes 1997:I
- Hopcroft/Motwani/Ullman (2001)