Zusammenfassung zur Vorlesung Basismodul Computerlinguistik

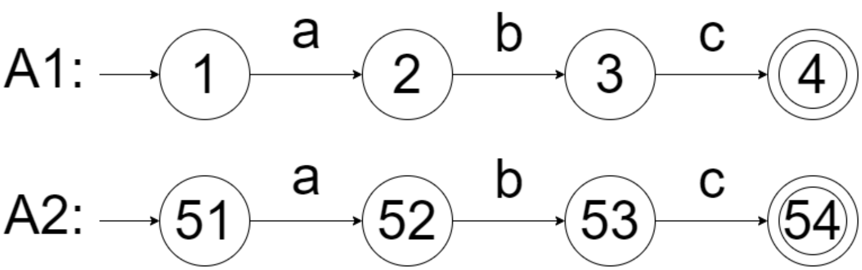
3 Methoden der Minimierung von DEA

18.11.2021

Inhaltsverzeichnis

DEA-Minimierung

- **Fakt:** Für jeden DEA existiert genau ein mDEA.
  mDEA ist eindeutig bis auf Isomorphie.



A1 $\neq$ A2, aber A1 und A2 sind gleich bis auf Isomorphie.

- **minimale DEAs = reduzierte DEAs**
- Wie wird der mDEA bestimmt?

1. Allgemeines Verfahren zur Minimierung beliebiger DEAs

Allgemeine Konstruktion

Ausgang von DEA

Ein DEA wird minimiert, gdw.

- Es existieren keine Zustände mehr, die vom Startzustand nicht erreichbar sind.
- Es existieren keine äquivalenten Zustände mehr.

Voraussetzung: Die Methode funktioniert **nur** für DEAs mit der **totalen** Übergangsfunktion.

Verfahren zur Konstruktion:

1. Zustände löschen, die nicht vom Startzustand erreichbar sind.
2. Induktive Berechnung der äquivalenten Zuständen oder äquivalente Klassen.

k-Ununterscheidbarkeit

Definition: k-Ununterscheidbarkeit
Zwei Zustände $p, q \in Q$ werden als *k-ununterscheidbar* $(p \sim_k q)$ bezeichnet (für $k \geq 0$), gdw. gilt:

$$\forall w \in \Sigma^*, |w| \leq k : \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F$$

Induktive Berechnung der k-Ununterscheidbarkeit

- $p \sim_0 q \Leftrightarrow p, q \in F$ oder $p, q \notin F$
- $p \sim_{i+1} q \Leftrightarrow p \sim_i q$ und $\delta(p, \sigma) \sim_i \delta(q, \sigma) \forall \sigma \in \Sigma$

Definition: Äquivalenz / Ununterscheidbarkeit
Zwei Zustände $p, q \in Q$ heißen *äquivalent* oder *ununterscheidbar*, wenn sie für jedes $k \in \mathbb{N}$ ununterscheidbar sind, also falls gilt:

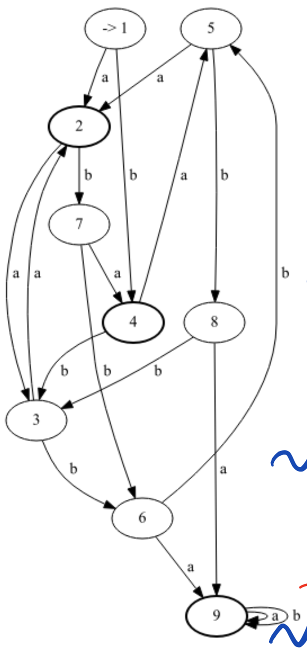
$$\forall w \in \Sigma^* : \delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w)$$

Aufgabe

Aufgabe 1.3 Gegeben sei der deterministische Automat (mit vollständiger Übergangsfunktion)

$$A = (\{a, b\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, 1, \{2, 4, 9\}, \delta),$$

die Übergangsfunktion δ soll die Paare $((1,a),2)$, $((1,b),4)$, $((2,a),3)$, $((2,b),7)$, $((3,a),2)$, $((3,b),6)$, $((4,a),5)$, $((4,b),3)$, $((5,a),2)$,$((5,b),8)$, $((6,a),9)$, $((6,b),5)$, $((7,a),4)$, $((7,b),6)$, $((8,a),9)$, $((8,b),3)$, $((9,a),9)$, $((9,b),9)$ enthalten. Berechnen Sie einen äquivalenten minimalen deterministischen Automaten.



	F			NF					
	2	4	9	1	3	5	6	7	8
	a:3	a:5	a:9	a:2	a:2	a:2	a:9	a:4	a:9
	b:7	b:3	b:9	b:4	b:6	b:8	b:5	b:6	b:3
$\sim_0$	1	1	1	2	2	2	2	2	2
	122	122	111	211	212	212	212	212	212
$\sim_1$	1	1	2	3	4	4	4	4	4
	144	144	222	311	414	414	424	414	424
$\sim_2$	1	1	2	3	4	4	5	4	5

gleich

144 144 222 311 415 415 524 415 524

N_3 1 1 2 3 4 4 5 4 5

Fertig!

$F = \{1,2\}$ $s = 3$ $Q = \{1,2,3,4,5\}$

$\delta = \{(1,a,4)\ (1,b,4)\ (2,a,2)\ (2,b,2)\ (3,a,1)$
$(3,b,1)\ (4,b,5)\ (4,a,1)\ (5,a,2)\ (5,b,4)\}$

2. Nerode-Konstruktion

$\Delta = \{\Sigma, Q, F, s, \delta\}$

$\Sigma = \{\ \}$
$Q = \{\ \}$

...

Nerode-Äquivalenzrelation

- Anders als die allgemeine Konstruktion ist bei Nerode Konstruktion der **Ausgang die Sprache** L, d.h. der minDEA wird von einer Sprache abgeleitet.

Definition: Nerode-Äquivalenzrelation $\sim_L$

Sei L eine beliebige Sprache über dem Alphabet Σ, wird die *Nerode-Äquivalenz* $\sim_L$ wie folgt definiert:

$$\forall u, v \in \Sigma^*, u \sim_L v :\Leftrightarrow (\forall w \in \Sigma^* : u \cdot w \in L \text{ gdw. } v \cdot w \in L)$$

d.h. die Rechtkontexte / Rechtssprachen von zwei Wörtern sind gleich.

Äquivalenzklassen

Eine Äquivalenzklasse $[w]_{\sim_L}$ enthält alle Wörter, die mit dem Wort w Nerode-Äquivalenzrelation haben.

Nerode-Konstruktion eines mDEA

Der mDEA A für eine reguläre Sprache L wird mittels Äquivalenzklassen wie folgt aufgebaut:

$$A = (\sigma^*/\sim_L, \Sigma, [\epsilon]_L, \{[w]_{\sim_L} | w \in L\}, \delta)$$

mit

- $\sigma^*/\sim_L$ Menge der Äquivalenzklassen von $\sim_L$
- $[\epsilon]_L$ Äquivalenzklasse von ϵ
- $[w]_{\sim_L} | w \in L\}$ Äquivalenzklassen der Wörter aus L

Beispiel

Sei $L = a^* b^*$ eine reguläre Sprache über $\Sigma = \{a, b\}$, berechne den minimalen DEA der Sprache L.

$L = a^* b^*$

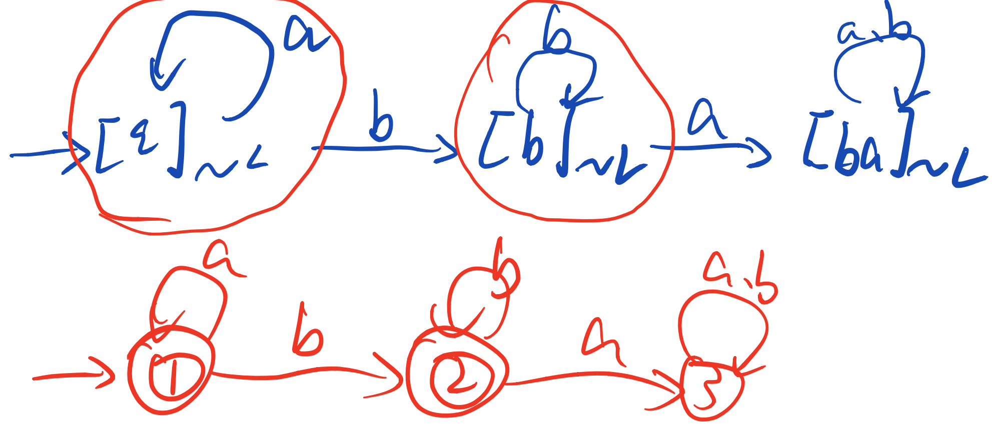
$\Rightarrow$ alle Wörter über Σ^*, die 'ba' als Teilsequenz nicht enthalten.

$= \{\varepsilon, a, aa, aaa, \dots, \mid b, bb, \dots, \mid ab, aab, \dots\}$

① $[a]_{\sim L} = \{\varepsilon, a, aa, \dots\}$ Suffixe $= \{\varepsilon, a, aa, \dots, ab, aab, b, bb, \dots\}$

② $[b]_{\sim L} = \{b, ab, aab, \dots, bb, bbb, \dots\}$ Suffixe $= \{\varepsilon, b, bb, \dots\}$

③ $[ba]_{\sim L} = \{ba, aba, aaba, \dots\}$ Suffixe $= \emptyset$


[ε]~L
a
b
[b]~L
b
a
[ba]~L
a,b
1
a
b
2
b
a
3
a,b

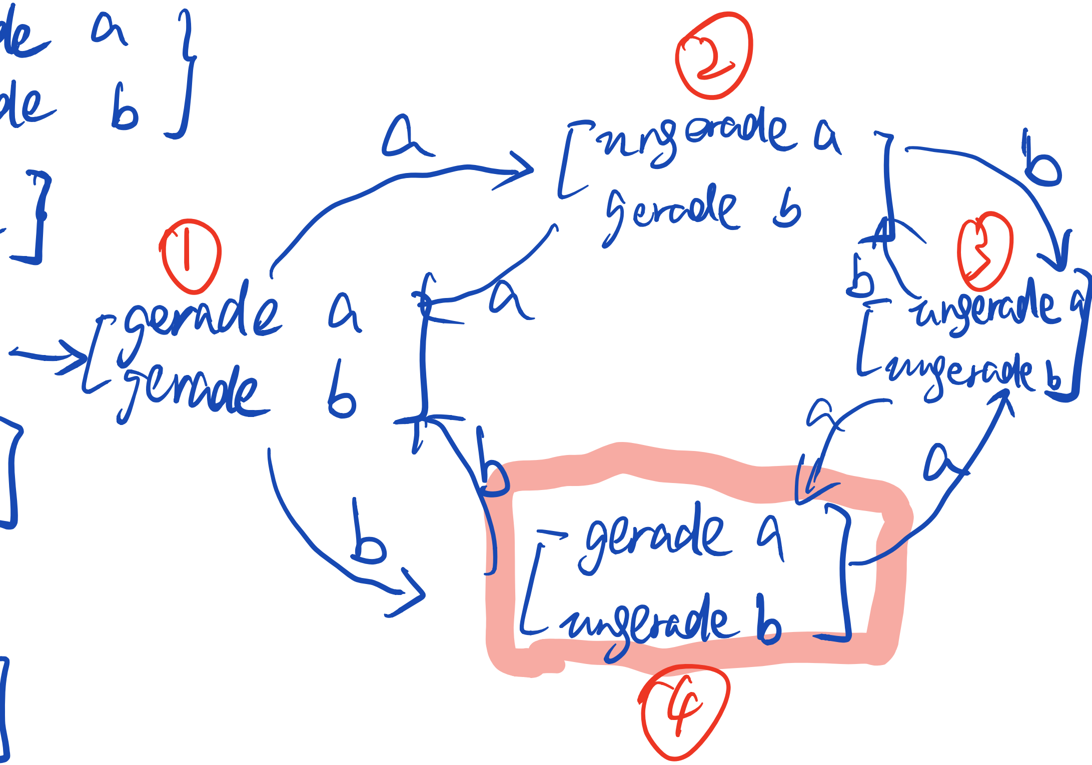
Aufgabe 1.5 Es sei $\Sigma := \{a, b\}$ und L die Menge aller Wörter aus Σ mit einer geraden Zahl von Vorkommen von a und einer ungeraden Zahl von Vorkommen von b. Wie sieht die Nerode-Äquivalenzrelation für L aus? Verwenden Sie diese, um direkt den minimalen deterministischen Automaten für L anzugeben.

$\to [\varepsilon] = \{$ gerade a, gerade b $\}$

[ungerade a, gerade b]

[gerade a, ungerade b]

[ungerade a, ungerade b]



Minimierung von Tries

thee

Minimierung eines vorhandenen Tries

Ausgang: Lexikon

Trie (Präfixbaum, Lexikonautomat): ein azyklischer DEA

- geeignet für die Darstellung und Speicherung von Lexika

Wann sind 2 Zustände äquivalent?

Wenn sie

- beide final oder nicht-final sind
- die gleichen ausgehenden Übergänge haben (Anzahl, label)
- zu äquivalenten Zuständen gehen (für die wiederum die obigen Bedingungen gelten).

Achtung: Die Konstruktion eines minimierten Tries muss anhand einer **sortierten** Wortliste durchgeführt werden.

Beispiel

Gegeben sei das Lexikon mit den sortieren Wörtern *abladen*, *ablauf*, *abgelaufen*, *lauf*, *laufen*, berechne den minimalen DEA für das Lexikon.

Hausaufgabe

- Implementierung von Daciuk-Algorithmus
- Zwei Wochen eingeplant, Abgabe bis 02.12.2021
- `https://www.cip.ifi.lmu.de/~nie/p1/HA_05.html`