Formale Sprachen und Komplexität
Theoretische Informatik für Medieninformatiker
Sommersemester 2022

Minimierung von Deterministischen Endlichen Automaten

Prof. Dr. David Sabel

LFE Theoretische Informatik

Letzte Änderung der Folien: 10. Juni 2022

Äquivalenzklassenautomat

Definition (Äquivalenzklassenautomat)

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Wir nennen zwei Zustände $z, z' \in Z$ äquivalent und schreiben $z \equiv z'$ falls gilt: für alle $w \in \Sigma^*$: $\widehat{\delta}(z, w) \in E \iff \widehat{\delta}(z', w) \in E$. Der Äquivalenzklassenautomat zu M ist der DFA $M' = (Z', \Sigma, \delta', z_0', E')$ mit

$$\begin{aligned} Z' &= \{[z]_\equiv \mid z \in Z\}, \\ z_0' &= [z_0]_\equiv, \\ E' &= \{[z]_\equiv \mid z \in E\} \text{ und} \\ \delta'([z]_\equiv, a) &= [\delta(z, a)]_\equiv. \end{aligned}$$

Äquivalenzklassenautomat: Korrektheit und Minimalität

Satz

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA und $M' = (Z', \Sigma, \delta', z_0', E')$ der Äquivalenzklassenautomat zu M. Dann gilt

1. $L(M) = L(M')$
2. Falls alle Zustände in Z vom Startzustand z_0 erreichbar sind, dann ist M' minimal.

Beweis (nur Teil 1):

Sei $w \in \Sigma^*$. Dann gilt:

- M durchläuft die Zustandsfolge $q_0, \ldots, q_{|w|}$ entlang w und akzeptiert w genau dann, wenn $q_{|w|} \in E$ gilt.
- M' durchläuft die Zustandsfolge $[q_0]_\equiv, \ldots, [q_{|w|}]_\equiv$ und akzeptiert w genau dann, wenn $[q_{|w|}]_\equiv \in E'$ gilt.

Da per Definition $[q_{|w|}]_\equiv \in E'$ genau dann gilt, wenn $q_{|w|} \in E$ gilt, folgt, dass M und M' dieselben Worte akzeptieren.

Zustandsminimierung von DFAs

Idee:

- Entferne nicht-erreichbare Zustände
- Berechne äquivalente Zustände (bezüglich $\equiv$)
- Bilde Äquivalenzklassenautomat, indem äquivalente Zustände verschmolzen werden.

Berechnung äquivalenter Zustände

Ideen

- Markiere Paare von Zuständen, die verschieden sein müssen
  markiere initial: alle $\{z, z'\}$ mit $z \in E$, $z' \notin E$.
- Vervollständige das Markieren durch Untersuchen von Übergängen:
  - Wenn $\{z, z'\}$ noch nicht markiert:
    Prüfe für jedes $a \in \Sigma$, ob die beiden Nachfolger $\{\delta(z, a), \delta(z', a)\}$ markiert sind.
  - Falls ja, dann markiere $\{z, z'\}$.
  - Wiederhole, bis sich nichts mehr ändert.
- Alle am Ende unmarkierten Paare sind äquivalente Zustände..

Algorithmus 3: Berechnung aller äquivalenten Zustände

Eingabe: DFA $M = (Z, \Sigma, \delta, z_0, E)$, der keine unerreichbaren Zustände hat
Ausgabe: Zustandspaare $\{z, z'\}$ mit $z \neq z'$ für die gilt $z \equiv z'$
Beginn
    stelle Tabelle T aller Zustandspaare $\{z, z'\}$ mit $z \neq z'$ und $z, z' \in Z$ auf;
    markiere alle Paare $\{z, z'\}$ in T mit $z \in E$ und $z' \notin E$;
    wiederhole
        für jedes *unmarkierte Paar* $\{z, z'\}$ *in* T **tue**
            für jedes $a \in \Sigma$ **tue**
                wenn $\{\delta(z, a), \delta(z', a)\}$ *in* T *markiert ist* **dann**
                    markiere $\{z, z'\}$ in T;
                Ende
            Ende
        Ende
    bis *sich* T *nicht mehr verändert*;
    return $\{\{z, z'\} \mid \{z, z'\}$ *ist nicht markiert in* $T\}$
Ende

Korrektheit

Satz

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der keine unerreichbaren Zustände hat. Algorithmus 3 berechnet äquivalente Zustandspaare von M und es gibt keine weiteren äquivalenten Paare.

Beweis (Teil 1): Wird $\{z, z'\}$ markiert, dann gilt $z \not\equiv z'$.

- Wir zeigen: Für jedes markierte Paar $\{z, z'\}$ gibt es Wort w mit $\neg(\widehat{\delta}(z, w) \in E \iff \widehat{\delta}(z', w) \in E)$
- Induktion über Anzahl Schleifeniterationen bis $\{z, z'\}$ markiert wird.
- Basis: 0 Iterationen, $\{z, z'\}$ wird vor der Schleife markiert, $w = \varepsilon$ erfüllt Behauptung.
- Schritt: Mehr als 0 Iterationen. Dann wird $\{z, z'\}$ markiert, weil es $a \in \Sigma$ und ein markiertes Paar $\{q, q'\}$ gibt mit $\delta(z, a) = q$ und $\delta(z', a) = q'$. Induktionsannahme liefert Wort w' mit $\neg(\widehat{\delta}(q, w') \in E \iff \widehat{\delta}(q', w') \in E)$.
- Mit $w = aw'$ folgt: $\neg(\widehat{\delta}(z, w) \in E \iff \widehat{\delta}(z', w) \in E)$.

Korrektheit (2)

Satz

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA, der keine unerreichbaren Zustände hat. Algorithmus 3 berechnet äquivalente Zustandspaare von M und es gibt keine weiteren äquivalenten Paare.

Beweis (Teil 2): Wenn $z \not\equiv z'$, dann markiert Alg. 3 das Paar $\{z, z'\}$

- Beweis durch Widerspruch. Annahme es gibt Paare $z \not\equiv z'$, die Alg. 3 nicht markiert. Wähle Paar $\{z, z'\}$, für welches es ein minimal langes Wort w gibt mit $\neg(\widehat{\delta}(z, w) \in E \iff \widehat{\delta}(z', w) \in E)$.
- Wenn $w = \varepsilon$, dann wird $\{z, z'\}$ vor Schleife markiert. Widerspruch!
- Wenn $w = aw'$ mit $a \in \Sigma$, dann gilt: Wenn $\{\delta(z, a), \delta(z', a)\}$ vom Algorithmus markiert wird, dann auch $\{z, z'\}$. Daher: $\{\delta(z, a), \delta(z', a)\}$ wird nicht markiert. Aber dann gilt für w': $\neg\left(\widehat{\delta}(\delta(z, a), w') \in E \iff \widehat{\delta}(\delta(z', a'), w') \in E\right)$ und $|w'| < |w|$.
  Widerspruch zur Minimalität!

Laufzeit

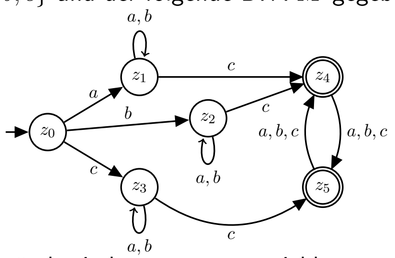
- Darstellung der Tabelle T:
  zweidimensionales Array der Größe $|Z| \times |Z|$
- Ermöglicht konstanten Zugriff auf Markierungen
- Pro Durchlauf der Schleife: $O(|Z|^2|\Sigma|)$
- Anzahl der Durchläufe ist durch $|Z|^2$ begrenzt, da es nur $|Z|^2$ Paare gibt und mindestens 1 Paar pro Durchlauf markiert wird
- Restliche Schritte: Konstante Laufzeit
- Daher: Algorithmus 3 kann in Zeit $O(|Z|^4 \cdot |\Sigma|)$ implementiert werden
- Tatsächlich gibt es effizientere Implementierungen: $O(|\Sigma| \cdot |Z| \log |Z|)$

Algorithmus 4: Minimierung von DFAs

Eingabe: DFA $M = (Z, \Sigma, \delta, z_0, E)$
Ausgabe: Minimaler DFA M' mit $L(M) = L(M')$
Beginn
  entferne Zustände aus M, die nicht vom Startzustand aus erreichbar sind;
  berechne äquivalente Zustände mit Algorithmus 3;
  erzeuge den Äquivalenzklassenautomat, indem die berechneten äquivalenten Zustände vereinigt werden;
Ende

Beispiel zur Minimierung

Sei $\Sigma = \{a, b, c\}$ und der folgende DFA M gegeben:



z_1	X				
z_2	X				
z_3	X				
z_4	X	X	X	X	
z_5	X	X	X	X	
	z_0	z_1	z_2	z_3	z_4

- alle Zustände sind von z_0 aus erreichbar
- äquivalente Zustände berechnen: Tabelle T erstellen
- Initiales Markieren: $\{z, z'\}$ mit $z \in \{z_0, z_1, z_2, z_3\}$ und $z' \in \{z_4, z_5\}$
- $\{z_0, z_1\}$, da $\{\delta(z_0, c), \delta(z_1, c)\} = \{z_3, z_4\}$ bereits markiert ist,
- $\{z_0, z_2\}$, da $\{\delta(z_0, c), \delta(z_2, c)\} = \{z_3, z_4\}$ bereits markiert ist, und
- $\{z_0, z_3\}$, da $\{\delta(z_0, c), \delta(z_3, c)\} = \{z_3, z_5\}$ bereits markiert

Beispiel zur Minimierung (2)

Ergibt $z_1 \equiv z_3$, $z_1 \equiv z_2$, $z_2 \equiv z_3$, $z_4 \equiv z_5$ und daher die Äquivalenzklassen

$$[z_0]_\equiv = \{z_0\},\ [z_1]_\equiv = \{z_1, z_2, z_3\},\ z_{4\equiv} = \{z_4, z_5\}$$

und den Minimalautomaten