



Theoretische Informatik – Übung Gruppe 4

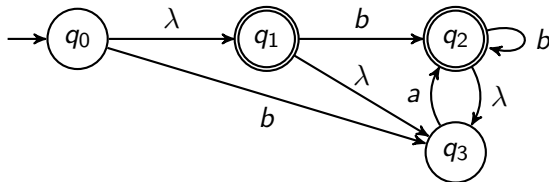
Roman Langrehr

Varianten Nichtdeterministischer endlicher Automaten

λ -NEAs: NEA bei dem auch Übergänge ohne ein Zeichen zu lesen erlaubt sind
($\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow \mathcal{P}(Q)$).

Aufgabe

1. Gebe ein Verfahren an, mit dem man jeden beliebigen λ -NEA in einen äquivalenten NEA umwandeln kann.
2. Wende das Verfahren auf folgenden NEA an:

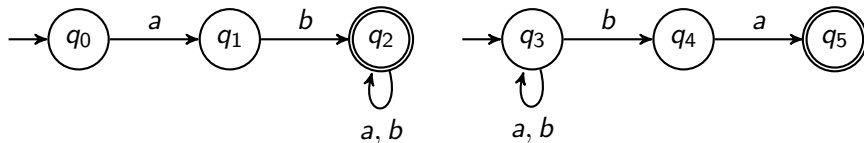


Varianten Nichtdeterministischer endlicher Automaten

NEAs mit mehreren Startzuständen: Akzeptieren ein Wort w wenn ein Startzustand *existiert* von dem aus der NEA das w akzeptiert.

Aufgabe

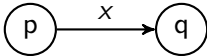
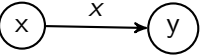
1. Gebe ein Verfahren an, mit dem man jeden beliebigen NEA mit mehreren Zuständen in einen äquivalenten λ -NEA umwandeln kann.
2. Wende das Verfahren auf folgenden NEA mit mehren Startzuständen an:



Potenzmengenkonstruktion

Theorem

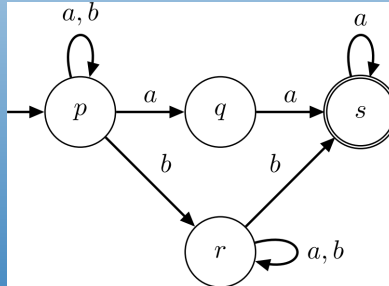
Zu jedem NEA M existiert ein EA A , so dass $L(M) = L(A)$.

- A hat Zustandsmenge $\mathcal{P}(Q)$ (Q : Zustandsmenge von M).
- A hat Übergang  genau dann wenn $\exists x \in p, \exists y \in q$ so dass M einen Übergang  hat.
- $p \in \mathcal{P}(Q)$ ist akzeptierend, genau dann wenn ein $x \in p$ existiert dass in M akzeptierend ist.

Potenzmengenkonstruktion

Aufgabe

Wende die Potenzmengenkonstruktion auf folgenden NEA an. Nicht erreichbare Zustände können weggelassen werden.



Potenzmengenkonstruktion

- Wenn ein NEA M n Zustände hat, hat der per Potenzmengenkonstruktion konstruierte EA 2^n Zustände.
- Manchmal ist das exponentielle Wachstum unvermeidbar (Buch: $L_k := \{x1y \mid x \in \{0,1\}^*, y \in \{0,1\}^{(k-1)}\}$, es gibt NEA mit $k+1$ Zuständen, EA braucht mindestens 2^k).
- Manchmal nicht (z.B. wenn der NEA M bereits deterministisch ist.)

Potenzmengenkonstruktion

Aufgabe

Wie kann man zu einem NEA M den *kleinsten* äquivalenten EA A effizient (Laufzeit polynomiell in der Anzahl Zustände von M und A) finden?

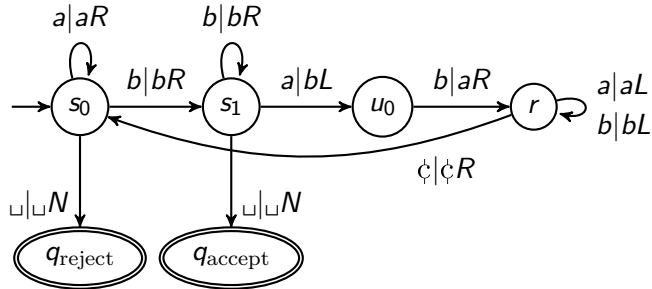
- Potenzmengenkonstruktion + Minimieren ist nicht effizient genug
- 1 Mio. \$ Preisgeld für denjenigen der das schafft!
- Die meisten Informatiker vermuten, dass das unmöglich ist.

Turingmaschinen

“TM = Endlicher Automat + (unendliches) Band mit einem Kopf der ein Zeichen lesen/schreiben kann”

- Übergangsfunktion
 - Eingabe: Aktueller Zustand + Aktuelles Bandsymbol
 - Ausgabe: Neuer Zustand + Neues Bandsymbol + Armbewegung (L/N/R)
- Eingabewort: Steht zu Beginn auf dem Band.
- 2 besondere Zustände:
 - Akzeptierender Zustand
 - Verwerfender Zustand
 - In beiden ist keine weitere Aktion mehr möglich.
- Ausgabe: Akzeptierender Zustand, Verwerfender Zustand, Endlosschleife.
 - Bandinhalt kann als zusätzliche Ausgabe interpretiert werden.

Turingmaschinen



Aufgabe

Welche Worte akzeptiert/verwirft diese TM? Was steht nach Eingabe von $w \in \{a, b\}^*$ auf dem Band?

Turingmaschinen

Aufgabe

Entwerfe eine TM, die jede Eingabe $u \in \{0, 1\}^+$ akzeptiert und auf ihr Band ein Wort $w \in \{0, 1\}^+$ mit $\text{Nummer}(u) + 1 = \text{Nummer}(w)$ schreibt.

Turingmaschinen

- L ist *rekursiv aufzählbar* : $\iff$ Es gibt eine Turingmaschine die L akzeptiert (d.h. die Turingmaschine hält genau für die Worte $w \in L$ in dem akzeptierenden Zustand.)
- L ist *rekursiv* oder *entscheidbar* : $\iff$ Es gibt eine Turingmaschine die L für die Worte $w \in L$ in dem akzeptierenden Zustand hält und für $w \notin L$ in dem verwerfenden Zustand hält.

Theorem

L ist entscheidbar genau dann wenn L und $\Sigma^ - L$ beide rekursiv aufzählbar sind.*

Rekursiv aufzählbare Sprachen

Aufgabe

Eine unendliche Sprache L ist genau dann rekursiv, wenn ein Algorithmus existiert, der alle Wörter in L in *kanonischer Reihenfolge* auflistet.

Aufgabe

Beweis: Eine unendliche Sprache L ist genau dann rekursiv aufzählbar, wenn ein Algorithmus existiert, der alle Wörter in L auflistet (ohne sich an irgendeine Reihenfolge halten zu müssen).