

Minimierungsalgorithmus¹

2

Wir erstellen eine *Zustandspaatabelle* mit allen möglichen Zustandspaaren. Dabei spielt es *keine Rolle*, ob die Zustände im Automaten *verbunden sind oder nicht*.

Zustandspaatabelle
keine Rolle
verbunden sind oder nicht

Die Diagonale der Tabelle wird komplett gestrichen, da Zustände nicht mit sich selbst überprüft werden müssen. Außerdem werden die *Paare nur in einer Richtung betrachtet*. Die *obere Hälfte* der Tabelle über der Diagonalen *kann gestrichen werden*.

Paare nur in einer Richtung
obere Hälfte
kann gestrichen werden

Wir markieren die Zustandspaare, in denen ein Zustand *ein Endzustand und der andere kein Endzustand* ist.

ein Endzustand und der andere kein Endzustand

Als letztes werden mit Hilfe einer *Übergangstabelle* die noch nicht markierten Zustandspaare auf alle möglichen Übergangsmöglichkeiten überprüft. Entsteht hierbei ein *bereits gestrichenes Paar* so wird das aktuell überprüfte Paar *ebenfalls gestrichen*.

Übergangstabelle

bereits gestrichenes Paar
ebenfalls gestrichen

Wir testen jetzt für jedes Zustandspaar die Folgezustände bei der Eingabe der Zeichen des Alphabets.

Wir suchen uns nun die Zustandspaare, die bei einer Eingabe ein Zustandspaar ergeben, das bereits gestrichen ist.

Im Folgenden wird der Algorithmus zur Markierung der nicht-äquivalenten Zustände anhand einer Dreieckstabelle durchgeführt, die alle fraglichen Zustandspaare enthält.

Minimierungstabelle (Table filling)

Der Algorithmus trägt in seinem Verlauf eine Markierung in alle diejenigen Zellen der Tabelle ein, die zueinander nicht äquivalente Zustände bezeichnen. Die Markierung „ x_n “ in einer Tabellenzelle (i, j) bedeutet dabei, dass das Zustandspaar (i, j) in der k -ten Iteration des Algorithmus markiert wurde und die Zustände i und j somit zueinander $(k - 1)$ -äquivalent, aber nicht k -äquivalent und somit insbesondere nicht äquivalent sind. Bleibt eine Zelle bis zum Ende unmarkiert, sind die entsprechenden Zustände zueinander äquivalent.³

4

Literatur

- [1] Lukas König, Friederike Pfeiffer-Bohnen und Prof. Dr. Hartmut Schmeck. *100 Übungsaufgaben zu Grundlagen der Informatik. Band I: Theoretische Informatik*. 2014.
- [2] *Theoretische Informatik – Reguläre Sprachen*.
- [3] Gottfried Vossen und Kurt-Ulrich Witt. *Grundkurs Theoretische Informatik. Eine anwendungsbezogene Einführung – Für Studierende in allen Informatik-Studiengängen*. 2016.

¹Vossen und Witt, *Grundkurs Theoretische Informatik*, Seite 47-57.

²*Theoretische Informatik – Reguläre Sprachen*, Seite 51-62.

³König, Pfeiffer-Bohnen und Schmeck, *100 Übungsaufgaben zu Grundlagen der Informatik*, Seite 19.

⁴<https://studyflix.de/informatik/dea-minimieren-1212>