

Theoretische Informatik II

2. Übung

1. Aufgabe:

Sei $G = (V, \Sigma, P, S)$ eine Typ-3-Grammatik. Damit ist $L = L(G)$ eine reguläre Sprache. Aus dieser Sprache wird die neue Sprache L' durch Hinzufügen des leeren Wortes ε erzeugt: $L' = L \cup \{\varepsilon\}$.

Entwickeln Sie ein Verfahren, welches aus der Typ-3-Grammatik G eine korrekte Typ-3-Grammatik G' mit ε -Sonderregelung erzeugt, so daß $L' = L(G') = L(G) \cup \{\varepsilon\}$ gilt.

2. Aufgabe:

- a) Sei M ein NFA. Angenommen M sollen einige ε -Übergänge hinzugefügt werden um den Automaten M' zu erhalten. Es sollen also Zustandsübergänge möglich werden, ohne daß ein Buchstabe des Wortes gelesen werden muß.

Geben Sie an, wie sich die neuen Überföhrungsfunktionen δ' und $\hat{\delta}'$ aus den alten Funktionen δ und $\hat{\delta}$ ergeben.

Hinweis: Es ist der Begriff der ε -Hölle hilfreich. Die ε -Hölle eines Zustandes z ist dann die Menge der Zustände, die nur unter Benutzung von ε -Übergängen, von z aus erreicht werden können.

- b) Zeigen Sie, daß die beiden Automatenmodelle NFA und NFA mit ε -Übergang äquivalent sind. Entwickeln Sie eine Konstruktion, die einen gegebenen NFA M' mit ε -Übergang in einen NFA M ohne ε -Übergang überführt, welcher die gleiche Sprache akzeptiert.

3. Aufgabe:

Demonstrieren Sie das Verfahren zur Konstruktion von NFAs aus regulären Ausdrücken anhand von $(a|b)^*c$ über dem Alphabet $\Sigma = \{a, b, c\}$.

4. Aufgabe:

Wandeln Sie den NFA aus Aufgabe 2 mittels Potenzmengenkonstruktion in einen DFA um.

5. Aufgabe:

Demonstrieren Sie die Konstruktion von regulären Ausdrücken zu gegebenen DFAs anhand des folgenden DFAs.

6. Aufgabe:

Geben Sie einen nichtdeterministischen Automaten an, der bei einer gegebenen Eingabe ω über dem Alphabet $\{a, \dots, z\}$ feststellt, ob ω das Wort *wow* enthält.

Wandeln Sie diesen Automaten mittels der Potenzmengenkonstruktion in einen DFA um.