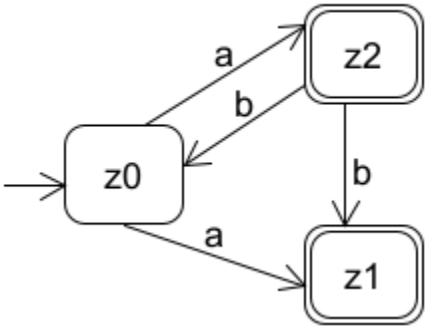

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing-Wiss. und Informatik
- Software-Entwicklung -

Theoretische Informatik

Sommersemester 2025
12. Aufgabenblatt

Aufgabe 50 (Reguläre Ausdrücke und Automaten [klausurähnlich], 4 + 5 Punkte)

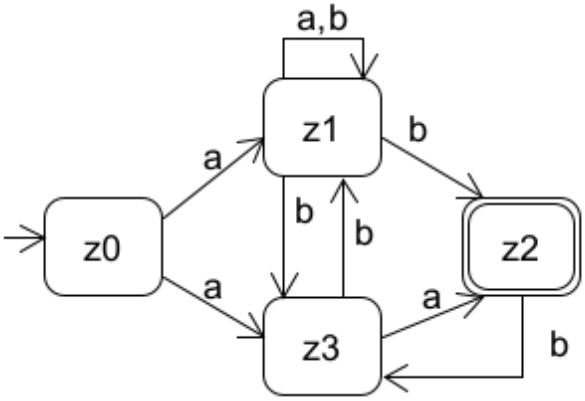
a) Geben Sie zu folgendem regulärem Ausdruck das Zustandsdiagramm eines Automaten an, der die Sprache des Ausdrucks akzeptiert:((a+(ab)*))*



b) Entwickeln Sie schrittweise durch Elimination von Zuständen zum obigen Automaten einen regulären Ausdruck, der die Sprache des Automaten beschreibt.

Aufgabe 51 (Reguläre Ausdrücke und Automaten [klausurähnlich], 4 + 5 Punkte)

a) Geben Sie zu folgendem regulärem Ausdruck das Zustandsdiagramm eines Automaten an, der die Sprache des Ausdrucks akzeptiert: ((ab)*(ba)*)*



b) Entwickeln Sie schrittweise durch Elimination von Zuständen zum obigen Automaten einen regulären Ausdruck, der die Sprache des Automaten beschreibt.

Aufgabe 52 (Reguläre Ausdrücke und Automaten)

Lesen Sie Kapitel 5.4 der theoriesammlung-Doku.

a) Geben Sie Ihre Lösung zu Aufgabe 50a) in die Datei beispiele/endlicheautomaten/AutomatAusRegulaeremAusdruck.atm ein und überprüfen Sie Ihr Ergebnis mit test.endlicherAutomat.AutomatAusRegulaeremAusdruckTest.java.
b) Geben Sie ihre Lösung zu Aufgabe 50b) in den String ausdruck in die Datei test.endlicherAutomat.AusdruckAusAutomatTest.java in die Zeile 16 an der markierten Stelle ein und überprüfen Sie Ihr Ergebnis. Hinweis: Beachten Sie, dass die in der Vorlesung und dem Werkzeug genutzte Grammatik viele Klammern nutzt.
c) Geben Sie ihre Lösung zu Aufgabe 51a) in die Datei beispiele/endlicheautomaten/AutomatAusRegulaeremAusdruck2.atm ein und überprüfen Sie Ihr Ergebnis mit test.endlicherAutomat.AutomatAusRegulaeremAusdruck2Test.java.
d) Geben Sie ihre Lösung zu Aufgabe 51b) in den String ausdruck in die Datei test.endlicherAutomat.AusdruckAusAutomat2Test.java in die Zeile 35 an der markierten

Stelle ein und überprüfen Sie Ihr Ergebnis. Hinweis: Beachten Sie, dass die in der Vorlesung und dem Werkzeug genutzte Grammatik viele Klammern nutzt.

Aufgabe 53 (eigene Umsetzung von Automatenalgorithmen)
In der gegebenen Bibliothek sind bereits die Berechnungsalgorithmen aus der Vorlesung umgesetzt. Dabei wurden die Algorithmen nur prototypisch ohne jedwede Optimierung programmiert. Lassen Sie eine Klasse von EndlicherAutomat.java erben und überschreiben Sie die folgenden oder einen der folgenden Algorithmen.

`public EndlicherAutomat epsilonEntfernen():` transformiert den Automaten sprachäquivalent so, dass keine ε-Transformationen im Automaten vorhanden sind (Rückgabe this).
`public EndlicherAutomat deterministisch():` transformiert den Automaten sprachäquivalent in einen vollständig definierten deterministischen Automaten. (Rückgabe this).
`public EndlicherAutomat minimieren():` transformiert den Automaten sprachäquivalent in einen Automaten mit minimaler Zustandsanzahl (Rückgabe this).

Tragen Sie Ihre Klasse in test.endlicherAutomat.OptimierungsTest.java in Zeile 22 ein und lassen Sie das Programm laufen. Es wird versucht, die Korrektheit zu prüfen und an Beispielen grob die Performance analysiert.
Sie können zur Bearbeitung auch die Klasse endlicherAutomat.OptimierterEndlicherAutomat.java nutzen, die einfach eine Kopie der existierenden Algorithmen enthält, die überarbeitet oder völlig neu konzipiert werden können. Anregungen für ein neues Konzept können z. B. dem Skript von Karsten Morisse entnommen werden, die es hier meist leicht abgewandelte Ansätze gibt.
Schwankungen bis mindestens 1% können durch die Laufzeitumgebung auftreten.
Da jeweils nur ein Automat zum Messen genutzt wird, könnten Sie weitere Beispiele konstruieren.

Aufgabe 54 (Pumping-Lemma für kontextfreie Sprachen)
Begründen oder widerlegen Sie möglichst detailliert, dass folgende Sprachen von einer kontextfreien Grammatik erzeugt werden können.

a) L1 = $\{a^n b^{2n} c^{3n} \mid n > 4\}$
b) L2 = $\{a^n b^{2n} c^{3n} \mid n < 3\}$
c) L3 = $\{ww \mid w \in \text{Alphabet}^*\}$ mit Alphabet = {a, b}