Prof. J. Křetínský
M. Weininger, T. Meggendorfer

Technische Universität München
PROGRAMMIERABGABE **5 Punkte**
Abgabefrist: Montag, 07.05.2018, 10 Uhr

Einführung in die theoretische Informatik

Sommersemester 2018 – Hausaufgabenblatt Lösungsskizze 3

Programmierabgabe

Für die Korrektur Ihrer Programmierabgaben wird TUMjudge(`https://judge.in.tum.de/theo/public/`) verwendet. Auf dem ersten Hausaufgabenblatt werden die für Sie relevanten Funktionen des Webinterfaces erklärt. Weitere Details finden Sie auf den Folien des Praktikums Algorithms for Programming Contests von 2017(`https://www7.in.tum.de/um/courses/theo/ss2018/materials/judge.pdf`).

Laden Sie das Codegerüst für die dritte Hausaufgabe auf `https://www7.in.tum.de/um/courses/theo/ss2018/homework/` herunter.
Betrachten Sie die Klassen State, Transition, DFA sowie NFA und EpsilonNFA. Sie sollten die Attribute, insbesondere die dafür gewählten Datentypen, verstehen, bevor Sie fortfahren.
Hinweis: Sie dürfen den Code des templates verändern. Allerdings sollten Sie main-Methode und parser so lassen, wie sie sind.

AUFGABE 3.1. (*Accept DFA*) 1 Punkte

- Implementieren Sie die Methode `accept(DFA d, String w)` in der Klasse AcceptDFA. Die Methode soll für einen DFA d und ein Wort w `true` zurück geben, wenn der DFA das Wort akzeptiert, ansonsten `false`. Sollte das Wort Buchstaben enthalten, die nicht im Alphabet des DFA sind, soll die Methode ebenfalls `false` zurück geben.
- Laden Sie dann für Problem A (AcceptDFA) *alle* Dateien hoch, die für diese Aufgabe relevant sind, also State, Transition, DFA, NFA, EpsilonNFA, Parser und AcceptDFA.

AUFGABE 3.2. (*Accept NFA*) 1 Punkte

- Implementieren Sie die Methode `accept(EpsilonNFA n, String w)` in der Klasse AcceptNFA. Die Methode soll für einen ε-NFA n und ein Wort w `true` zurück geben, wenn der ε-NFA das Wort akzeptiert, ansonsten `false`. Sollte das Wort Buchstaben enthalten, die nicht im Alphabet des ε-NFA sind, soll die Methode ebenfalls `false` zurück geben.
- Laden Sie dann für Problem B (AcceptNFA) *alle* Dateien hoch, die für diese Aufgabe relevant sind, also State, Transition, DFA, NFA, EpsilonNFA, Parser und AcceptNFA.

AUFGABE 3.3. (*Powerset*) 3 Punkte

- Implementieren Sie die Methode `powersetConstruction(NFA n)` in der Klasse Powerset. Die Methode soll für einen NFA n einen DFA zurückgeben, indem die Potenzmengenkonstruktion angewendet wird. Der DFA soll dieselbe Sprache akzeptieren wie der eingegebene NFA.
- Ihr Ergebnis-DFA wird auf Sprachgleichheit mit dem Ergebnis der Musterlösung getestet, d.h. Sie müssen nicht exakt denselben Automaten wie die Musterlösung produzieren. Sie sollten trotzdem versuchen, einen möglichst kleinen Automaten zu produzieren, um keinen Timeout zu bekommen.
- Laden Sie dann für Problem C (Powerset) *alle* Dateien hoch, die für diese Aufgabe relevant sind, also State, Transition, DFA, NFA, EpsilonNFA, Parser und Powerset.
- Die erste Zeile des Inputs ist die Zahl an Zeilen, die folgt. Die erste Zeile des Outputs ist "Case #1:". Diese beiden Dinge sind notwendig, damit der automatische Test auf Sprachgleichheit funktioniert.