

Programmierkurs Python II – SS 2013

Übung 5

1 Abschlusseigenschaften: Schnitt (3 Punkte)

Erweitere die vorgegebene Klasse `DFA` für deterministische Automaten um eine Methode `intersection(self, other)`, die den Schnitt zweier Automaten (Produktautomat) berechnet. Die Methode soll einen zweiten Automaten als Parameter nehmen und den den Schnitt-Automaten zurückgeben.

2 Infrastruktur (1 Punkt)

Implementiere in der Klasse *nicht*deterministischen Automaten `NFA` (aus Übung 4) die Methode `transitions(self)`, die die Übergänge des Automaten wieder in eine Liste geordneter Tripel (Tupel der Länge 3) konvertiert (das gleiche Format, das der Automat auch in `__init__` als Parameter nimmt).

3 Determinisierung (4 Punkte)

(1) Erweitere die gleiche Klasse `NFA`-Klasse (siehe vorherige Aufgabe) um eine Methode `is_deterministic(self)`, die `True` zurück gibt, wenn der `NFA` deterministisch ist (sonst `false`). (1 Punkt)

(2) Implementiere in `NFA` die Methode `determinize(self)`, die den Subset-Construction-Algorithmus umsetzt. Die Methode soll einen neuen Automaten (`NFA`) zurückgeben, der äquivalent mit dem aktuellen Automaten, aber deterministisch ist. Ihr könnt Mehrsymbolkanten und Vollständigkeit des `NFA` ignorieren. (3 Punkte)

4 Reguläre Ausdrücke (Bonus, 3 Punkte)

Implementiere eine Funktion `parse_regexp(strng)`, die reguläre Ausdrücke in nichtdeterministische Automaten übersetzt. Die Methode nimmt den regulären Ausdruck als String-Parameter und gibt einen äquivalenten Automaten zurück. Die Syntax des regulären Ausdrucks soll kompatibel zu den regulären Ausdrücken aus dem `re` Modul sein:

```
parse_regexp("a(bc|d)*e").recognize("abcdbce") => True
parse_regexp("a(bc|d)*e").recognize("abcdbe") => False
```

Hinweis: Es bietet sich hierfür an, die dritte Variante der Automaten-Implementierung zu verwenden. Die algorithmische Vorgehensweise zum Parsen der regulären Ausdrücke kann ähnlich zum Parser für Baum-Strings (siehe Übungsblatt 1) implementiert werden.

Abgabe bis Dienstag, 04.06.2013, 11:00 Uhr