

Programmierkurs Python II – SS 2013

Übung 4

1 Infrastruktur (1 Punkt)

Erweitere die in der Vorlesung gezeigte Beispielimplementierung nichtdeterministischer Automaten (zweite Version, siehe Code-Template) um zwei Methoden:

- `states(self)` gibt die Menge aller Zustände zurück
- `has_transition(self,src,char,tgt)` gibt `True` zurück, wenn es einen Übergang von `src` nach Zustand `tgt` durch Lesen von `char` gibt (sonst `False`).

2 Nicht-rekursive Suche (3 Punkte)

Re-Implementiere die Method `recognize(self)` ohne Rekursion.

3 Epsilon-Abschluss (3 Punkte)

In der Beispielimplementierung nichtdeterministischer Automaten (zweite Version) wird der ε -Abschluss zur Laufzeit und u.U. mehrfach berechnet. Implementiere eine Methode `removeEpsilon`, die einen neuen Automaten ohne ε -Übergänge berechnet, der die gleiche Sprache wie der ursprüngliche Automat akzeptiert.

4 Generieren mit Automaten (2 Punkte)

Mit endlichen Automaten kann man nicht nur testen, ob eine Zeichenkette erkannt (akzeptiert) wird, man kann Automaten auch verwenden, um alle Wörter einer Sprache aufzuzählen. Implementiere eine Methode `generate(self)` in der Automatenklasse, die einen Iterator über alle vom Automaten akzeptierten Wörter berechnet:

```
m = NFA(...)
for wort in m.generate():
    print(wort)
=> aa, abaa, aca, ababaa, abaca, acbaa, ...
```

Der Iterator kann (wie hier) unendlich sein; es bietet sich daher an, den Iterator so zu implementieren, dass der Automat per Breitensuche traversiert wird. Alternativ auch mit Tiefensuche benutzt werden, wobei in diesem Fall die Suche per Argument so eingeschränkt werden sollte, dass jeder Zustand maximal `n` mal besucht wird.

Abgabe bis Donnerstag, 23.05.2013, 11:00 Uhr