

# Superstring-DFA

## Aufgabenstellung

In Übungsaufgabe 2.8 wird gezeigt, wie der minimale DFA konstruiert wird, der die *Superstring-Sprache* zu einem Wort  $w$  akzeptiert (d.h. die Sprache aller Wörter, die  $w$  enthalten). In dieser Aufgabe sollen Sie diese Konstruktion implementieren. Wir beschreiben die Konstruktion im Folgenden noch einmal ausführlich:

- Jeder Zustand des Automaten entspricht einem Präfix von  $w$  (inkl.  $\varepsilon$  und  $w$  selbst).
- Das Alphabet  $\Sigma$  ist hier implizit einfach alle Zeichen die in  $w$  vorkommen.
- Der Zustand  $\varepsilon$  ist der Startzustand; der Zustand  $w$  ist der Endzustand.
- Die Idee hierbei ist, dass sich der Zustand merkt, wie viel "Fortschritt" beim Lesen von  $w$  schon gemacht wurde.
- Wenn wir uns im Zustand  $w'$  befinden, ist der Folgezustand beim Lesen eines Zeichens  $c \in \Sigma$  das längste Suffix von  $w'c$ , das gleichzeitig ein Präfix von  $w$  ist.

**Beispiel:** Das Suchwort sei  $w = abbabc$ . Es hat die Präfixe

$$\{abbabc, abbab, abba, abb, ab, a, \varepsilon\}.$$

Wenn wir uns z.B. in dem Zustand  $abbab$  befinden, gibt es drei Möglichkeiten:

- Wenn wir als nächstes ein  $c$  lesen, landen wir im (End-)zustand  $abbabc$ .
- Wenn wir als nächstes ein  $b$  lesen, haben wir nun  $abbabb$ . Die Suffixe hiervon sind

$$\{abbabb, bbabb, babb, abb, bb, b, \varepsilon\}.$$

Das längste dieser Suffixe, das auch ein Präfix von  $abbabc$  ist, ist  $abb$ . Somit landen wir im Zustand  $abb$ .

- Wenn wir ein  $a$  lesen, haben wir nun  $abbaba$ . Das längste passende Suffix ist somit  $a$ .
- Wenn wir ein Zeichen aus  $\Sigma \setminus \{a, b, c\}$  lesen, ist das längste Suffix  $\varepsilon$  und wir müssen zurück zum Startzustand.

Machen Sie sich mit der Java-Implementierung von DFAs in der Klasse `DFA` vertraut. Implementieren Sie dann die Methode `DFA.superstringDFA(String s)`, die die obige Konstruktion implementiert. Es ist empfehlenswert (aber nicht notwendig), die gleichen Zustandsnamen zu verwenden wie oben dargestellt. Der Zustandsname  $\varepsilon$  soll hierbei einfach als der String `"epsilon"` dargestellt werden. Ihr Automat muss jedoch die korrekte Sprache akzeptieren und genau  $|w| + 1$  Zustände haben.

**Hinweis:** Wenn Sie eines unserer Templates verwenden, wird die Ein- und Ausgabe komplett vom Template übernommen. Sie müssen nur die oben beschriebenen Methoden implementieren. Die folgende Beschreibung des Ein-/Ausgabeformats dient nur dem Verständnis, was die Beispiel-Testdaten bedeuten bzw. falls Sie selbst sich noch eigene Tests ausdenken wollen.

## Eingabe

Eine einzelne Zeile mit dem String, zu dem der Superstring-DFA konstruiert werden soll.

## Ausgabe

Ein DFA in dem Format, das Sie am einfachsten den Beispiel-Ausgaben unten entnehmen. Für die Beispiel-Ausgaben stehen auch PDFs zur Verfügung zur besseren Visualisierung.

## Beispiele

### Sample Input 1

a

### Sample Output 1

```
DFA
Alphabet: a
States: epsilon;a
Init: epsilon
Final: a
Transitions:
epsilon;a;a
a;a;a
END
```

### Sample Input 2

mia

### Sample Output 2

```
DFA
Alphabet: a;i;m
States: epsilon;m;mi;mia
Init: epsilon
Final: mia
Transitions:
epsilon;a;epsilon
epsilon;i;epsilon
epsilon;m;m
m;a;epsilon
m;i;mi
m;m;m
mi;a;mia
mi;i;epsilon
mi;m;m
mia;a;mia
mia;i;mia
mia;m;mia
END
```

### Sample Input 3

abba

### Sample Output 3

```
DFA
Alphabet: a;b
States: epsilon;a;ab;abb;abba
Init: epsilon
Final: abba
Transitions:
epsilon;a;a
epsilon;b;epsilon
a;a;a
a;b;ab
ab;a;a
ab;b;abb
abb;a;abba
abb;b;epsilon
abba;a;abba
abba;b;abba
END
```

**Sample Input 4**

abaabb

**Sample Output 4**

```

DFA
Alphabet: a;b
States: epsilon;a;ab;aba;abaa;abaab;abaabb
Init: epsilon
Final: abaabb
Transitions:
epsilon;a;a
epsilon;b;epsilon
a;a;a
a;b;ab
ab;a;aba
ab;b;epsilon
aba;a;abaa
aba;b;ab
abaa;a;a
abaa;b;abaab
abaab;a;aba
abaab;b;abaabb
abaabb;a;abaabb
abaabb;b;abaabb
END

```

**Sample Input 5**

mammamia

**Sample Output 5**

```

DFA
Alphabet: a;i;m
States: epsilon;m;ma;mam;mamm;mamma;mammam;mammami;mammamia
Init: epsilon
Final: mammamia
Transitions:
epsilon;a;epsilon
epsilon;i;epsilon
epsilon;m;m
m;a;ma
m;i;epsilon
m;m;m
ma;a;epsilon
ma;i;epsilon
ma;m;mam
mam;a;ma
mam;i;epsilon
mam;m;mamm
mamm;a;mamma
mamm;i;epsilon
mamm;m;m
mamma;a;epsilon
mamma;i;epsilon
mamma;m;mammam
mammam;a;ma
mammam;i;mammami
mammam;m;mamm
mammami;a;mammamia
mammami;i;epsilon
mammami;m;m
mammamia;a;mammamia
mammamia;i;mammamia
mammamia;m;mammamia
END

```