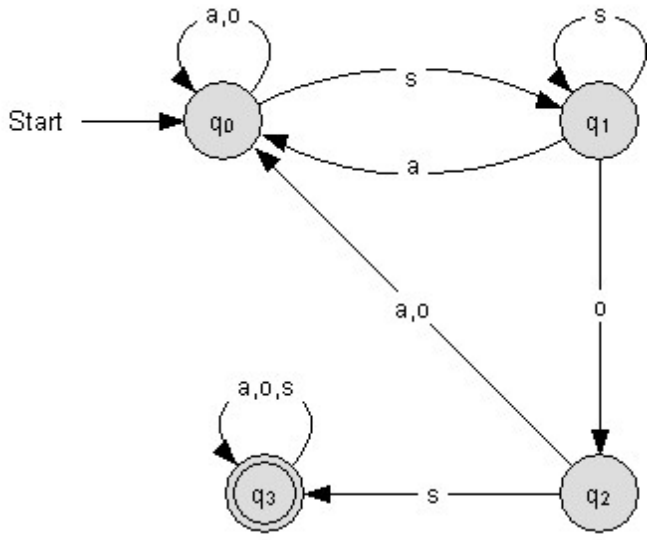
Montag, 23.09.2013

Table of Contents

Eine Lösung zur

- **Aufgabe 2**: Bearbeite das Problem mit dem Textmuster `sos` mit AtoCC. Vor dem Textmuster und danach können beliebig viele andere Zeichen vorkommen. Als Eingabe-Alphabet legen wir fest: **Σ** = {s,o,a}
  vom letzten Mal:



Wenn wir mit AtoCC einen Automaten erstellen und exportieren: `> Exportieren > Export als HTML`, dann erhalten wir neben dem Graphen und der Übergangstabelle noch eine sogenannte **Grammatik** (, in AtoCC: *Grammar*):

```
G = (N,T,P,S)

G = ({q0,q1,q2,q3},{a,o,s},P,q0)

P = {
   q0 -> a q0 | o q0 | s q1                    (1)
   q1 -> a q0 | o q2 | s q1                    (2)
   q2 -> a q0 | o q0 | s q3 | s                (3)
   q3 -> a q3 | a | o q3 | o | s q3 | s        (4)
}
```

- Was bedeutet das? Eine **Grammatik G** ist ein 4-Tupel, bestehend aus
  - N, die Menge der **Nonterminale**, in unserem Beispiel sind das *q0, q1, q2* und *q3*
  - T, die Menge der **Terminale**, in unserem Beispiel sind das *a, o* und *s*
  - P, die Menge der **Produktionen** (auch: Produktionsregeln), das ist der lange

Abschnitt zu *P*
- Hinweis: Die Zahlen in Klammern am Ende gehören **nicht** zu einer Produktionsregel und dienen lediglich zur einfachen Verständigung.

- und einem Startsymbol S ∈ N, in unserem Beispiel ist das *q0*

- Was soll das? Nun ausgehend von einem Startsymbol S (bei uns: *q0*) können wir mit Hilfe der Produktionen aus P schrittweise alle Wörter aus der Sprache L(M) des Automaten M herleiten. Statt herleiten sagt man auch **ableiten**.

- Beispiel: Wir leiten das Wort *asosa* ∈ L(M) her: *q0 -> a q0 -> a s q1 -> a s o q2 -> a s o s q3 -> a s o s a*

- Am Ende dürfen in einem Wort keine Nonterminale mehr stehen. Ich muss also solange Regeln ∈ P anwenden, bis keine Zeichen mehr aus N vorkommen! Die Worte, die hierbei entstehen, gehören alle zur Sprache L(M).

Reguläre Grammatiken

. . . verdanken wir einem amerikanischem Linguisten namens Chomsky.

Beispiel 2

Statt mit Automaten kann man auch mit **Grammatiken** arbeiten (!). Schauen wir uns ein Beispiel an:

1. **S → Ssss**
2. **S → ε**

Das ist eine einfache Grammatik G_1 mit zwei Regeln (!), und die Sprache zu dieser Grammatik besteht aus den Wörtern, die mindestens einmal drei sss in Folge haben. Beispiele für Wörter aus dieser Sprache:

- **sssssssssssssss**
- **ssssss**
- **ε**

Was bedeutet ε? Nun, das ist ein Wort aus null Buchstaben, das sogenannte **leere Wort**, in etwa vergleichbar mit der Null beim Addieren. Und warum gibt es große und kleine Buchstaben? Gleich, erst erweitern wir unsere Grammatik G_1 zur Grammatik G_2:

1. **S → Bsss**
2. **B → ε**
3. **B → Ba|Bb|Bc|Bd|Be|Bf|Bg|Bh|Bi|...|Bt|Bu|Bv|Bw|Bx|By|Bz**

Jetzt können wir zum Beispiel folgende Wörter erzeugen:

- **sss**
- **hallosss**

- **infogksss**
- **grammatiksss**

Die Punkte ... stehen für **Bj|Bk|Bl|Bm|Bn|Bo|Bp|Bq|Br**. In den Wörtern unserer Sprache kommen nur kleine Buchstaben vor, man nennt sie deshalb **Terminale**, die großen Buchstaben heißen **Nonterminale**, das **S** ist das **Startsymbol**, weil damit die Geschichte losgeht. Nonterminale müssen im Laufe der Ableitung alle durch Terminale ersetzt werden, so lautet die Regel. Und weil das wichtig ist, wiederholen wir das gleich nochmal:

> In einer Ableitung müssen schrittweise alle Nonterminale durch Terminale ersetzt werden.

Was ist eine **Ableitung**? Die "Wörter" unserer Sprache erzeugen wir durch schrittweise Anwendung der Regeln unserer Grammatik. Beispiel zur Grammatik G_1: Wir wollen das Wort **ssssss** erzeugen: **S** → **Ssss** → **Sssssss** → **ssssss** nbsp; **Das** ist eine Ableitung! Wir haben dabei 2x-mal Regel 1 angewendet und zuletzt Regel 2 aus der Grammatik G_1. Die von der Grammatik G_1 erzeugte Sprache wird mit **$L(G_1)$** bezeichnet. Eine nette Übung besteht in der Aufzählung aller Wörter der Sprache $L(G_1)$:-)

Statt 25 Regeln für jeden kleinen Buchstaben hinzuschreiben, können wir auch das Zeichen **|** benutzen, das soviel wie **oder** bedeutet. Dann kann man aber die Grammatik "vereinfachen":

2a. **B** → **Ba|Bb|Bc|Bd|Be|Bf|Bg|Bh|Bi|...|Bt|Bu|Bv|Bw|Bx|By|Bz| ε**

Wichtig: in unseren Regeln darf sowohl auf der linken als auch auf der rechten Seite nur ein Nonterminal stehen (Nein, wir haben nicht mehrmals ein Nonterminal auf der rechten Seite stehen, denn das Zeichen | ist nur als Abkürzung zu verstehen, damit wir Schreibarbeit sparen!) Muss das Nonterminal auf der rechten Seite stehen? Nein, aber es darf. Durch einen krassen Entwurfsfehler in der Grammatik G_2 haben wir aus der Sprache **$L(G_2)$** das leere Wort ε hinaus katapultiert.

- **Aufgabe 1**: Verändere die Grammatik G_2 so, dass das leere Wort ε wieder zur Sprache gehört.

Beispiele

Beispiel 3:

1. **S → aA|bB**
2. **A → aS|a**
3. **B → bS|b**

Gib einige Wörter aus dieser Sprache an: _____

Wie könnte ein Automat für diese Sprache aussehen?

Beispiel 4:

1. **S → aaS|bbS|ε**

Auch hier wieder die Aufgabe: Gib einige Wörter aus dieser Sprache an:

_____

Wie könnte ein Automat für diese Sprache aussehen?

Aufgaben

- **Aufgabe 2**: Bitte zuerst die Aufgaben oben im Abschnitt Beispiele bzw. im Abschnitt Reguläre Grammatiken bearbeiten.
- **Aufgabe 3**: Ein Kuh macht **muh** oder **muuuh**, aber nicht *mh*. Und wie sieht die Grammatik hierfür aus?
- **Aufgabe 4**: Erstelle eine Grammatik für den Info-Kaugummiautomaten (= Aufgabe 5 vom 09.09.2013).
- **Aufgabe 5**: Erstelle eine Grammatik für die Sprache aus geradzahlig vielen Nullen und geradzahlig vielen Einsen (= Aufgabe 3 vom 16.09.2013).
