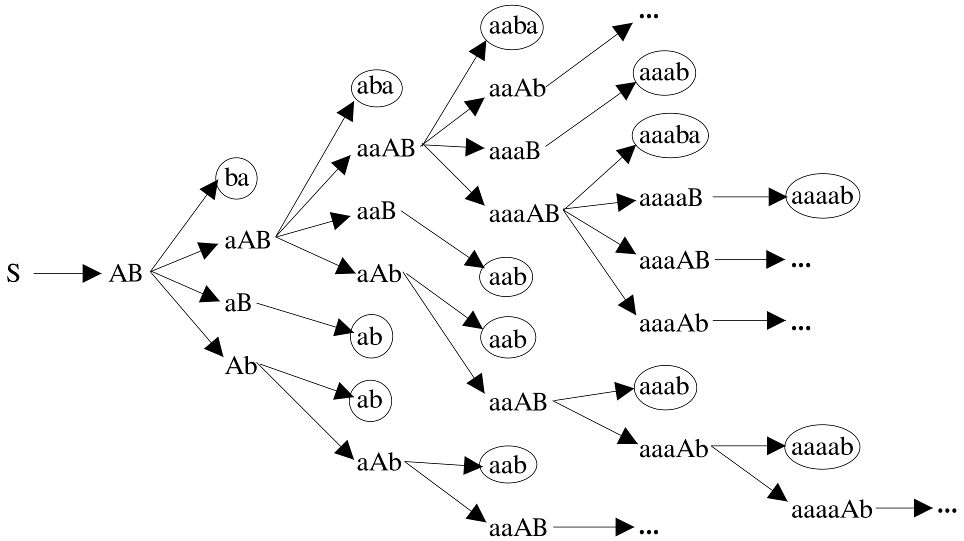
Technische Universität Chemnitz Sommersemester 2002
Professur Theoretische Informatik
Prof. Dr. A. Goerdt, Dipl.-Inf. F. Schädlich, Dipl.-Inf. A. Lanka

3. Übung „Theoretische Informatik II“
Lösungsvorschläge

Aufgabe 1 :

$\Sigma = \{a, b\}$; $P = \{\ S \to AB,\ A \to a \mid aA,\ B \to b,\ AB \to ba\}$; Sprachbaum :



Worte der Länge	1 : keine
	2 : ab, ba
	3 : aab, aba
	4 : aaab, aaba
	5 : aaaab, aaaba

Wir suchen : **a)** $x = b$ **b)** $x = aba$ **c)** $x = aaab$ **d)** $x = aabaa$

Somit sind die Worte x von Aufgabe b) und c) aus L und die x von a) und d) nicht.

In der Aufgabe konnten wir bei einer vorgegebenen Grammatik G für eine Reihe von Worten $x \in \Sigma^*$ die Frage beantworten, ob x zu der von G erzeugten Sprache gehört oder nicht. Letztere Frage nennt man das *Wortproblem*.

Für jede Grammatik G kann man einen „Grammatikbaum“ (nicht zu verwechseln mit dem Syntaxbaum !) zeichnen und jedes Blatt des Baumes ist ein Wort der von G erzeugten Sprache L.

Sei nun G eine Typ 1-Grammatik und sei ein Wort $x \in \Sigma^*$ der Länge n gegeben. Wegen der Bedingung

$$|w_1| \leq |w_2| \qquad (1)$$

für jede Regel $w_1 \to w_2$ aus G genügt es, den Sprachbaum nur so weit zu zeichnen, bis alle noch weiter auflösbaren Knoten ein Wort (aus $(\Sigma \cup V)^*$) der Länge größer als n enthalten. Ist x zu diesem Zeitpunkt als Blatt des Sprachbaumes aufgetaucht, so ist x offenbar ein Wort der Sprache L. Ist x noch nicht aufgetaucht, so wird auf Grund der nichtverkürzenden Regeln auch im gesamten Sprachbaum nicht mehr auftauchen und die Frage „$x \in L$?“ ist mit „nein“ zu beantworten.

Sprachen, für die das Wortproblem entscheidbar ist, nennt man selbst *entscheidbar*.

Wegen der obigen Überlegungen sind Typ 1 (und somit Typ 2 und 3)-Sprachen offenbar entscheidbar. Auch einige Typ 0-Sprachen sind entscheidbar - aber nicht alle !

Übrigens deckt die Menge der Typ 0-Sprachen längst nicht die Menge aller Sprachen ab. Die Menge der Typ 0-Sprachen ist abzählbar, d. h.:

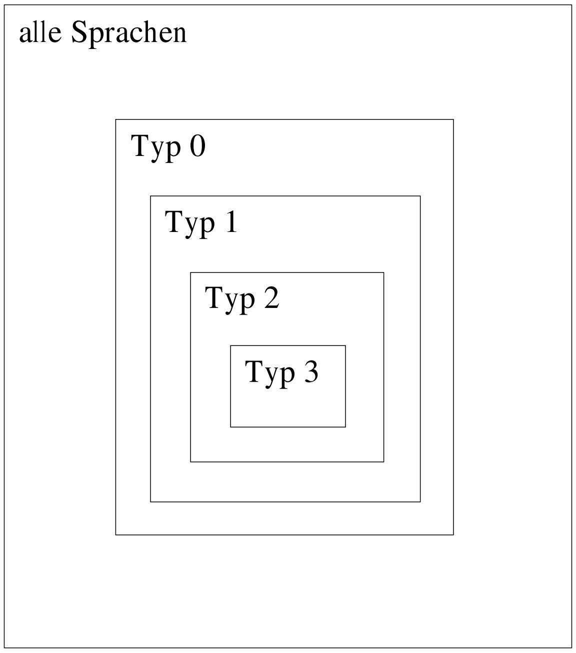
Es gibt so viele Typ 0-Sprachen wie es natürliche Zahlen gibt.

Die Menge aller Sprachen hingegen - ja, bereits die Menge aller Sprachen über einem zweielementigen Alphabet $\{a, b\}$ (=Menge aller Teilmengen, d.h. Potenzmenge von $\{a, b\}^*$) - ist überabzählbar.

Es gibt so viele Sprachen wie es reelle Zahlen gibt.

So gesehen ist die Menge der Typ 0-Sprachen, verglichen mit der Menge aller Sprachen, verschwindend klein. In der Maßtheorie sagt man, sie ist „vom Maße Null“. Für die absolute Mehrzahl der Sprachen gibt es also überhaupt keine Grammatik.

Hierarchie der Sprachen :



Endliche Automaten

In der ersten Übung hatten wir gesehen, daß die meisten relevanten Sprachen unendliche Objekte sind und daß es uns gelingen muß, möglichst viele von ihnen mit endlichen Mitteln zu beschreiben. Eine Möglichkeit hierzu haben wir mittlerweile hinreichend gründlich kennengelernt: Grammatiken.

Eine andere Möglichkeit sind die sogenannten *endlichen Automaten*. Ein endlicher Automat ist gegeben durch ein Alphabet Σ und durch einen gerichteten Graphen, dessen Knoten wir als *Zustände* bezeichnen. Die Menge der Zustände sei Z. Einen dieser Zustände nennen wir den *Startzustand* s und eine gewisse Teilmenge F von Z sei die Menge der *Endzustände*.

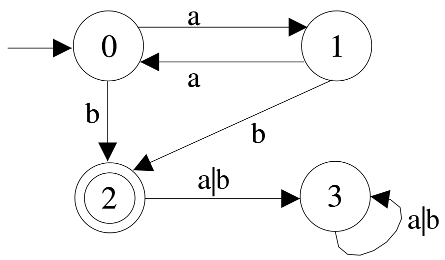
Die Kanten des Graphen sind gegeben durch eine sogenannte *Überführungsfunktion* δ. Die Abbildung

$$\delta \; : \; Z \times \Sigma \; \rightarrow \; Z$$

ordnet jedem Paar $(z_1 \in Z, a \in \Sigma)$ einen Zustand $z_2 \in Z$ zu. Für die Gestalt des Graphen bedeutet die Tatsache $\delta(z_1, a) = z_2$, daß es eine (gerichtete) Kante vom Knoten z_1 zum Knoten z_2 gibt, die wir außerdem mit a beschriften.

Beispiel : Automat $A = \Big(Z = \{S_0, S_1, S_2, S_3\} \; , \; \Sigma = \{a, b\} \; , \; \delta \; , \; S = S_0 \; , \; F = \{S_2\} \Big)$

Σ / Z_{alt}	a	b
S_0	S_1	S_2
S_1	S_0	S_2
S_2	S_3	S_3
S_3	S_3	S_3



Von jedem Knoten geht offenbar pro Terminalzeichen genau eine Kante aus.

Man sagt, der Automat A *akzeptiert das Wort* $x = a_1 a_2 ... a_n \in \Sigma^n \subset \Sigma^*$, falls die sequentielle Abarbeitung der Zeichen $a_1, a_2, ..., a_n$, beginnend beim Startzustand s in einem Endzustand $z \in F$ endet. Ein Zeichen $a \in \Sigma$ abzuarbeiten heißt dabei, vom aktuellen Zustand in den Zustand zu wechseln, zu dem die Kante, die mit dem Terminal a beschriftet ist, führt.

Die Menge aller Worte, die der Automat A akzeptiert nennen wir die von ihm *akzeptierte Sprache* L.

Die vom Automaten im obigen Beispiel akzeptierte Sprache ist gerade

$$L = \{a^n b : n \geq 0\}.$$

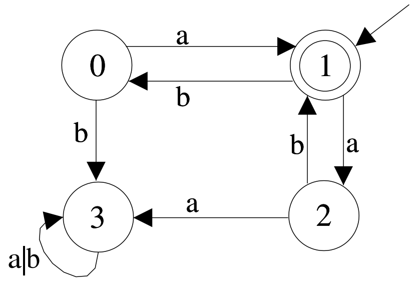
Aufgabe 2

Sowohl in **a)** als auch in **b)** gilt für den Automaten $A = (Z, \Sigma, \delta, S, F)$:

Σ / Z_{alt}	a	b
0	1	3
1	2	0
2	3	1
3	3	3

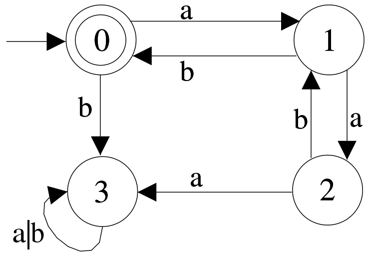
und der Unterschied besteht nur in der Wahl der Start- bzw. Endzustände.

a) $S = S_1,\ F = \{S_1\}$, Zustandsgraph :



Akzeptierte Sprache $L_1 = \{ab, ba\}^*$

b) $S = S_0,\ F = \{S_0\}$, Zustandsgraph :



Akzeptierte Sprache $L_2 = L\left(\left(a(ab)^*b\right)^*\right)$.

Aufgabe 3

a)

$G = (V, \Sigma, P, S)$
$\Sigma = \{a, b\}$
$V = \{S_0, S_1, S_2, S_3\}$
$S = S_1'$

$$
\begin{aligned}
P = \{\ & S_1' \to \varepsilon, \\
& S_1' \to aS_2, \\
& S_1' \to bS_0, \\
& S_0 \to aS_1, \\
& S_0 \to bS_3, \\
& S_0 \to a, \\
& S_1 \to aS_2, \\
& S_1 \to bS_0, \\
& S_2 \to aS_3, \\
& S_2 \to bS_1, \\
& S_2 \to b, \\
& S_3 \to aS_3, \\
& S_3 \to bS_3\}
\end{aligned}
$$

b)

$G = (V, \Sigma, P, S)$
$\Sigma = \{a, b\}$
$V = \{S_0, S_1, S_2, S_3\}$
$S = S_0'$

$$
\begin{aligned}
P = \{\ & S_0' \to \varepsilon, \\
& S_0' \to aS_1, \\
& S_0' \to bS_3, \\
& S_0 \to aS_1, \\
& S_0 \to bS_3, \\
& S_1 \to aS_2, \\
& S_1 \to bS_0, \\
& S_1 \to b, \\
& S_2 \to aS_3, \\
& S_2 \to bS_1, \\
& S_3 \to aS_3, \\
& S_3 \to bS_3\}
\end{aligned}
$$

Aufgabe 4

a)

$$M = (Z,\ \Sigma,\ \delta,\ S,\ E)$$

$$\delta\colon Z \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(\Sigma)$$

$$\begin{aligned}\epsilon\text{-Hülle}(z) &= \{z\} \cup \{q\colon \text{ Es existiert ein mit } \epsilon \text{ markierter Pfad von } z \text{ nach}\\ &\qquad q \text{ im Zustandsgraphen des Automaten}\}\\ \epsilon\text{-Hülle}(Z') &= \bigcup_{z \in Z'} \epsilon\text{-Hülle}(z)\end{aligned}$$

$$\hat{\delta}\colon \mathcal{P}(Z) \times \Sigma^* \to \mathcal{P}(\Sigma)$$

$$\begin{aligned}\hat{\delta}(Z', \epsilon) &= \epsilon\text{-Hülle}(Z')\\ \hat{\delta}(Z', x\omega) &= \bigcup_{z \in \epsilon\text{-Hülle}(Z')} \hat{\delta}(\delta(z, x), \omega)\end{aligned}$$

b) M' sei der zum Automaten M äquivalente NEA ohne ϵ-Transition:

$$M' = (Z,\ \Sigma,\ \delta',\ S,\ E')$$

$$E' = E \cup \{z\colon\ z \in S \text{ und } \epsilon\text{-Hülle}(z) \cap E \neq \emptyset\}$$

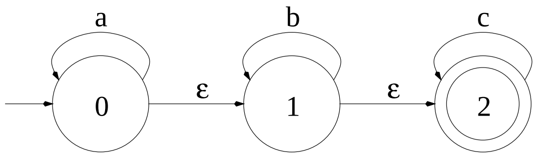
$$\delta\colon Z \times \Sigma \to \mathcal{P}(\Sigma)$$

$$\delta(z, x) = \hat{\delta}(\{z\}, x)$$

Die Korrektheit der erweiterten Überführungsfunktion $\hat{\delta}$ für den NEA M mit ϵ-Bewegung in Teilaufgabe (a) bzw. die Korrektheit der Konstruktion des zu M äquivalenten NEA M' ohne ϵ-Transition in Teilaufgabe (b) kann leicht per Induktion nachgewiesen werden.

Aufgabe 5

a)



b)