

Aufgabe 1

1.1: Die Grammatik G ist eine **Phrasenstrukturgrammatik (Typ 0)**, weil sie die ε -Produktion erhält und damit $1 = |l| \not\leq |r| = 0$ für $\langle B \rangle \rightarrow \varepsilon$.

Sieht man darüber hinweg, ist die Grammatik G eine **kontextfreie Grammatik (Typ 2)**, weil die linke Seite nur aus Einzel-Nicht-Terminal-Symbolen besteht. Sie ist keine reguläre Grammatik, weil z.B. die Regel $\langle A \rangle \rightarrow a \langle B \rangle \langle B \rangle a$ nicht die Anforderungen erfüllt.

1.2:

$G = \text{Grammatik}(V, \Sigma, P, S)$ mit

$V := \{\langle S \rangle; \langle A \rangle; \langle B \rangle\}$

$\Sigma := \{a; b\}$

$P := \{(\langle S \rangle, a \langle S \rangle); (\langle S \rangle, \varepsilon); (\langle S \rangle, \langle A \rangle \langle B \rangle);$
 $(\langle A \rangle, a); (\langle A \rangle, aa); (\langle A \rangle, a \langle B \rangle a); (\langle A \rangle, a \langle B \rangle \langle B \rangle a);$
 $(\langle B \rangle, \varepsilon); (\langle B \rangle, b \langle B \rangle); \}$

$S := \langle S \rangle$

1.3: G ohne ε -Produktionen. Die unnötige Regel $\langle A \rangle \rightarrow a \langle B \rangle \langle B \rangle a$ weggelassen.

$\langle S \rangle \rightarrow \langle T \rangle \mid \varepsilon$

$\langle T \rangle \rightarrow a \mid a \langle T \rangle \mid \langle A \rangle \langle B \rangle \mid \langle A \rangle$

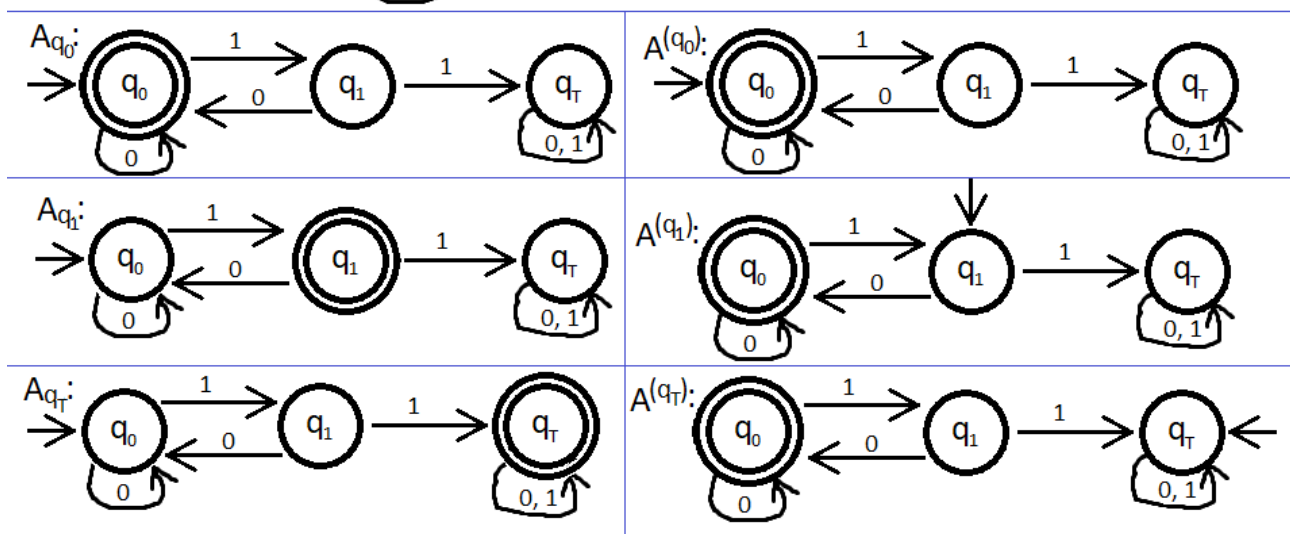
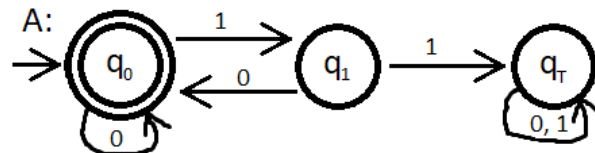
$\langle A \rangle \rightarrow a \mid aa \mid a \langle B \rangle a$

$\langle B \rangle \rightarrow b \mid b \langle B \rangle$

Aufgabe 2

2.1: $L_0 \setminus L = \{b; c\}$

2.2:



Aufgabe 3

programm calculator(expression job) {

Das Programm enthält die Funktion calc, die eine „expression“ erwartet und mit einer ganzen Zahl antwortet.

```
    structure expression {  
        int type, value;  
        expression *lhs, *rhs;  
    }
```

berechne calc(job), gib das Ergebnis am Bildschirm aus und beende dann das Programm;

Funktion calc(expression job) {

```
    if ((job.lhs == nil) AND (job.rhs == nil)) then {  
        antworte mit job.value;
```

```
    } else if ((job.lhs != nil) AND (job.rhs == nil)) then {  
        berechne calc(job.rhs) und antworte mit dem Ergebnis;
```

```
    } else if ((job.lhs != nil) AND (job.rhs != nil)) then {  
        berechne calc(job.lhs) und calc(job.rhs);
```

```
        verrechne die beiden Ergebnisse entsprechend des job.type und antworte mit dem Ergebnis.
```

```
    } else {  
        Fehler!
```

```
    }
```

```
}
```

```
}
```

Aufgabe 4

4.1: $M := DEA(\Theta; \Sigma; \delta; q_0; \zeta)$ mit

$\Theta := \{q_0; q_F; q_T\} \supseteq \zeta := \{q_F\}$

$\Sigma := \{\omega \mid \omega \text{ ist ASCII-Zeichen}\}$

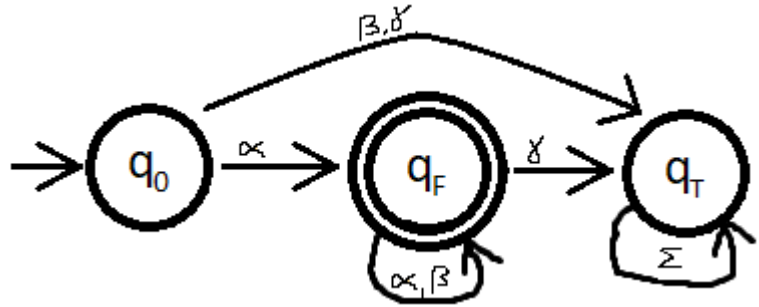
$\alpha := \{A;a; B;b; C;c; D;d; \dots; Z;z\}$

$\beta := \{0; 1; 2; 3; \dots; 9; _ \}$

$\gamma := \Sigma \setminus (\alpha \cup \beta)$

$\delta: \Theta \times \Sigma \rightarrow \Theta$ ist gegeben durch

$\begin{array}{c} \Theta \\ \backslash \Sigma \end{array}$	α	β	γ
q_0	q_F	q_T	q_T
q_F	q_F	q_F	q_T
q_T	q_T	q_T	q_T



4.2: Wort Hallo:

$q_0, \text{Hallo} \Rightarrow q_F, \text{allo} \Rightarrow q_F, \text{llo} \Rightarrow q_F, \text{lo} \Rightarrow q_F, \text{o} \Rightarrow q_F$

$q_0, \text{E=mcc} \Rightarrow q_F, \text{=mcc} \Rightarrow q_T, \text{mcc} \Rightarrow q_T, \text{cc} \Rightarrow q_T, \text{c} \Rightarrow q_T$

4.3: expression, type und value werden akzeptiert. *lhs und *rhs werden nicht akzeptiert.

$$\begin{aligned}
 \hat{\delta}(q_0; \text{value}) &= \delta(\hat{\delta}(q_0; \text{valu}); e) = \delta(\delta(\hat{\delta}(q_0; \text{val}); u); e) = \dots \\
 &= \delta(\delta(\delta(\hat{\delta}(q_0; \text{va}); l); u); e) = \delta(\delta(\delta(\delta(\hat{\delta}(q_0; v); a); l); u); e) = \\
 &= \delta(\delta(\delta(\delta(\delta(\hat{\delta}(q_0; \varepsilon); v); a); l); u); e) = \\
 &= \delta(\delta(\delta(\delta(\delta(q_0; v); a); l); u); e) = \\
 &= \delta(\delta(\delta(\delta(q_F; a); l); u); e) = \\
 &= \delta(\delta(\delta(q_F; l); u); e) = \\
 &= \delta(\delta(q_F; u); e) = \\
 &= \delta(q_F; e) = q_F
 \end{aligned}$$

$$\begin{aligned}
 \hat{\delta}(q_0; *rhs) &= \delta(\hat{\delta}(q_0; *rh); s) = \delta(\delta(\hat{\delta}(q_0; *r); h); s) = \delta(\delta(\delta(\hat{\delta}(q_0; *); r); h); s) = \dots \\
 &= \delta(\delta(\delta(\delta(\hat{\delta}(q_0; \varepsilon); *); r); h); s) = \\
 &= \delta(\delta(\delta(\delta(q_0; *); r); h); s) = \\
 &= \delta(\delta(\delta(q_T; r); h); s) = \\
 &= \delta(\delta(q_T; h); s) = \\
 &= \delta(q_T; s) = q_T
 \end{aligned}$$

4.4: Man könnte das leere Wort ε nicht mehr in die Funktion einsetzen.