



## 5. Übungsblatt zu FGdI 1

### Gruppenübung

#### Aufgabe G1

Betrachten Sie die kontextfreie Grammatik  $G = (\{a, b\}, \{X_0, X, Y\}, P, X_0)$  mit

$$\begin{aligned}P: X_0 &\rightarrow aXaY \\ X &\rightarrow aXa \mid Y \\ Y &\rightarrow bY \mid \epsilon.\end{aligned}$$

- (a) Bestimmen Sie die von  $G$  erzeugte Sprache.
- (b) Bestimmen Sie eine zu  $G$  äquivalente Grammatik  $G'$  ohne  $\epsilon$ -Produktionen.
- (c) Zeigen Sie, dass die Sprache

$$L = \{x \in \{a, b\}^* : |x|_a = |x|_b\}$$

kontextfrei ist, indem Sie eine kontextfreie Grammatik angeben, die diese Sprache erzeugt. (Begründen Sie Ihre Antwort!) Vgl. Übung 3.1.13 im Skript.

#### Musterlösung:

- (a) Man zeigt mit Induktion, dass sich aus  $X$  genau  $a^n X a^n$ ,  $a^n b^m Y a^n$  und  $a^n b^m a^n$  erzeugen lassen. Daraus folgt  $L(G) \subseteq \{a^n b^m a^n b^k : n \geq 1, m \geq 0, k \geq 0\}$ . Umgekehrt kann man mit Induktion zeigen, dass aus  $X$  damit leicht eine Ableitung für jedes Wort aus dieser Menge angeben. Damit  $L(G) = \{a^n b^m a^n b^k : n \geq 1, m \geq 0, k \geq 0\}$ .
- (b) Wir ersetzen  $Y \rightarrow \epsilon$  durch

$$\begin{aligned}X_0 &\rightarrow aXa \\ X &\rightarrow \epsilon \\ Y &\rightarrow b.\end{aligned}$$

Und dann ersetzen wir  $X \rightarrow \epsilon$  durch

$$\begin{aligned}X_0 &\rightarrow aaY \\ X &\rightarrow aa \\ X_0 &\rightarrow aa.\end{aligned}$$

Insgesamt:

$$\begin{aligned}X_0 &\rightarrow aXaY \mid aXa \mid aaY \mid aa \\ X &\rightarrow aXa \mid Y \mid aa \\ Y &\rightarrow bY \mid b.\end{aligned}$$

- (c)

$$X \rightarrow \epsilon \mid aXbX \mid bXaX$$

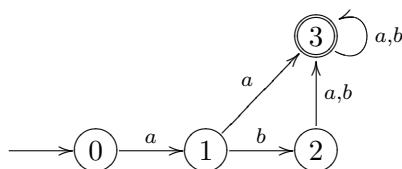
Es ist klar, dass jedes Wort  $w$  dieser Sprache  $|w|_a = |w|_b$  erfüllt, da alle Produktionen diese Gleichheit erhalten (sie gilt also per Induktion für alle Zwischenergebnisse aller

Ableitungen; insbesondere für die Endprodukte). Wir müssen nun begründen, warum ein beliebiges Wort  $w$  mit dieser Eigenschaft auch von der Grammatik erzeugt wird. Dazu benutzen wir Induktion über die Länge des Wortes  $w$ . Das leere Wort  $\epsilon$  ist in  $L$  enthalten und kann mit unserer Grammatik erzeugt werden. Für den Induktionsschritt nehmen wir an, dass unsere Grammatik alle Wörter  $u$  mit  $|u| < n$  und  $|u|_a = |u|_b$  erzeugt. Sei nun  $w$  ein Wort der Länge  $n$  mit  $|w|_a = |w|_b$ . Wir nehmen an, dass das Wort mit  $a$  beginnt. Nun gibt es einen minimalen Präfix  $v$  von  $w$  mit  $|v|_a = |v|_b$ . Man macht sich leicht klar, dass dieser Präfix auf  $b$  enden muss, da es sonst einen kürzeren Präfix mit den gleichen Eigenschaften gäbe. Also gibt es eine Zerlegung  $w = av'bu$ , so dass  $v'$  und  $u$  in  $L$  enthalten sind ( $v'$  ist  $v$  ohne den ersten und letzten Buchstaben, demnach folgt  $|v'|_a = |v'|_b$  und  $|u|_a = |u|_b$  aus der Voraussetzung  $|w|_a = |w|_b$ ). Nach Induktionsvoraussetzung können wir  $v'$  und  $u$  aus  $X$  mit unseren Ableitungsregeln erzeugen. Wenn wir also  $X \rightarrow aXbX$  benutzen und dann aus dem ersten  $X$   $v'$  und aus dem zweiten  $X$   $u$  erzeugen, haben wir eine Ableitung für das Wort  $w$  konstruiert. Diese Argument gilt analog auch für Worte, die mit  $b$  beginnen, hier benutzt man die Produktion  $X \rightarrow bXaX$ .

### Aufgabe G2

Sei  $\Sigma = \{a, b\}$ .

- (a) Betrachten Sie den Automaten  $\mathcal{A}$ :



Geben Sie eine reguläre Grammatik an, die die Sprache  $L(\mathcal{A})$  erzeugt.

- (b) Zeigen sie dass die folgenden beiden Grammatiken  $G_1, G_2$  mit Startsymbol  $X$  dieselbe reguläre Sprache erzeugen:

$$\begin{array}{ll}
 G_1 : X \rightarrow XaXaX \mid Y & G_2 : X \rightarrow aY \mid bX \mid \epsilon \\
 Y \rightarrow YY \mid b \mid \epsilon & Y \rightarrow bY \mid aZ_1 \mid aX \\
 & Z_1 \rightarrow aZ_2 \mid bX \mid \epsilon \\
 & Z_2 \rightarrow bY \mid aZ_1 \mid aX
 \end{array}$$

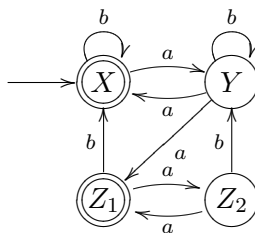
Dazu übersetzt man zunächst die zweite, rechtslineare Grammatik in einen (möglichst kompakten) DFA um die erzeugte Sprache zu identifizieren. Dann zeige man, dass die erste Grammatik ebenfalls genau diese Sprache erzeugt (mit Induktionsbeweisen für die beiden Inklusionen!).

### Musterlösung:

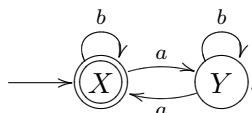
- (a)  $G = (\{a, b\}, \{X_0, X_1, X_2, X_3\}, P, X_0)$

$$\begin{array}{ll}
 P : X_0 & \rightarrow aX_1 \\
 X_1 & \rightarrow aX_3 \mid bX_2 \\
 X_2 & \rightarrow aX_3 \mid bX_3 \\
 X_3 & \rightarrow aX_3 \mid bX_3 \mid \epsilon
 \end{array}$$

- (b) Da  $G_2$  rechtslinear ist, lässt sie sich direkt in folgenden NFA übersetzen:



Nach einer Determinisierung und Minimierung erhält man folgenden DFA:



Alternativ kann die Grammatik  $G_2$  auch vorher vereinfacht werden. Dafür kennen wir aber *kein* Verfahren, das in jedem Fall funktioniert:  $Z_2$  und  $Y$  können miteinander identifiziert werden, da sie das Gleiche produzieren. Nach dieser Identifikation sieht man, dass auch  $Z_1$  und  $X$  identifiziert werden können. Wir erhalten folgende Grammatik

$$\begin{aligned} X &\rightarrow aY \mid bX \mid \epsilon \\ Y &\rightarrow bY \mid aX \end{aligned}$$

und auch den obigen DFA.

Die Grammatiken erzeugen damit die Sprache  $L := \{x \in \{a, b\}^* : |x|_a \equiv_2 0\}$ .

Mit Induktion sieht man leicht, dass die Sprache der ersten Grammatik in  $L$  enthalten ist.

Wir zeigen mit Induktion über  $|w|_a/2$ , dass mit der ersten Grammatik alle Worte aus  $L$  erzeugt werden können.

Induktionsanfang: Falls  $|w|_a/2 = 0$ , dann besteht  $w$  nur aus  $bs$  und kann offensichtlich erzeugt werden.

Induktionsschritt: Angenommen alle Worte  $v \in L$  mit  $|v|_a/2 = n$  können erzeugt werden. Sei  $w \in L$  nun ein Wort mit  $|w|_a/2 = n + 1$ . Dann lässt sich  $w$  schreiben als

$$w = uau'av$$

mit  $u, u' \in L(b^*)$  und  $v \in L$  und  $|v|_a/2 = n$ .  $u, u'$  können trivial aus  $X$  produziert werden und  $v$  nach Induktionsvoraussetzung. Damit produziert  $XaXaX$  das Wort  $w$ .

### Aufgabe G3

Sei

$$L = \{ss^{-1}t : s, t \in \{a, b\}^+\},$$

wobei  $s^{-1}$  die Umdrehung von  $s$  bezeichnet (wie auf dem letzten Übungsblatt definiert).

- Geben Sie eine kontextfreie Grammatik für  $L$  an.
- Zeigen Sie, dass  $L$  die Aussage im Pumping Lemma erfüllt.
- Zeigen Sie, dass  $L$  trotzdem nicht regulär ist.

Hinweis: Benutzen Sie den Satz von Myhill-Nerode!

### Musterlösung:

- $G = (\{a, b\}, \{S, P, U\}, R, S)$

$$\begin{aligned} R : S &\rightarrow aPaUa \mid bPbUa \mid aPaUb \mid bPbUb \\ P &\rightarrow aPa \mid bPb \mid \epsilon \\ U &\rightarrow Ua \mid Ub \mid \epsilon \end{aligned}$$

- (b) Wir zeigen, dass  $L$  die Behauptung des Pumping Lemmas für  $n = 4$  erfüllt. Dafür sei  $x = ss^{-1}t$  ein Wort, das zu  $L$  gehört, und mindestens Länge 4 hat. Es gibt zwei Möglichkeiten:
- $s$  hat Länge 1. Dann hat  $t$  mindestens Länge 2. Wir wählen  $u := ss^{-1}$ , also enthält  $u$  die ersten beiden Buchstaben von  $x$ . Wähle nun  $v$  als den dritten Buchstaben von  $x$  und  $w$  sei der Rest von  $x$ . Da  $u$  ein nicht-leeres Palindrom gerader Länge und  $w$  nicht leer ist, gilt  $uv^m w \in L$  für alle  $m \in \mathbb{N}$ .
  - $s$  hat mindestens Länge 2. Dann wählen wir  $u := \epsilon$  und  $v$  der erste Buchstabe von  $x$ .  $w$  bezeichne wieder den (nicht leeren) Rest von  $x$ . Dann gehört  $uv^m w = v^m w$  immer zu  $L$ : für  $m > 1$  fängt  $v^m w$  nämlich mit  $vv$  an. Da  $v$  nur aus einem Buchstaben besteht, ist  $vv = vv^{-1}$  und das Wort  $v^m w$  damit in  $L$ . Gilt  $m = 0$ , dann ist  $v^m w = w$  und  $w$  gehört zu  $L$ , da  $w = s_0 s_0^{-1} v w$ , wobei  $s_0$  das Wort  $s$  ohne den ersten Buchstaben ist.
- (c) Aus dem Satz von Myhill-Nerode folgt, dass es reicht, eine unendliche Menge von Wörtern zu finden, die bezüglich der Äquivalenzrelation  $\sim_L$  paarweise inäquivalent sind. Betrachte die Wörter  $\{ab^{2n+1}a : n \in \mathbb{N}\}$ . Für  $n \neq n'$  gilt  $ab^{2n+1}a \not\sim_L ab^{2n'+1}a$ , denn  $ab^{2n+1}aw \in L$  und  $ab^{2n'+1}aw \notin L$  für  $w := ab^{2n+1}ab$ .

## Hausübung

### Aufgabe H1

(6 Punkte)

Die Grammatik  $G = (\Sigma, V, P, S)$  mit  $\Sigma = \{\mathbf{a}, \dots, \mathbf{z}, \mathbf{0}, \dots, \mathbf{9}, \mathbf{:}, \mathbf{=}, \mathbf{(}, \mathbf{)}\}$  und  $V = \{S, E\}$  spezifiziert eine einfache Programmiersprache, wobei  $P$  geben sei durch

$$\begin{aligned} S &\rightarrow \mathbf{y} := \mathbf{0} \mid \mathbf{y} := \mathbf{1} \mid \mathbf{if } E \mathbf{ then } S \mathbf{ else } S \mid \mathbf{if } E \mathbf{ then } S \\ E &\rightarrow \mathbf{x} = \mathbf{0} \mid \mathbf{y} = \mathbf{0} \end{aligned}$$

(Die Nichtterminale  $S$  und  $E$  stehen für Anweisungen (*statements*) und Ausdrücke (*expressions*)).

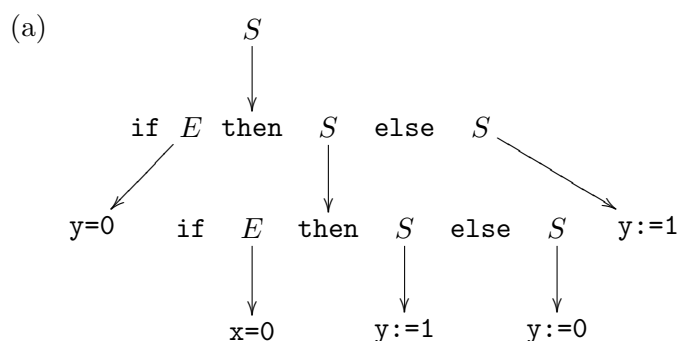
- (a) Zeichnen Sie den Ableitungsbaum (siehe Skript, Übung 3.1.5) zum Wort

```
if y = 0 then if x = 0 then y:=1 else y:=0 else y:=1
```

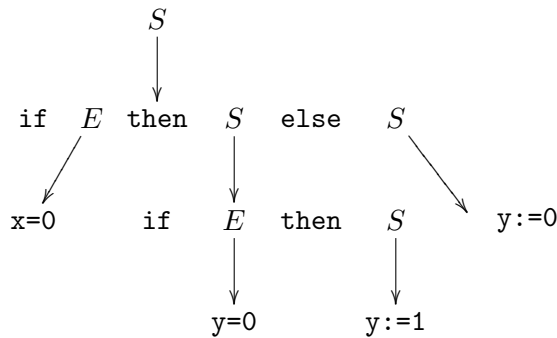
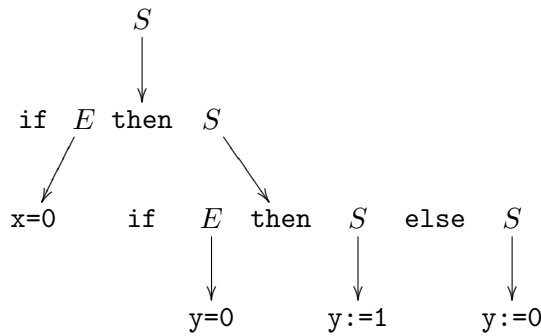
- (b) Finden Sie ein Wort mit zwei verschiedenen Ableitungsbäumen.  
 (c\*) Geben Sie eine zu  $G$  äquivalente Grammatik an, bei der jedes Wort höchstens einen Ableitungsbaum besitzt.

Zusatz: Überlegen Sie sich, welche Rolle die eindeutige Ableitbarkeit bei der Implementierung eines Compilers spielt.

### Musterlösung:



- (b)
- ```
if x=0 then if y=0 then y:=1 else y:=0
```



- (c\*) Die in (b) aufgezeigte Uneindeutigkeit bei der Beschreibung von if-then-else Konstrukten ist unter dem Begriff “dangling else” bekannt und wird z.B. auf Wikipedia diskutiert. Sei nun  $G' := (\Sigma, V \cup \{S'\}, P', S)$ , wobei  $P'$  als

$$\begin{aligned} S &\rightarrow y := 0 \mid y := 1 \mid \text{if } E \text{ then } S' \text{ else } S \mid \text{if } E \text{ then } S \\ S' &\rightarrow y := 0 \mid y := 1 \mid \text{if } E \text{ then } S' \text{ else } S' \\ E &\rightarrow x = 0 \mid y = 0 \end{aligned}$$

definiert ist.

Es ist klar, dass  $L(G') \subseteq L(G)$ . Für die andere Richtung kann man zeigen, dass man in jeden  $G$ -Ableitungsbaum in einen  $G'$ -Ableitungsbaum umformen kann. (Man verschiebt die Ableitung des **else**  $S$  an das letzte nachfolgende **if**, das kein **else** hat.) Man kann zeigen, dass diese Ableitungsbäume eindeutig sind.

### Aufgabe H2

(6 Punkte)

Sei  $\mathcal{A}$  ein DFA mit  $n$  Zuständen. Der Beweis des Pumping Lemmas zeigt dann, dass es für jedes Wort  $x \in L(\mathcal{A})$  mit  $|x| \geq n$  eine Zerlegung  $x = u \cdot v \cdot w$  mit  $|u \cdot v| \leq n$  und  $|v| > 0$  gibt, so dass

$$u \cdot v^m \cdot w \in L(\mathcal{A})$$

für jedes  $m \in \mathbb{N}$ .

- Zeigen Sie, dass aus  $L(\mathcal{A}) \neq \emptyset$  folgt, dass es ein Wort  $x \in L(\mathcal{A})$  gibt mit  $|x| < n$ .  
Hinweis: Betrachten Sie ein Wort  $x \in L(\mathcal{A})$ , das minimale Länge hat.
- Wie können Sie die Tatsache aus (a) benutzen um das Leerheitsproblem für reguläre Sprachen zu entscheiden?
- Zeigen Sie, dass  $L(\mathcal{A})$  genau dann unendlich ist, wenn es ein Wort  $x \in L(\mathcal{A})$  mit  $n \leq |x| < 2n$  gibt.  
Hinweis: Wenn die Sprache  $L(\mathcal{A})$  unendlich ist, gibt es Wörter die zu  $L(\mathcal{A})$  gehören und mindestens Länge  $n$  haben (warum?). Unter diesen, betrachten Sie ein Wort minimaler Länge.

**Musterlösung:**

- (a) Sei  $x$  ein Wort in  $L(\mathcal{A})$  von minimaler Länge. Wir zeigen, dass die Annahme  $|x| \geq n$  zu einem Widerspruch führt: ist  $|x| \geq n$ , dann gibt es eine Zerlegung  $x = u \cdot v \cdot w$ , so dass  $|u \cdot v| \leq n$ ,  $|v| > 0$  und  $u \cdot v^m \cdot w \in L(\mathcal{A})$  für jedes  $m \in \mathbb{N}$ . Insbesondere gehört für  $m = 0$  auch  $u \cdot w$  zur Sprache  $L(\mathcal{A})$ . Da aber  $|v| > 0$ , ist  $|uw| < |x|$  und das widerspricht der Minimalität von  $x$ . Wir schließen, dass die Länge von  $x$  kleiner als  $n$  sein muss.
- (b) Um zu entscheiden, ob die Sprache, die durch einen DFA mit  $n$  Zustände erkannt wird, leer ist, braucht man nur zu überprüfen, ob es ein Wort mit Länge kleiner als  $n$  gibt, das von diesem Automaten akzeptiert wird. Da es nur endlich viele solcher Wörter gibt, kann man einfach durch systematisches Durchsuchen feststellen, ob das zutrifft. Wenn es kein solches Wort gibt, ist die Sprache leer.
- (c)  $\Leftarrow$ : Sei  $x \in L(\mathcal{A})$  ein Wort mit  $n \leq |x| < 2n$ . Aufgrund des Pumping Lemmas gibt es eine Zerlegung  $x = u \cdot v \cdot w$  mit  $|v| > 0$  und  $u \cdot v^m \cdot w \in L(\mathcal{A})$  für jedes  $m \in \mathbb{N}$ . Da  $|v| > 0$ , sind alle Wörter der Form  $u \cdot v^m \cdot w$  verschieden, und deshalb ist die Sprache  $L(\mathcal{A})$  unendlich.

$\Rightarrow$ : Ist die Sprache  $L(\mathcal{A})$  unendlich, dann gibt es Wörter in  $L(\mathcal{A})$  von beliebiger Länge (einfach, weil die Anzahl von Wörtern bestimmter Länge endlich ist). Insbesondere gibt es Wörter, die mindestens die Länge  $2n$  haben. Sei  $y \in L(\mathcal{A})$  von minimaler Länge unter der Nebenbedingung  $|y| \geq 2n$  gewählt.

Wegen des Pumping Lemmas gibt es eine Zerlegung  $y = u \cdot v \cdot w$ , so dass  $0 < |v| \leq n$  und  $u \cdot w \in L(\mathcal{A})$ . Sei  $x = u \cdot w$ . Da  $x \in L(\mathcal{A})$  und  $|x| = |y| - |v| < |y|$  gilt  $|x| < 2n$  ( $x \geq 2n$  würde der Minimalität von  $y$  widersprechen). Andererseits ist  $|x| = |y| - |v| \geq 2n - n = n$ . Deshalb ist  $x$  wie gewünscht.