

# Imperative Programmierung, Aufgaben A01

Hauke Schümann, Martikelnr.: 219203901

November 7, 2019

## 1 Aufgabe 1: Turing Maschine

### 1.1 Teilaufgabe (a)

Zustand	Band	Zeigerposition
$S_1$	11000	1
$S_2$	01000	2
$S_2$	01000	3
$S_3$	01000	4
$S_4$	01010	3
$S_5$	01010	2
$S_5$	01010	1
$S_1$	11010	2
$S_2$	10010	3
$S_3$	10010	4
$S_3$	10010	5
$S_4$	10011	4
$S_4$	10011	3
$S_5$	10011	2
$S_1$	11011	3
$S_6$	11011	3

Man endet mit dem Band 11011.

### 1.2 Teilaufgabe (b)

Die beschriebene Maschine ist eine die im ersten Schritt terminiert.

$$\begin{aligned}\Sigma &= \{s_1\}, \\ A &= \{0, 1\}, \\ \sigma &= s_1, \\ F &= \{s_1\}, \\ \delta &= \emptyset\end{aligned}$$

## 2 Aufgabe 2: Syntax

Wir beginnen beim Startsymbol *Zuweisung* und bestimmen alle mögliche Ableitungen.

	Ausdruck	Nächste Regel
1.	Zuweisung	Regel 1
2.	Variable "a" := Ausdruck	Regel 4
2.	"a := " Ausdruck	Regel 5
3.	"a := " Ausdruck " + " Ausdruck	Regel 5 oder Regel 4
Fall 1.		Anwendung von Regel 5
4.	"a := a + " Ausdruck	Regel 4
5.	"a := a + " Ausdruck " + " Ausdruck	Regel 6
6.	"a := a + b + " Ausdruck	Regel 7
7.	"a := a + b + 1"	
Fall 2.		Anwendung von Regel 4
4.	"a := " Ausdruck " + " Ausdruck " + " Ausdruck	Regel 5
5.	"a := a + " Ausdruck " + " Ausdruck	Regel 6
6.	"a := a + b + " Ausdruck	Regel 7
7.	"a := a + b + 1"	

Es gibt zwei mögliche Ableitung.

## 3 Aufgabe 3: Nur ein Weg nach Rom.

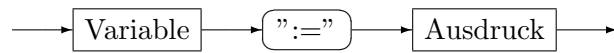
Wir nennen den in der Aufgabe geforderte Grammatik  $G_1 = (T_1, N_1, P_1, S_1)$  mit

$T_1 = \{ "a", "b", " := ", " + ", "1", "0" \}$   
 $N_1 = \{ \textit{Zuweisung}, \textit{Variable}, \textit{Ausdruck}, \textit{Konstante}, \textit{Wert} \}$   
 $P_1 = \{$   
 $\textit{Zuweisung} =_1 \textit{Variable} " := " \textit{Ausdruck},$   
 $\textit{Ausdruck} =_2 \textit{Wert},$   
 $\textit{Ausdruck} =_3 \textit{Wert} " + " \textit{Ausdruck},$   
 $\textit{Wert} =_4 \textit{Konstante},$   
 $\textit{Wert} =_5 \textit{Variable},$   
 $\textit{Konstante} =_6 "0",$   
 $\textit{Konstante} =_7 "1",$   
 $\textit{Variable} =_8 "a",$   
 $\textit{Variable} =_9 "b"$   
 $\}$

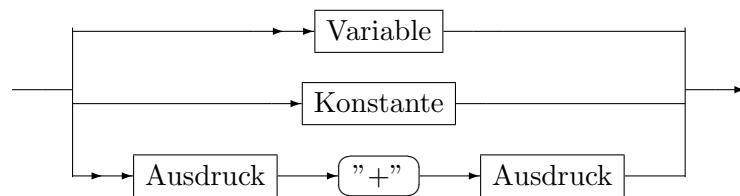
## 4 Aufgabe 4

### 4.1 Syntax Diagramme

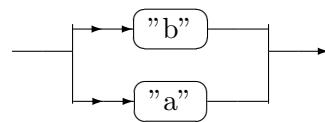
Zuweisung



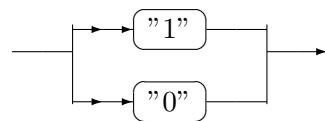
Ausdruck



Variable



Konstante



### 4.2 EBNF

*Zuweisung* = *Variable* " := " *Ausdruck*.

*Ausdruck* = ( *Variable* | *Konstante* ) [ " + " *Ausdruck* ].

*Variable* = ( " a " | " b " ).

*Konstante* = ( " 1 " | " 0 " ).

## 5 Aufgabe 5

### 5.1 EBNF

$Adresse = Text ";" (Text " " Zahl—"Postfach " Zahl) ";" Text ";" Zahl ";" ["Tel " Zahl ";" ] ["Fax " Zahl ";" ] ["Email " Email]$ .

$Email = Text "@" Text "." Text$ .

$Text = Buchstabe \{ Buchstabe \}$

$Zahl = Ziffer \{ Ziffer \}$ .

$Ziffer = (1|2|3|4|5|6|7|8|9|0)$

$Buchstabe = ('a'|'b'|...|'A'|...|'ä'|...|' '|...|Ziffer)$

Es gibt eine immer noch eine große Menge unzulässiger Kombinationen die man hiermit erschaffen könnte aber ich möchte hier nicht 30 Stunden sitzen.

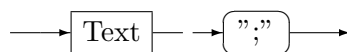
Zum Beispiel ist "2a" eine zulässige Hausnummer, der Einfachheit halber ist das jedoch nicht repräsentiert.

### 5.2 Syntaxdiagramme

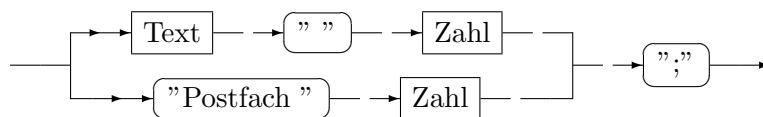
*Adresse*



*Name*



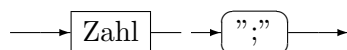
*Anschrift*



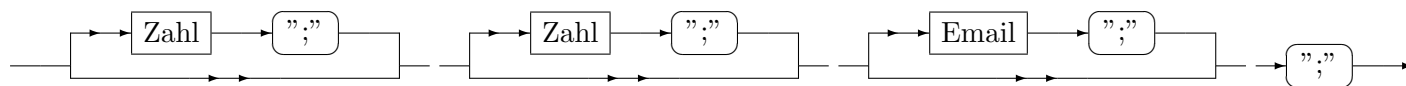
*Stadt*



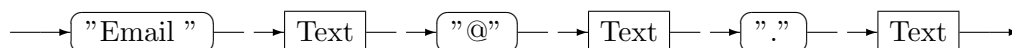
*PLZ*



*Optionales*



*Email*



## 6 Aufgabe 6

### 6.1 Algorithmusprüfung

Prüfen ob die Formel ein Algorithmus ist:

Operative Finitheit. Nicht gegeben, Formel läuft endlos weiter.

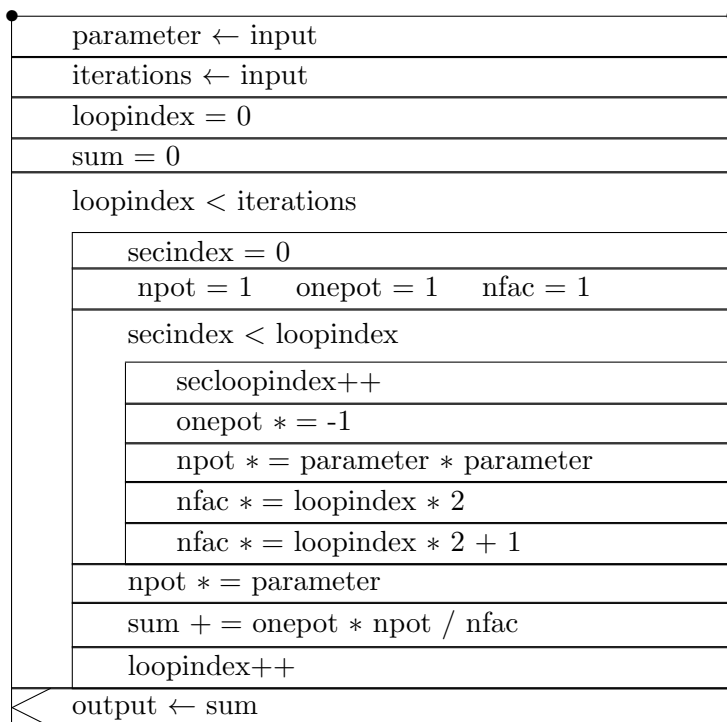
Effektivität. Es sind keine unrealisierbaren Elementaroperationen gefragt.

Vollständigkeit. Formel is vollständig. Es kann stets angegeben werden welche Operation durchgeführt werden soll.

Finitheit. Die Formel kann endlich dargestellt werden.

Dynamische Finitheit. Ergebnis kann irrational sein. Daher nicht gegeben.

### 6.2 Struktogramm



• Kann man das optimieren? Klar aber warum sollte ich? •