

# Serie 12

# Test

Name

Vorname

49  
Pt. total

→  
Note

- Punkteabzug oder keine Punkte für schwer verständliche oder unnötig komplizierte Lösungen.
- Hilfsmittel: ein A4-Blatt, beidseitig, selber geschrieben.
- Zeit: 80 Minuten.

## Aufgabe 1 (4x1 + 2 + 3 Pt.)

Pt. 9

Sie sehen eine Grammatik:

- Sentence ::= Subject Predicate
- Subject ::= Article Noun
- Predicate ::= Verb Object
- Object ::= Article Noun
- Article ::= the
- Noun ::= dog
- Noun ::= cat
- Verb ::= chases

(a) Von welchem Typ ist diese Grammatik?

Typ 2 = kontext-frei

①

(b) Welches sind die *terminals*?

the, dog, cat, chases (4 Stk.)

①

(c) Welches sind die *nonterminals*?

Sentence, Subject, Predicate, Object, Article, Noun, Verb (7 Stk.)

①

(d) Welches ist das Startsymbol?

Sentence

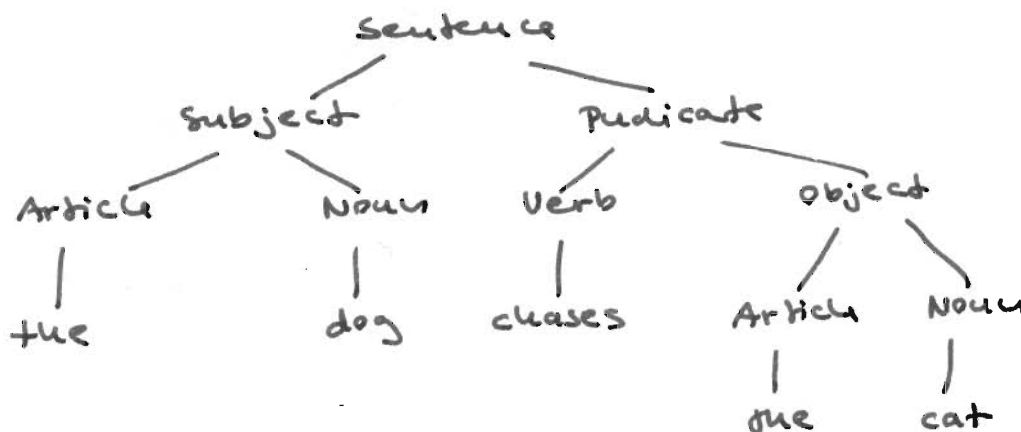
①

(e) Nennen Sie ein Wort der Sprache dieser Grammatik.

the dog chases the cat

②

(f) Zeichnen Sie den *derivation tree* Ihres Wortes von Teilaufgabe e.



③

Aufgabe 2 (4 + 4 x 1 Pt.)

- (a) Es gilt: Die reellen Zahlen zwischen 0.4 und 0.5 sind überabzählbar.  
Beweisen Sie diese Behauptung von Grund auf!

- Indirekter Beweis

- Annahme: Aufzählung:

0.47132865...	}	①
0.42119857...		
0.40021212...		
⋮		

- konstruiere neue Zahl:

0.4823 ...

} ①

- Kann nicht in der Aufzählung sein:  
Widerspruch!

- Einwandfreie Erkennung, alles beachtet, alle nicht wörtl. nicht näher folgt: 4 Pt.
- Beweis ohne Schritte vorlesen, aber weit alles stringent: 3 Pt.
- relevante Punkte: 1 Pt.

4

- (b) Von welchem Typ ist die Grammatik  $\{S \rightarrow aAB, \underline{AS} \rightarrow AAB, A \rightarrow a, B \rightarrow b\}$ ?

Typ 1 = kontext-sensitiv

1

- (c) Von welchem Typ ist die Grammatik  $\{S \rightarrow aB, B \rightarrow bB, B \rightarrow bA, \underline{A} \rightarrow \lambda, B \rightarrow b\}$ ?

Typ 2 = kontext-frei

1

- (d) Von welchem Typ ist die Grammatik  $\{S \rightarrow aB, B \rightarrow AB, \underline{aA} \rightarrow b, A \rightarrow b\}$ ?

Typ 0 = allgemein

1

- (e) Von welchem Typ ist die Grammatik  $\{S \rightarrow Bb, B \rightarrow a, \underline{aa} \rightarrow B, B \rightarrow a\}$ ?

Ist keine Grammatik

1

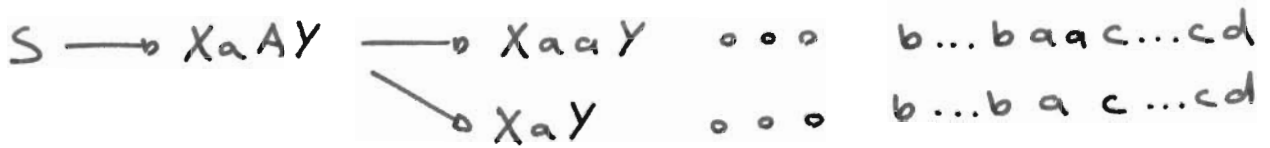
(a) - (c) : - 100% wert : 3 Pt.  
 - kleiner Fehler : 2 Pt.  
 - kleinere Punkte : 1 Pt.

Pt. 11

Aufgabe 3 (3+3+3+2 Pt.)

(a) Welche Sprache erzeugt folgende Grammatik:

$$\{S \rightarrow XaAY, A \rightarrow a, aA \rightarrow a, Y \rightarrow d, X \rightarrow Xb, Y \rightarrow cY, X \rightarrow \lambda\}$$



$$L = \{b^i a^j c^k d \mid i \geq 0, j = 1, 2, k \geq 0\}$$

3

(b) Konstruieren Sie eine kontextfreie Grammatik, die folgende Sprache erzeugt:

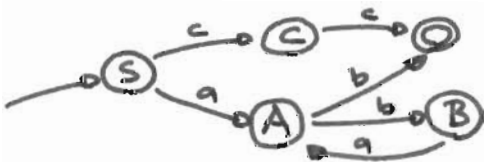
$$\{0^{2n}1^n \mid n = 0, 1, 2, \dots\}$$

$$S \rightarrow 00S1, S \rightarrow \lambda$$

3

(c) Finden Sie eine Typ-3 Grammatik, die folgende Sprache erzeugt:

$$\{cc, ab, abab, ababab, \dots\}$$



$$S \rightarrow cC, C \rightarrow c, S \rightarrow aA, A \rightarrow b, A \rightarrow bB, B \rightarrow aA$$

... nicht Typ 3; max. 1 Pt.

Vorsicht!  
 (Fehl: -1 Pt.)

Variante:  $S \rightarrow cC, C \rightarrow c$   
 $S \rightarrow A, A \rightarrow aB, B \rightarrow b, B \rightarrow bA$

3

(d) Was halten Sie von folgender Aussage?

Da man jede Grammatik als Syntax-Diagramm zeichnen kann, kann man *im Prinzip* jede Grammatik in einem Computer speichern. Dies kann jedoch für Typ-0 Grammatiken sehr mühsam sein.

Falsch: Nur Typ-2 und Typ-3 lassen sich als Syntax-Diagramm zeichnen!

2

Aufgabe 4 (3x1 + 2 + 3 + 2x1 + 2 Pt.)

Finden Sie reguläre Ausdrücke zu folgenden Sprachen in 0 und 1:

- (a)  $\{0, \lambda, 1, 11, 111, \dots\}$

0|1\*

- (b) Alle Bit-Strings, die genau zwei 0-er enthalten.

1\*01\*01\*

- (c)  $\{01^n0 \mid n \geq 2\}$

0111\*0

- (d) Alle Bit-Strings, in denen nirgends 00 vorkommt.

d.h. nach 0 muss eine 1 kommen:

(011)\*(\lambda10)

Idee: 1 Pt.

0 alleine oder 0 am Schluss!

- nur 2 Pt. falls korrek. Konzept  
- es können "00" vorkommen: 0 Pt.

- (e) Alle Bit-Strings, die nicht mit 00 enden.

d.h. enden mit 01, 10, 11 (... oder noch kürzer)

(011)\* (01|10|11) | (011)\lambda

= (011)\* (110)

2 Pt.

Welche Sprache erzeugen folgende regulären Ausdrücke?

- (f)  $0(0|1)^*1$

Alle Worte, die mit 0 anfangen und mit 1 aufhören.

- (g)  $0^*(0|1)^*1^*$

$A^*$  = alle Worte überhaupt

- (h)  $0^*(01^*01^*)^*$

Alle Worte, die mit beliebig vielen Nullen beginnen, gefolgt von einem Wort mit einer ungeraden Anzahl Nullen (oder  $\lambda$ ). (kann mit 0 nicht korrek.)

(f)+(g): nur richtig/falsch!

nur korrek. Antwort /  
falsch: 1/0 Pt.

1

1

1

2

3

1

1

2

Aufgabe 5 (2 + 3 + 4 Pt.)

- (a) Erklären Sie in wenigen Worten, was man unter einer *Meta Language* versteht und nennen Sie ein Beispiel einer *Meta Language*.

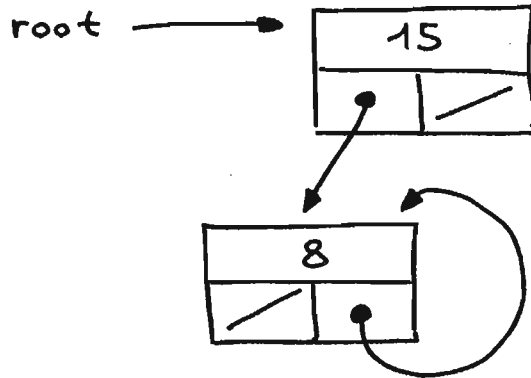
Eine Metasprache ist eine Sprache, die etwas über andere Sprachen aussagt, die z.B. andere Sprachen definiert. Unsere Beispiele: BNF, EBNF.

unabhängig: - Erklärung: 1 Pt.  
- Beispiel: 1 Pt.

2

- (b) Sie haben folgende Datenstruktur:

```
Class Tree {
    byte type;
    Tree left, right;
}
```



Wie lautet in *Java* der Code, der nebenstehende Datenstruktur erstellt?

```
Tree next = new Tree();
next.type = 8;
next.right = next;
Tree root = new Tree();
root.type = 15;
root.left = next;
```

Spez. komplett:  
- je Fehler / unvollständige / zu viel  
1 Pt. Abzug.

3

- (c) Schreiben Sie in *Java* die Methode zum Parsen der folgenden Zeile EBNF. Die Methoden zum Einlesen von Zeichen, Fehlerbehandlung, etc. sind bereits vorhanden. Bitte sauberes Top-Down-Parser Design, so wie in der Stunde gelernt!

$$S = "b" "x" \mid "b" S \mid \lambda$$

```
void SC() {
    if (ch == 'b') {
        nextchar();
        if (ch == 'x') nextchar(); else SC();
    }
}
```

- 100% korrekt: 4 Pt.  
- Details (z.B. ";" vergessen): 3 Pt.  
- ein großer Fehler: 2 Pt.  
- mehr als ein großer Fehler / Ausfall vorhanden: 1 Pt.

4