

22. Die Produktionen einer natürlichen Sprache seien in der BNF-Notation gegeben durch

```

<SATZ>           ::= <NOMINALGRUPPE> <VERBALGRUPPE>
<NOMINALGRUPPE> ::= <ARTIKEL> <SUBSTANTIV> | <SUBSTANTIV>
<VERBALGRUPPE>  ::= <VERB> <NOMINALGRUPPE>
<VERB>            ::= jagt | sieht | beißt | frißt
<ARTIKEL>          ::= der | die | das
<SUBSTANTIV>       ::= Katze | Maus | Merlin | Tablet

```

Zeige (Syntaxbaum!), daß

- a) „Merlin sieht das Tablet“
- b) „die Maus beißt der Katze“

korrekte Sätze gemäß obenstehenden Syntaxregeln sind.

23. Eine einfache *formale Sprache* **L(G)** ist durch folgende Grammatik **G** gegeben:

T = {a, b}

N = {**S**}; **S** ist Startzeichen

Produktionen **P**:

- (1) **S** ::= a
- (2) **S** ::= a **S** a
- (3) **S** ::= **S** b

Bemerkung: Die in der BNF-Notation für Nonterminals vorgesehenen spitzen Klammern wurden hier weggelassen.

Zeige (Linksableitung und Syntaxbaum):

Die Wörter a) abbb , b) aabbab gehören zu **L(G)**.

24. Die Grammatik **G** = (**T**, **N**, **P**, **S**) sei gegeben durch

T := {x, +, (,)}

N := {A, B, C, **S**}

Produktionen **P**:

- (1) **S** ::= x | (B)
- (2) B ::= S C
- (3) C ::= + S | ε

Erzeuge das Wort (Linksableitung!): ((x + (x + x)))

25. Lexikalische Analyse von Namen (Bezeichner, identifier):

T := {a, b, . . . , z, A, B, . . . , Z, _, 0, 1, . . . , 9}

N := {<name>, <buchstabe>, <ziffer>}

Startzeichen: **S** = <name> (beachte: **S** ∈ **N**)

Produktionen **P**:

- (1) <name> ::= <buchstabe> | <name> <buchstabe> | <name> <ziffer>
- (2) <buchstabe> ::= a | b | c | . . . | z | _ | A | B | C | . . . | Z
- (3) <ziffer> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Zeige

a) mittels Linksableitung („top-down“), b) mittels Syntaxbaum („bottom-up“):

Das Wort a3Xyz ist ein gültiger Name.

Bemerkung: In Python wie in anderen Programmiersprachen sind Schlüsselwörter oder reservierte Wörter (z. B. except, if, elif, else, . . .) als Bezeichner unzulässig.