

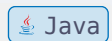
Exercise 01

01 Grundbegriffe

Task a

Wie lang (in Terminalsymbolen) ist der kürzeste Satz in MicroJava, der mindestens ein Tilde (“~”) enthält? Geben Sie ein Beispiel für einen Satz dieser Länge an. Der Satz muss nur syntaktisch korrekt (entspricht der Grammatik) sein, nicht semantisch (hält alle Kontextbedingungen der Sprache ein). Beispielsweise ist $x = x + 1$, ohne x zuvor zu definieren, eine syntaktisch korrekte Zuweisung, semantisch korrekt ist sie aber nicht.

```
1 x[~1]++;
```



The statement above contains the following tokens:

- x (iden, terminal)
- “[” (literal, terminal)
- “~” (literal, terminal)
- “1” (number, terminal)
- “]” (literal, terminal)
- “++” (literal, terminal)
- “;” (literal, terminal)

To sum it up the statement has a ~ and has a terminal symbol count of 7.

Task b

Sind die Nonterminalsymbole Statement und ActPars rekursiv? Wenn ja, geben Sie alle Rekursionsarten an (Transitivität: direkt, indirekt; Orientierung: links, rechts, zentral). Beispiele für Rekursionsarten sind direkt zentralrekursiv, indirekt linksrekursiv, . . . Pro NT kann es mehrere Rekursionsarten geben. Geben Sie bei indirekten Rekursionen einen möglichen Pfad an (Beispiel: Factor ist indirekt zentralrekursiv über Factor $\rightarrow$ Expr $\rightarrow$ Term $\rightarrow$ Factor).

Statement

The relevant Productions for the Statement Symbol are:

```
1 Statement = Designator ( Assignop Expr | ActPars | "++" | "--" ) ";"
2 | "if" "(" Condition ")" Statement [ "else" Statement ]
3 | "while" "(" Condition ")" Statement
4 | "break" ";"
5 | "return" [ Expr ] ";"
6 | "read" "(" Designator ")" ";"
7 | "print" "(" Expr [ "," number ] ")" ";"
8 | Block
9 | ";"
10
11 Block = "{" { Statement } "}";
```

We have the following Recursions:

- line 2: Statement is directly central recursive

- line 3: Statement is directly right recursive
- line 8: Block is indirectly central recursive

ActPars

The relevant Productions for the ActPars Symbol are:

```
1  ActPars = "(" [ Expr { "," Expr } ] ")" .
2
3  Expr = [ "-" ] Term { Addop Term } .
4
5  Term = Factor { Mulop Factor } .
6
7  Factor = Designator [ ActPars ]
8  | number
9  | charConst
10 | "new" ident [ "[" Expr "]" ]
11 | "(" Expr ")" .
```

We have the following Recursion:

- line 1: Factor is indirectly central recursive

Task c