rules

```
──┬─▶│ directiveLine │─┬─▶
  │                    │
  └─▶│ program │───────┘
```
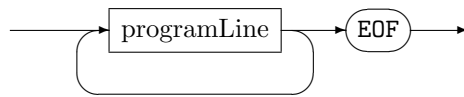
directiveLine

```
──┬─▶│ command │──┬─▶( kwVK )─▶
  │               │
  └──( lexSmcl )◀─┘
```

program

```
──┬─▶│ programLine │─┬─▶( EOF )─▶
  │                  │
  └──────────────────┘
```

programLine

```
──▶│ lineNum │─┬───────────────┬─▶( kwVK )─▶
              │                │
              └─┬─▶│ command │─┘
                │              │
                └──( lexSmcl )◀┘
```

lineNum

```
──▶│ grpNum │─▶( lexPeriod )─▶( lexInt )─▶
```

grpNum

```
──▶( lexInt )─▶
```

1

command

```
         ┌─ set ─────┐
         ├─ ask ─────┤
         ├─ type ────┤
         ├─ write ───┤
         ├─ goto ────┤
         ├─ if ──────┤
         ├─ for ─────┤
         ├─ quit ────┤
         ├─ do ──────┤
         ├─ return ──┤
         ├─ comment ─┤
         ├─ erase ───┤
         ├─ modify ──┤
         └─ xecute ──┘
```

set

```
──▶( S )──┬──────────▶ var ─▶(lexEQ)─▶ expr ──▶
          └──▶( ET )──┘
```

var

```
──▶(lexId)──┬───────────────────────────────────────────▶
            └─▶(lexLftBr)─▶ indExpr ──┬──────────────────┬─▶(lexRghBr)─┘
                                      └─▶(lexComma)─▶ indExpr ─┘
```

indExpr

```
        ┌─────────────┐
───────→│   lexInt    │───────→
        ├─────────────┤
        │     var     │
        ├─────────────┤
        │    expr     │
        └─────────────┘
```

expr

```
────→│ addExpr │────→
```

addExpr

```
───→│ mulExpr │──────────────────────────────────────→
         ┌──────────┐    ┌──────────┐
         │ lexPlus  │───→│ mulExpr  │
         ├──────────┤
         │ lexMinus │
         └──────────┘
```

mulExpr

```
───→│ powExpr │──────────────────────────────────────→
         ┌──────────┐    ┌──────────┐
         │  lexMul  │───→│ powExpr  │
         ├──────────┤
         │  lexDiv  │
         └──────────┘
```
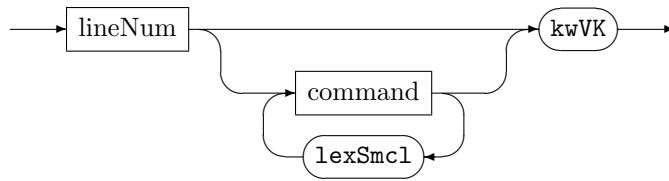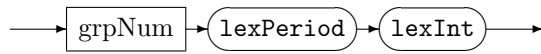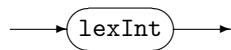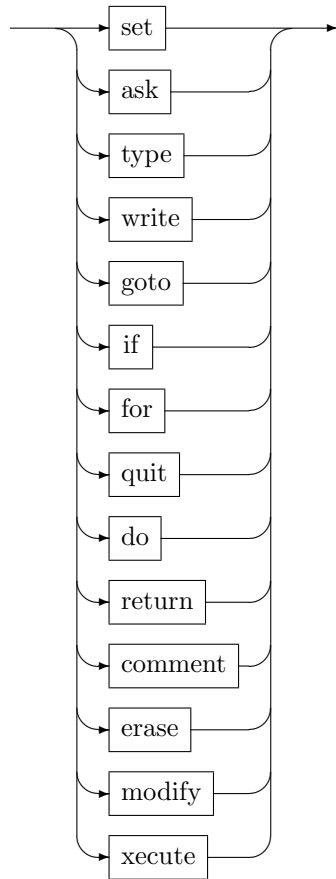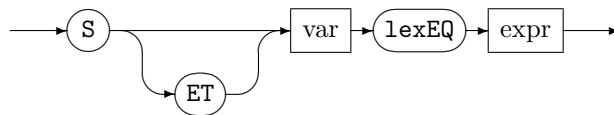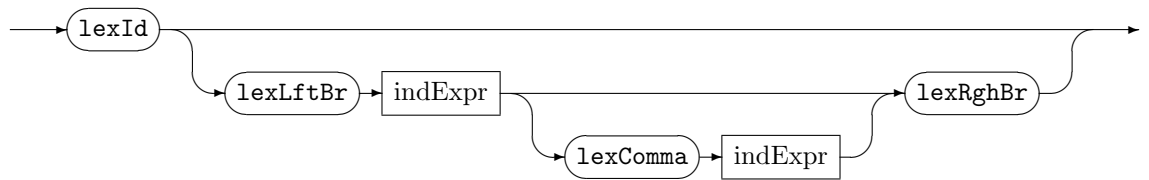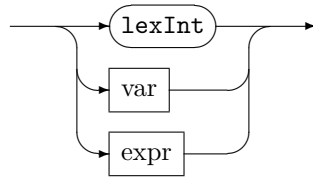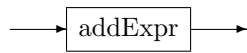
powExpr
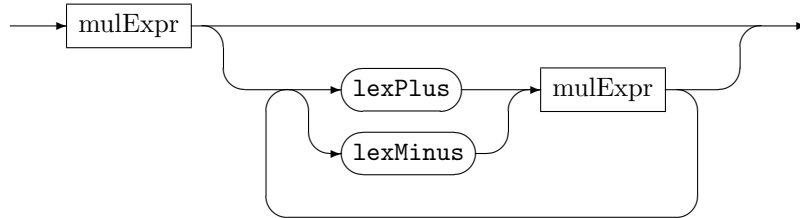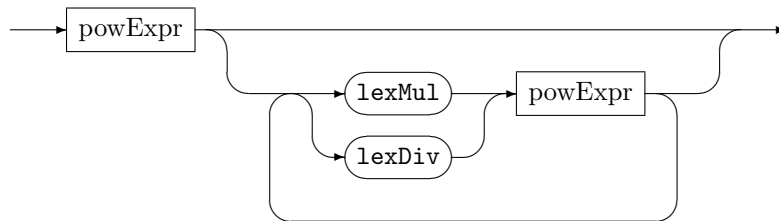
```
───→│ unaryExpr │──────────────────────────────────→
          ┌──────────┐   ┌───────────┐
          │ lexCaret │──→│ unaryExpr │
          └──────────┘
```

unaryExpr

```
──────────────────────────→│ primaryExpr │────→
        ┌──────────┐
        │ lexPlus  │
        ├──────────┤
        │ lexMinus │
        └──────────┘
```

primaryExpr
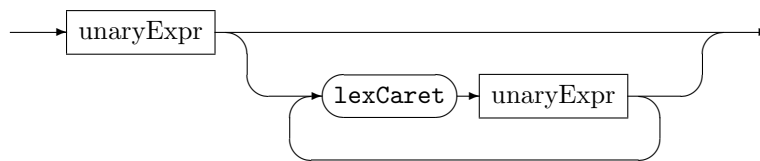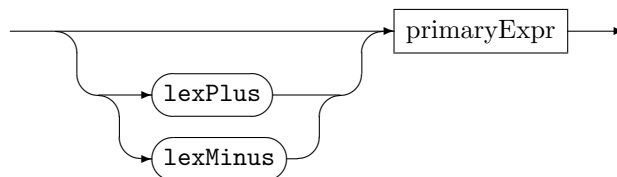
```
        ┌─( lexReal )─────────────────────┐
        │                                 │
        ├─[ var ]─────────────────────────┤
        │                                 │
        ├─[ function ]────────────────────┤
        │                                 │
        └─( lexLftBr )─[ expr ]─( lexRghBr )─┘
```

function

```
─[ nameFunction ]─( lexLftBr )─┬─────────────────┬─( lexRghBr )─
                               └─┬─[ expr ]─┬────┘
                                 └─( lexComma )─┘
```

nameFunction

```
        ┌─( kwFABS )─┐
        ├─( kwFCHR )─┤
        ├─( kwFCOS )─┤
        ├─( kwFITR )─┤
        ├─( kwFLOG )─┤
        ├─( kwFRAN )─┤
        ├─( kwFSBR )─┤
        ├─( kwFSGN )─┤
        ├─( kwFSQT )─┤
        ├─( kwFSIN )─┤
        └─( kwFX )───┘
```

4
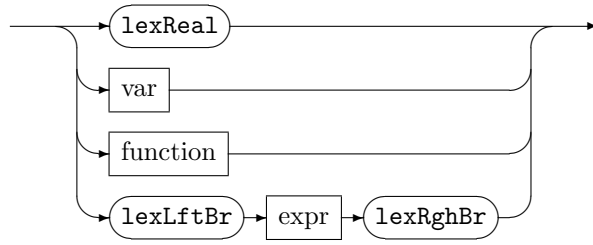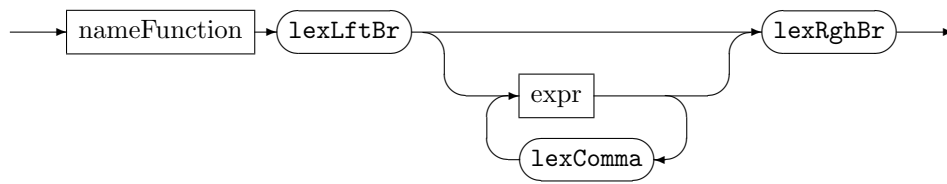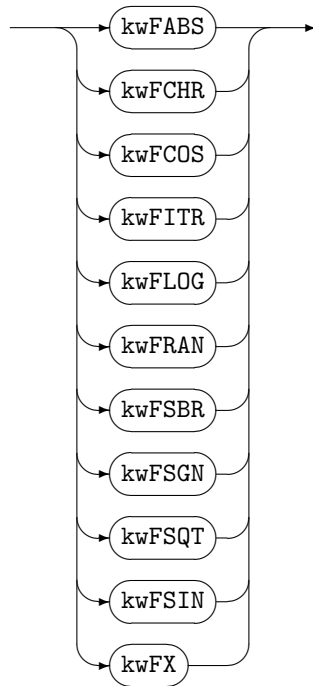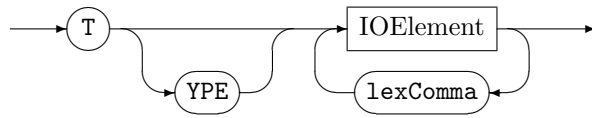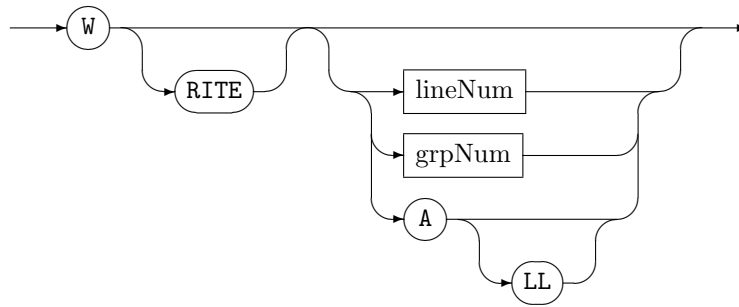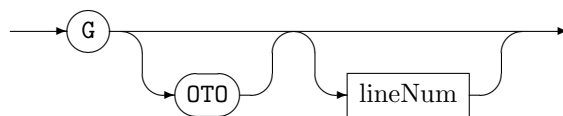
ask
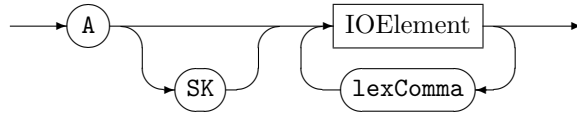
IOElement

type

write

goto

5

if

```
──●(I)──┬──────────┬──(lexLftBr)──▶(expr)──(lexRghBr)──┐
        └──●(F)──┘                                      │
   ┌───────────────────────────────────────────────────┘
   └──┬──[lineNum]──┬─────────────────────────┬──────────────────────────────▶
                    └──(lexComma)──[lineNum]──┘   └──(lexComma)──[lineNum]──┘
```

for

```
──●(F)──┬──────────┬──[var]──(lexEQ)──┐
        └──●(OR)──┘                    │
   ┌─────────────────────────────────┘
   └──┬──[expr]──┬─────────────────────┬──────────────────────┐
                 └──(lexComma)──[expr]──┘  └──(lexComma)──[expr]──┘
   ┌──────────────────────────────────────────────────────────┘
   └──(lexSmcl)──┬───────────────────┬──▶
                 └──[command]──┘
```
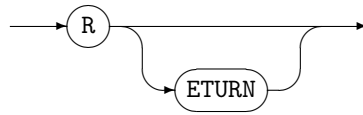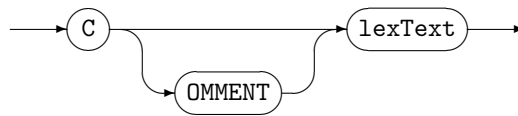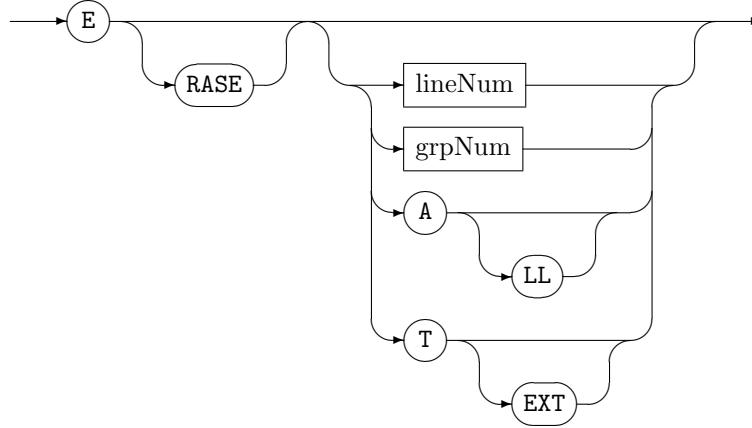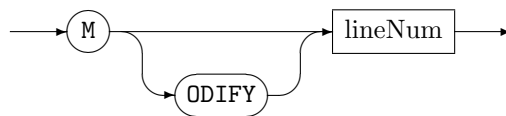
quit

```
──●(Q)──┬──────────┬──▶
        └──(UIT)──┘
```

6

do

return

comment

erase

modify

xecute