

CSC 600-01 (SECTION 1)  
**Homework 1 - Syntax**  
*prepared by Ilya Kopyl*

# CSC 600 HOMEWORK 1 - SYNTAX

February 15, 2018

*Homework is prepared by: Ilya Kopyl.*

*It is formatted in LaTeX, using TeXShop editor (under GNU GPL license).*

*Syntax diagrams are created in LucidChart online editor (lucidchart.com).*

## 1. Using BNF write the syntax definitions of the following objects:

a) Natural number (1, 2, 3, ...). The answer:

$$\begin{aligned}\langle \textit{natural number} \rangle &::= \langle \textit{non-zero digit} \rangle \mid \langle \textit{natural number} \rangle \langle \textit{digit} \rangle \\ \langle \textit{digit} \rangle &::= 0 \mid \langle \textit{non-zero digit} \rangle \\ \langle \textit{non-zero digit} \rangle &::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\end{aligned}$$

b) Unsigned integer (0, 1, 2, 3, ...). The answer:

$$\begin{aligned}\langle \textit{unsigned integer} \rangle &::= \langle \textit{digit} \rangle \mid \langle \textit{unsigned integer} \rangle \langle \textit{digit} \rangle \\ \langle \textit{digit} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\end{aligned}$$

Example of BNF definition of unsigned integer in languages that do not support leading zeroes (e.g. Python):

$$\begin{aligned}\langle \textit{unsigned integer} \rangle &::= 0 \mid \langle \textit{natural number} \rangle \\ \langle \textit{natural number} \rangle &::= \langle \textit{non-zero digit} \rangle \mid \langle \textit{natural number} \rangle \langle \textit{digit} \rangle \\ \langle \textit{digit} \rangle &::= 0 \mid \langle \textit{non-zero digit} \rangle \\ \langle \textit{non-zero digit} \rangle &::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\end{aligned}$$

c) Integer (... , -2, -1, 0, 1, 2, ...). The answer:

$\langle integer \rangle$	$::= \langle sign \rangle \langle unsigned integer \rangle$
$\langle sign \rangle$	$::= + \mid - \mid \langle empty \rangle$
$\langle empty \rangle$	$::=$
$\langle unsigned integer \rangle$	$::= \langle digit \rangle \mid \langle unsigned integer \rangle \langle digit \rangle$
$\langle digit \rangle$	$::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Example of BNF definition of an integer in languages that do not support leading zeroes (e.g. Python):

$\langle integer \rangle$	$::= \langle sign \rangle \langle unsigned integer \rangle$
$\langle sign \rangle$	$::= + \mid - \mid \langle empty \rangle$
$\langle empty \rangle$	$::=$
$\langle unsigned integer \rangle$	$::= 0 \mid \langle natural number \rangle$
$\langle natural number \rangle$	$::= \langle non-zero digit \rangle \mid \langle natural number \rangle \langle digit \rangle$
$\langle digit \rangle$	$::= 0 \mid \langle non-zero digit \rangle$
$\langle non-zero digit \rangle$	$::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

d) Odd number (... , -3, -1, 1, 3, ..., 101, ..., 2047, ...). The answer:

$\langle odd number \rangle$	$::= \langle sign \rangle \langle unsigned odd number \rangle$
$\langle sign \rangle$	$::= + \mid - \mid \langle empty \rangle$
$\langle empty \rangle$	$::=$
$\langle unsigned odd number \rangle$	$::= \langle odd digit \rangle \mid \langle unsigned integer \rangle \langle odd digit \rangle$
$\langle unsigned integer \rangle$	$::= \langle digit \rangle \mid \langle unsigned integer \rangle \langle digit \rangle$
$\langle digit \rangle$	$::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Example of BNF definition of an odd number in languages that do not support leading zeroes (e.g. Python):

```

<odd number>      ::= <sign> <unsigned odd number>
<sign>            ::= + | - | <empty>
<empty>           ::=
<unsigned odd number> ::= <odd digit> | <natural number> <odd digit>
<natural number>  ::= <non-zero digit> | <natural number> <digit>
<digit>           ::= 0 | <non-zero digit>
<non-zero digit>  ::= 2 | 4 | 6 | 8 | <odd digit>
<odd digit>       ::= 1 | 3 | 5 | 7 | 9

```

e) Even number (... , -4, -2, 0, 2, 4, ..., 332, ..., 1022, ...). The answer:

```

<even number>      ::= <sign> <unsigned even number>
<sign>            ::= + | - | <empty>
<empty>           ::=
<unsigned even number> ::= <even digit> | <unsigned integer> <even digit>
<unsigned integer>  ::= <digit> | <unsigned integer> <digit>
<digit>            ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Example of BNF definition of an even number in languages that do not support leading zeroes (e.g. Python):

```

<even number>      ::= <sign> <unsigned even number>
<sign>            ::= + | - | <empty>
<empty>           ::=
<unsigned even number> ::= <even digit> | <natural number> <even digit>
<natural number>  ::= <non-zero digit> | <natural number> <digit>
<digit>           ::= 0 | <non-zero digit>
<non-zero digit>  ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<even digit>      ::= 0 | 2 | 4 | 6 | 8

```

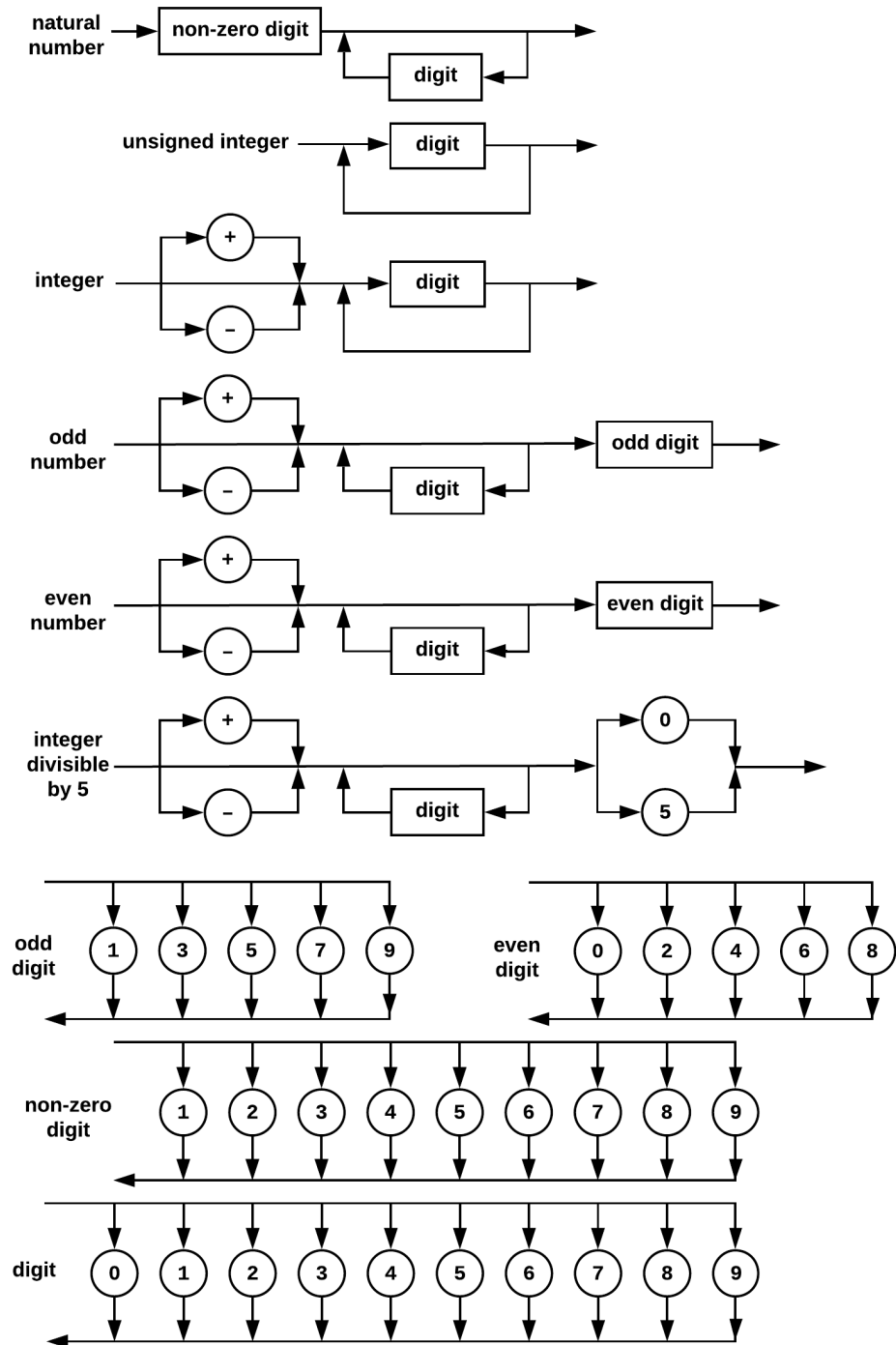
f) Integer divisible by five (... , -10, 5, 0, 5, 10, ...). The answer:

$$\begin{aligned}
\langle \text{integer div-by-5} \rangle &::= \langle \text{sign} \rangle \langle \text{unsigned int div-by-5} \rangle \\
\langle \text{sign} \rangle &::= + \mid - \mid \langle \text{empty} \rangle \\
\langle \text{empty} \rangle &::= \\
\langle \text{unsigned int div-by-5} \rangle &::= \langle \text{div-by-5 suffix} \rangle \mid \langle \text{unsigned integer} \rangle \langle \text{div-by-5 suffix} \rangle \\
\langle \text{unsigned integer} \rangle &::= \langle \text{digit} \rangle \mid \langle \text{unsigned integer} \rangle \langle \text{digit} \rangle \\
\langle \text{div-by-5 suffix} \rangle &::= 0 \mid 5 \\
\langle \text{digit} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}$$

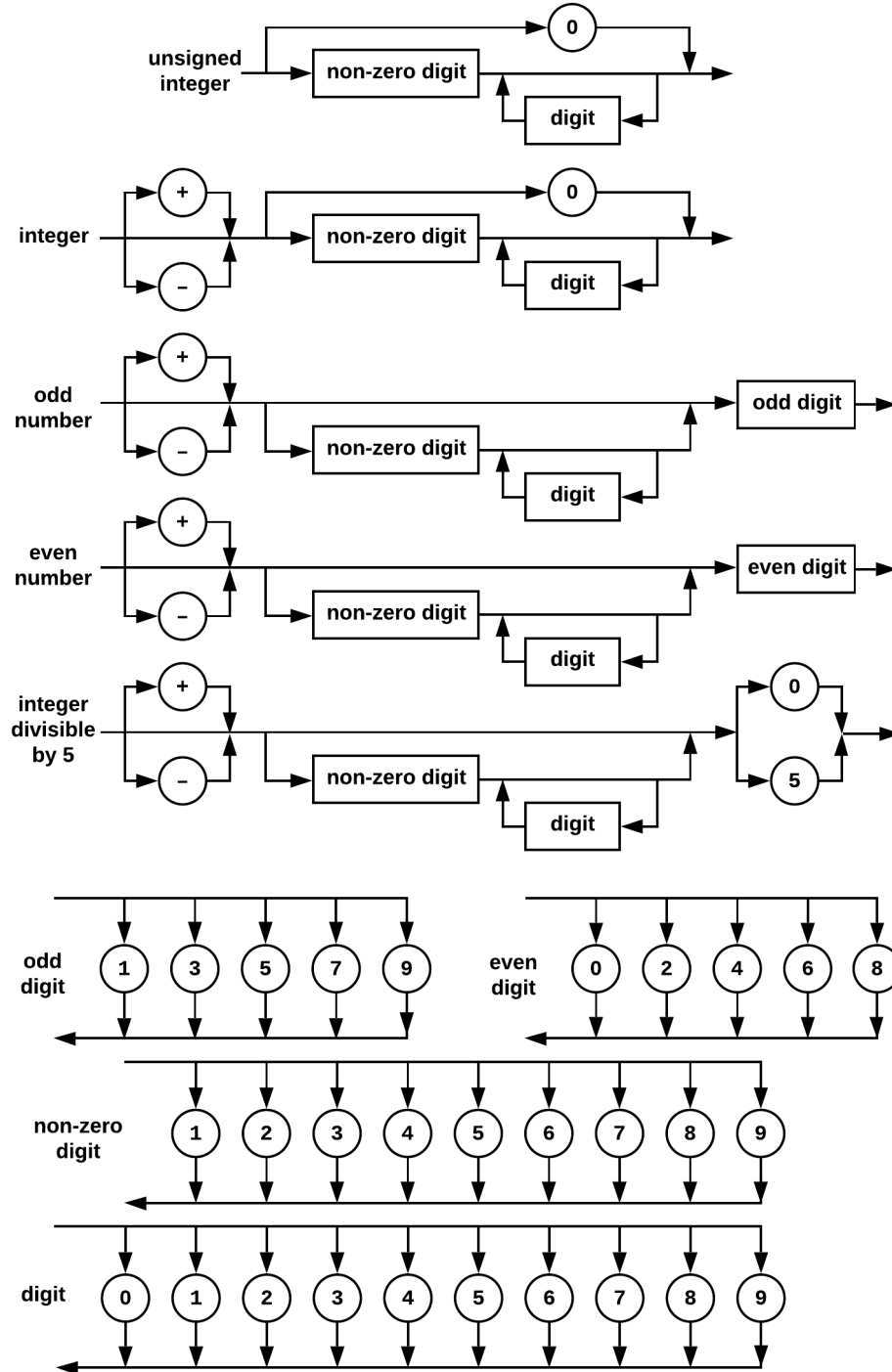
Example of BNF definition of an integer divisible by 5 in languages that do not support leading zeroes (e.g. Python):

$$\begin{aligned}
\langle \text{integer div-by-5} \rangle &::= \langle \text{sign} \rangle \langle \text{unsigned int div-by-5} \rangle \\
\langle \text{sign} \rangle &::= + \mid - \mid \langle \text{empty} \rangle \\
\langle \text{empty} \rangle &::= \\
\langle \text{unsigned int div-by-5} \rangle &::= \langle \text{div-by-5 suffix} \rangle \mid \langle \text{natural number} \rangle \langle \text{div-by-5 suffix} \rangle \\
\langle \text{natural number} \rangle &::= \langle \text{non-zero digit} \rangle \mid \langle \text{natural number} \rangle \langle \text{digit} \rangle \\
\langle \text{div-by-5 suffix} \rangle &::= 0 \mid 5 \\
\langle \text{digit} \rangle &::= 0 \mid \langle \text{non-zero digit} \rangle \\
\langle \text{non-zero digit} \rangle &::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}$$

2. Show syntax diagrams for questions (a), ..., (f) of problem 1.



Example of syntax diagrams for integers with no support of leading zeroes.



**3. Write a BNF definition of the syntax of (all possible) input statements in C++.**

Following is an example of input statement in C++:

```
cin » sclr » vec[2 * i - 1] » mat[f(i)][j + k] » t[i/3][j][k];
```

The answer:

$\langle \text{input statement} \rangle$	$::= \text{cin } \langle \text{input arguments} \rangle ;$
$\langle \text{input arguments} \rangle$	$::= \gg \langle \text{input value} \rangle \mid \langle \text{input arguments} \rangle \gg \langle \text{input value} \rangle$
$\langle \text{input value} \rangle$	$::= \langle \text{struct member} \rangle \mid \langle \text{array element} \rangle \mid \langle \text{identifier} \rangle$
$\langle \text{struct member} \rangle$	$::= \langle \text{identifier} \rangle . \langle \text{identifier} \rangle$ $\mid \langle \text{struct member} \rangle . \langle \text{identifier} \rangle$
$\langle \text{array element} \rangle$	$::= \langle \text{identifier} \rangle \langle \text{array indices} \rangle$
$\langle \text{identifier} \rangle$	$::= \langle \text{non-digit character} \rangle \mid \langle \text{identifier} \rangle \langle \text{digit} \rangle$
$\langle \text{array indices} \rangle$	$::= \langle \text{array index} \rangle \mid \langle \text{array indices} \rangle \langle \text{array index} \rangle$
$\langle \text{array index} \rangle$	$::= [ \langle \text{numerical expression} \rangle ]$
$\langle \text{numerical expression} \rangle$	$::= \langle \text{arithmetic expression} \rangle$ $\mid \langle \text{unary expression} \rangle$ $\mid \langle \text{function call} \rangle$
$\langle \text{arithmetic expression} \rangle$	$::= \langle \text{operand} \rangle$ $\mid \langle \text{arithmetic expression} \rangle \langle \text{operator} \rangle \langle \text{arithmetic expression} \rangle$
$\langle \text{operator} \rangle$	$::= + \mid - \mid * \mid / \mid \%$
$\langle \text{unary expression} \rangle$	$::= ++ \langle \text{operand} \rangle \mid \langle \text{operand} \rangle ++$ $\mid -- \langle \text{operand} \rangle \mid \langle \text{operand} \rangle --$
$\langle \text{operand} \rangle$	$::= \langle \text{identifier} \rangle \mid \langle \text{integer number} \rangle \mid \langle \text{floating point number} \rangle$
$\langle \text{integer number} \rangle$	$::= \langle \text{digits} \rangle \mid \langle \text{digits} \rangle \text{L} \mid \langle \text{digits} \rangle \text{LL}$
$\langle \text{floating point number} \rangle$	$::= \langle \text{real number} \rangle \mid \langle \text{real number} \rangle \text{F}$
$\langle \text{real number} \rangle$	$::= \langle \text{digits} \rangle . \langle \text{digits} \rangle \mid \langle \text{digits} \rangle . \mid . \langle \text{digits} \rangle$
$\langle \text{function call} \rangle$	$::= \langle \text{identifier} \rangle ( \langle \text{function arguments} \rangle )$
$\langle \text{function arguments} \rangle$	$::= \langle \text{argument} \rangle \mid \langle \text{function arguments} \rangle , \langle \text{argument} \rangle$
$\langle \text{argument} \rangle$	$::= \langle \text{function call} \rangle \mid \langle \text{expression} \rangle$



$\langle expression \rangle$	$::= \langle numerical\ expression \rangle$   $\langle string \rangle$   $\langle character\ literal \rangle$   $\langle void \rangle$
$\langle string \rangle$	$::= " \langle characters \rangle "$
$\langle character\ literal \rangle$	$::= ' \langle char \rangle ' \mid ' \backslash \langle char \rangle '$
$\langle characters \rangle$	$::= \langle char \rangle \mid \langle characters \rangle \langle char \rangle$
$\langle char \rangle$	$::= \langle non-digit\ character \rangle \mid \langle digit \rangle \mid \langle whitespace \rangle$
$\langle whitespace \rangle$	$::= ' '$
$\langle non-digit\ character \rangle$	$::= A \mid B \mid C \mid D \mid E \mid F \mid G \mid H \mid I \mid J$   $K \mid L \mid M \mid N \mid O \mid P \mid Q \mid R \mid S \mid T$   $U \mid V \mid W \mid X \mid Y \mid Z \mid a \mid b \mid c \mid d$   $e \mid f \mid g \mid h \mid i \mid j \mid k \mid l \mid m \mid n$   $o \mid p \mid q \mid r \mid s \mid t \mid u \mid v \mid w \mid x$   $y \mid z \mid \_$
$\langle digits \rangle$	$::= \langle digit \rangle \mid \langle digits \rangle \langle digit \rangle$
$\langle digit \rangle$	$::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
$\langle void \rangle$	$::= \langle empty \rangle$
$\langle empty \rangle$	$::=$

#### 4. Write a BNF definition of the syntax of (all possible) output statements in C++.

Following is an example of output statement in C++:

```
cout << 12.34 * a / rate << " " << 43.21 << " "
<< alpha + x[2*i-1] << " " << (p && q) << " "
<< pow(t[i][j],1.2) << " string " << 's'
<< " " << myfun(x, sin(x+y), third_argument) ;
```

The answer (the definitions for auxiliary BNF productions are listed in the previous answer):

$\langle output\ statement \rangle$	$::= \text{cout } \langle output\ arguments \rangle ;$
$\langle output\ arguments \rangle$	$::= \langle < output\ value \rangle \mid \langle output\ arguments \rangle \langle < output\ value \rangle$
$\langle output\ value \rangle$	$::= \langle expression \rangle$