

Seminar 1 – Programming Languages' Specification

BNF (Backus-Naur Form)

Constructs:

1. Meta-linguistic variables (non-terminals) - written between < >
2. Language primitives (terminals) - no special delimiters
3. Meta-linguistic connectors

::= equals by definition

| alternative/OR

<construct> ::= expr_1 | expr_2 | ... | expr_n

Ex.1: Specify, using BNF, all nonempty sequences of letters

Onita Andrei

LetterSeq ::= ["Lorem ipsum"]

Petcu Dragos

<LetterSeq> ::= A | B | C | ... | Z

Oana Nourescu

<LetterSeq> ::= <Letter> | <Letter> <LetterSeq>

<Letter> ::= A|B|...|Z |a |b|...|z

Ex.2: Specify, using BNF, both signed and unsigned integers, with the following constraints:

0 does not have a sign

numbers of at least two digits do not start with 0

Neta Razvan

$\langle \text{Number} \rangle ::= 0 \mid \langle \text{sign} \rangle \langle \text{non zero digit} \rangle \mid \langle \text{non zero digit} \rangle \langle \text{Number} \rangle$

Onita Andrei

$\langle \text{Number} \rangle ::= 0 \mid \langle \text{sign} \rangle \langle \text{non zero digit} \rangle \mid \langle \text{non zero digit} \rangle \mid \langle \text{sign} \rangle \langle \text{non zero digit} \rangle \langle \text{Number} \rangle \mid \langle \text{non zero digit} \rangle \langle \text{Number} \rangle$

Neta Razvan

$\langle \text{Int} \rangle ::= 0 \mid \langle \text{sign} \rangle \langle \text{DigitSeq} \rangle$

Onita Andrei

$\langle \text{Int} \rangle ::= 0 \mid \langle \text{sign} \rangle \langle \text{Number} \rangle \mid \langle \text{Number} \rangle$

$\langle \text{Number} \rangle ::= \langle \text{Non0digit} \rangle \mid \langle \text{Number} \rangle \langle \text{digit} \rangle$

$\langle \text{digit} \rangle ::= 1 \mid 2 \mid 3 \mid \dots \mid 9$

$\langle \text{Non0digit} \rangle ::= 0 \mid \langle \text{digit} \rangle$

Petcu Dragos

$\langle \text{sign} \rangle ::= + \mid -$

$\langle \text{finalNumber} \rangle ::= 0 \mid \langle \text{sign} \rangle \langle \text{number} \rangle \mid \langle \text{number} \rangle$

$\langle \text{number} \rangle ::= \langle \text{nonZeronumber} \rangle \mid \langle \text{nonZeronumber} \rangle \langle \text{anyDigit} \rangle \mid \langle \text{number} \rangle \langle \text{anyDigit} \rangle$

$\langle \text{anyDigit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{digitSequence} \rangle ::= \langle \text{anyDigit} \rangle \mid \langle \text{anyDigit} \rangle \langle \text{digitSequence} \rangle$

$\langle \text{nonZeroNumber} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Miclea George

$\langle \text{Integer} \rangle ::= \langle \text{Digit} \rangle \mid \langle \text{Minus} \rangle \langle \text{NonZeroDigit} \rangle \mid \langle \text{Minus} \rangle \langle \text{NonZeroDigit} \rangle \langle \text{DigitSeq} \rangle \mid \langle \text{NonZeroDigit} \rangle \langle \text{DigitSeq} \rangle$

$\langle \text{DigitSeq} \rangle ::= \langle \text{Digit} \rangle \mid \langle \text{Digit} \rangle \langle \text{DigitSeq} \rangle$

$\langle \text{NonZeroDigit} \rangle ::= 1 \mid 2 \mid \dots \mid 9$

$\langle \text{Digit} \rangle ::= 0 \mid \langle \text{NonZeroDigit} \rangle$

Onita Andrei (wip)

$\langle \text{sign} \rangle ::= + \mid -$

$\langle \text{finalNumber} \rangle ::= 0 \mid \langle \text{sign} \rangle \langle \text{number} \rangle \mid \langle \text{number} \rangle$

$\langle \text{number} \rangle ::= \langle \text{digitSequence} \rangle$

$\langle \text{anyDigit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{digitSequence} \rangle ::= \langle \text{nonZeroNumber} \rangle \mid \langle \text{digitSequence} \rangle \langle \text{anyDigit} \rangle$

$\langle \text{nonZeroNumber} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Moldovanu Dragos

$\langle \text{Number} \rangle ::= 0 \mid \langle \text{non-zero-digit} \rangle \langle \text{NumberSeq} \rangle \mid - \langle \text{non-zero-digit} \rangle \langle \text{NumberSeq} \rangle \mid \langle \text{digit} \rangle \mid - \langle \text{non-zero-digit} \rangle$

$\langle \text{NumberSeq} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{digit} \rangle \langle \text{NumberSeq} \rangle$

$\langle \text{digit} \rangle ::= 0 \mid 1 \mid \dots \mid 9$

$\langle \text{non-zero-digit} \rangle ::= 1 \mid 2 \mid \dots \mid 9$

Miclea George ✓

$\langle \text{Integer} \rangle ::= 0 \mid \langle \text{No} \rangle \mid \langle \text{Sign} \rangle \langle \text{No} \rangle$

$\langle \text{No} \rangle ::= \langle \text{NonZeroDigit} \rangle \mid \langle \text{NonZeroDigit} \rangle \langle \text{DigitSeq} \rangle$

$\langle \text{DigitSeq} \rangle ::= \langle \text{Digit} \rangle \mid \langle \text{Digit} \rangle \langle \text{DigitSeq} \rangle$

$\langle \text{NonZeroDigit} \rangle ::= 1 \mid 2 \mid \dots \mid 9$

$\langle \text{Digit} \rangle ::= 0 \mid \langle \text{NonZeroDigit} \rangle$

$\langle \text{Sign} \rangle ::= + \mid -$

EBNF (Extended BNF)

Wirth's dialect

Nonterminals loose $\langle \rangle \Rightarrow$ written without delimiters

Terminals are written between `''`

$::=$ becomes $=$

$\{ \}$ repetition 0 or more times

$[]$ optionality

$()$ grouping

$(* *)$ comments

rules end with $.$

Ex.2 reloaded in EBNF

Onita Andrei

$\text{Integer} = ["+" \mid "-"] \text{Non0digit} \{ \text{Number} \} \mid \{ \text{Number} \}$

Mihalcea Leonard ✓

$\text{Integer} = '0' \mid ['+'|-'] ('1'|'2'|...|'9') \{ '0'|'1'|'2'|...|'9' \}$

Moldovanu Dragos ✓

Number= |

0" | ["-" | "+"]non-zero-digit{digit}

Non-zero-digit="1"|"2"|...|"9"

Digit="0"|"1"|...|"9"

Petcu Dragos✓

Integer = "0" | ["+"|" -"] nonZeroDigit {digit}

Pascotescu Iuliana

Number = 0 | [-] nonZeroDigit {digit}

Correct EBNF rules for identifiers and constants (discussed in Seminar 2)

Identifiers:

Moldovanu Dragos

identifier=letter{alphanumeric}

alphanumeric="0"|"1"|...|"9"|"A"|...|"z"

Onita Andrei

identifier ::= letter {seq}

seq ::= letter | digit

letter ::= "A" | "B" | . . . | "Z"|"a" |...|"z"

digit ::= "0" | "1" |...| "9"

Costants (char & string):

Moldovanu Dragos

character= " ' " placeholder " ' "

placeholder="0" |...|"9"|"A" |...|"z"|"!" |...|"#"

Neta Razvan

String = ""{letter|digit|specialchar}""