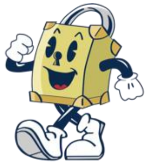
Compiler

– 3-A. Syntax Analysis (BNF) –

JIEUNG KIM

jieungkim@yonsei.ac.kr



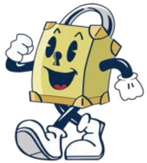
Outlines

- BNF

Backus–Naur Form

BNF notation

- BNF notation
  - Grammars for programming languages are often written in BNF (Backus-Naur Form)
  - Variables are words in <…>
    - Example: <statement>
  - Terminals are often multicharacter strings indicated by boldface or underline
    - Example: **while** or WHILE



BNF notation

- BNF notation
  - Symbol ::= is often used for ->
  - Symbol | is used for "or"
    - A shorthand for a list of productions with the same left side
  - Example: S -> 0S1 | 01 is shorthand for S -> 0S1 and S -> 01

BNF notation

- Kleene closure
  - Symbol ··· is used for "one or more"
  - Example:
    - <digit> ::= 0|1|2|3|4|5|6|7|8|9
    - <unsigned integer> ::= <digit>···
  - Translation: Replace α··· with a new variable A and productions A -> Aα | α
    - Note: that's not exactly the * of RE's
  - Grammar for unsigned integers can be replaced by
    - U -> UD | D
    - D -> 0|1|2|3|4|5|6|7|8|9

BNF notation

- Optional elements
  - Surround one or more symbols by [⋯] to make them optional
  - Example: <statement> ::= **if** <condition> **then** <statement> [; **else** <statement>]
  - Translation: replace [α] by a new variable A with productions A -> α | ε
  - Grammar for if-then-else can be replaced by:
    - S -> iCtSA
    - A -> ;eS | ε

BNF notation

- Example

```
<s>::=<n> <v>
<n>::=Marty | Victoria | Stuart | Jessica
<v>::=cried | slept | belched
```

  - Some sentences that could be generated from this grammar:

```
Marty slept
Jessica belched
Stuart cried
```

BNF notation

- Example

```
<s>::=<np> <v>
<np>::=<pn> | <dp> <n>
<pn>::=Marty | Victoria | Stuart | Jessica
<dp>::=a | the
<n>::=ball | hamster | carrot | computer
<v>::=cried | slept | belched
```

- Some sentences that could be generated from this grammar:

```
the carrot cried
Jessica belched
a computer slept
```

BNF notation

- Recursion
  - Grammar rules can be defined *recursively*, so that the expansion of a symbol can contain that same symbol
  - There must also be expressions that expand the symbol into something non-recursive, so that the recursion eventually ends

```
<s>::=<np> <v>
<np>::=<pn> | <dp> <adjp> <n>
<pn>::=Marty | Victoria | Stuart | Jessica
<dp>::=a | the
<adjp>::=<adj> <adjp> | <adj>
<adj>::=silly | invisible | loud | romantic
<n>::=ball | hamster | carrot | computer
<v>::=cried | slept | belched
```

Questions?