CP 465 Database II

BNF/EBNF notation

Ilias S. Kotsireas

ikotsire@wlu.ca

BNF notation

- BNF is an acronym for "Backus Naur Form"
- invented by John Backus and Peter Naur
- formal notation to describe the syntax of a given (programming) language
- 3 BNF meta-symbols:

```
::=     definition
|       disjunction
< >     non-terminal symbol
```

PASCAL BNF snippet:

```
<program> ::=  program
                   <declarationSeq>
               begin
                   <statementSequence>
               end ;

<declarationSeq> ::=    <constantDeclaration>; |
                        <typeDeclaration>; |
                        <variableDeclaration>;

<constantDeclaration> ::=  |
   const  <identifier> = <constant>
   <constantDeclaration> ;  <identifier> = <constant>
```

FAPP, BNF notation is (highly) recursive

EBNF notation

- Extend BNF with some additional meta-symbols
- similar notation with regular expressions
  - ▷ ? optional symbol (it can appear zero or one times)
  - ▷ * repeated any number of times (it can appear 0 or more times)
  - ▷ + repeated at least once (it can appear 1 or more times)
- optional items are enclosed within [ ]
- repetitive items (zero or more times) are enclosed in meta symbols { }
- terminals of only one character are surrounded by double quotes ” to distinguish them from meta-symbols
- terminal symbols (such as keywords) appear in bold

EBNF examples:

```
<identifier> ::= <letter> { <letter> | <digit> }

<identifier> ::= <letter> |
                 <identifier> [ <letter> | <digit> ]

ifStatement ::= IF booleanExpression THEN
                    statementSequence
                [ ELSE
                    statementSequence ]
                END IF ";"
```