HW_03_BNF_AND_EBNF

Actividad # 3

Lenguajes de programación

A01703613 - Daniel Cu Sánchez

BNF Switch statement in C

```
# All cases
<switch> ::= switch ( <variable> ) { <general-cases> }

<general-cases>
  ::= <cases> | <case> | <default> # default case
<cases>
  ::= <case> | <case> # One case per case
<cases>

<case>
  ::= <header>
  <code> break; # End of each case
  <header>
    ::= <head>|<head>
  <header>

<head>
  ::= case <value>:
<default> ::= default: <code> # Default case
```

EBNF Switch statement in C

```
SWITCH ::= switch '('variable')' '{'{case} [default]'}'
CASE ::= {head} CODE break;
HEAD ::= case VALUE:
DEFAULT ::= default: CODE
```

BNF foor loop in Python

```
<For>::= for <expression> in range (start, stop, step)
```

EBNF foor loop in Python

```
FOR ::= for in range
RANGE ::= '('variable','variable', 'variable')'
```

BNF case or cond statement in Racket

```
<cond>
  ::= (cond <cond-clause> | <cond-clause> <cond-clause> )

<cond-clause>
  ::= [ ( <test-expr> ) then-body ] #if true
    | [ ( <test-expr> ) ⇒ proc-expr ]
    | [ ( <test-expr> ) ]
    | [ else then-body ] # case not true
```

EBNF case or cond statement in Racket

```
COND ::= '('cond {COND-CLAUSE} ')'
COND-CLAUSE ::= '[' '(' TEST-EXPR ')'
  ( [THEN-BODY] ['⇒' PROC-EXPR ] [else THEN-BODY ] ) ']'
```