



# Théorie des langages (formels)

le langage WHILE

Olivier Ridoux

# Définitions formelles

- Unités lexicales
  - expressions régulières
- Syntaxe
  - grammaires

# Syntaxe des programmes

**Program** → **Function** <CR> **Program**

**Function** → **'function'** **LC** **Symbol** **':'** **LC** **Definition**

**Definition** → **'read'** **LC** **Input**

**'%'** **Commands**

**'%'** **'write'** **LC** **Output**

**Input** → **Variable** **','** **Input**

**Input** → **Variable**

**Output** → **Variable** **','** **Output** | **Variable**

# Syntaxe des commandes simples

**Commands  $\rightarrow$  Command ';' Commands**

**Commands  $\rightarrow$  Command**

**Command  $\rightarrow$  'nop' | Vars ':=' Exprs**

**Vars  $\rightarrow$  Variable ',' Vars | Variable**

**Exprs  $\rightarrow$  Expr ',' Exprs | Expr**

# • Syntaxe des commandes complexes

• Command → 'while' LC Expr LC 'do' LC Commands LC 'od'

• Command → 'for' LC Expr LC 'do' LC Commands LC 'od'

• Command → 'if' LC Expr LC 'then' LC Commands LC  
'else' LC Commands LC 'fi'

- Confort

• Command → 'foreach' LC Expr LC 'in' LC Expr LC  
'do' LC Commands LC 'od'

• Command → 'if' LC Expr LC 'then' LC Commands LC 'fi'

# Syntaxe des expressions simples

**Expr**  $\rightarrow$  **ExprSimple**

**ExprSimple**  $\rightarrow$  'nil' | **Variable** | **Symbol**

**ExprSimple**  $\rightarrow$  '(' 'cons' Lexpr ')' | '(' 'list' Lexpr ')'

**ExprSimple**  $\rightarrow$  '(' 'hd' **LC** Expr ')' | '(' 'tl' **LC** Expr ')'

**ExprSimple**  $\rightarrow$  '(' **Symbol** LExpr ')'

**LExpr**  $\rightarrow$  **LC** Expr LExpr |

# Syntaxe des expressions logiques

**Expr  $\rightarrow$  ExprAnd**

**ExprAnd  $\rightarrow$  ExprOr LC and LC ExprAnd | ExprOr**

**ExprOr  $\rightarrow$  ExprNot LC or LC ExprOr | ExprNot**

**ExprNot  $\rightarrow$  not LC ExprEq | ExprEq**

**ExprEq  $\rightarrow$  ExprSimple '=' ExprSimple | '(' Expr ')'**

# Unités lexicales

- Variables et symboles (base + confort)

- variables : [A-Z]

[0-9, a-z, A-Z]\*

(([-, +, ., /, \_, &]|"->") [0-9, a-z, A-Z]+)\*

[?, !]

- symboles : [a-z]

[0-9, a-z, A-Z]\*

(([-, +, ., /, \_, &]|"->") [0-9, a-z, A-Z]+)\*

[?, !]

...sauf mots-clés

le langage while



# Unités lexicales

- Ponctuations : ( , | ; | : | '(' | ')' )

- Caractères de mise-en-page (*layout chars*)

**LC = <SP> | <CR> | <TAB> | <LF>**

...on peut insérer des caractères de mise-en-page partout entre deux non-terminaux