

## Capítol 3 Anàlisi Sintàctica

3.1 Definir les gramàtiques per als següents llenguatges:

(a)  $l_1 = \{ a^n b^n \mid n > 0 \}$

(b)  $l_2 = \{ a^n b^m \mid m > n \geq 0 \}$

(c)  $l_3 = \{ a^n b a^m \mid m, n > 0 \}$

(d)  $l_4 = \{ a^n b a^n \mid n > 0 \}$

(e)  $l_5 = \{ a^n b^n c^n \mid n > 0 \}$

3.2 Expressar la gramàtica en BNF dels següents llenguatges:

(a) Expressions amb operands números i variables. Les operacions possibles son: +, -, \*, /, (), -(unari).

(b) Afegir a les expressions anteriors la possibilitat de fer crides a funcions amb paràmetres:

(c) Afegir a les expressions anteriors la possibilitat de arrays i el factorial (n!): Si una funció NO pot retornar un array i si SI pot retornar un array:

## Capítol 3 Anàlisi Sintàctica

3.3 Estudiar l'arbre sintàctic generat per les gramàtiques anteriors per a les següents expressions. Si la gramàtica es ambigua (més d'un arbre possible) o si l'arbre sintàctic generat no correspon amb l'estructura de l'expressió, corregir la gramàtica.

a)  $1+2*(a-b)-5$

b)  $-(a+b)*c$

c)  $\text{fib}(n-1)+\text{fib}(n-2)$

d)  $\text{array}[1,3*a][2]$

e)  $5+3!$

3.4 Considerar la següent sintaxi

$$\langle I \rangle ::= \langle A \rangle$$

$$\langle A \rangle ::= \langle B \rangle | \text{If } \langle A \rangle \text{ Then } \langle A \rangle \text{ Else } \langle A \rangle$$

$$\langle B \rangle ::= \langle C \rangle | \langle B \rangle '+' \langle C \rangle | '+' \langle C \rangle$$

$$\langle C \rangle ::= \langle D \rangle | \langle C \rangle '*' \langle D \rangle | '*' \langle D \rangle$$

$$\langle D \rangle ::= 'x' | '(' \langle A \rangle ')' | '-' \langle D \rangle$$

Quins símbols son terminals i quins son no terminals?

## Capítol 3 Anàlisi Sintàctica

Construir els arbres sintàctics per a les frases següents:

a)  $x+x$

b)  $(x+x) * (+-x)$

c)  $(x*-+x)$

d) if  $x+x$  then  $x*x$  else  $-x$

e) if  $x$  then if  $-x$  then  $x$  else  $x+x$  else  $x*x$

f) if  $-x$  then  $x$  else if  $x+x$  else  $x$

3.5 Calcular els conjunts de símbols primers i següents dels símbols terminals i no terminals de la gramàtica de l'exercici anterior.

3.6 Calcular els conjunts de símbols primers i següents dels símbols no terminals, i verificar que les gramàtiques següents son LL(1).

a)  $\langle I \rangle ::= \langle A \rangle$

$\langle A \rangle ::= \langle B \rangle '+' \langle A \rangle \mid \langle D \rangle \langle C \rangle$

$\langle B \rangle ::= \langle D \rangle \mid \langle D \rangle '*' \langle B \rangle$

$\langle D \rangle ::= 'x' \mid '( C )'$

$\langle C \rangle ::= '+'x \mid '-'x$

b)  $\langle \text{exp} \rangle ::= \langle \text{ter} \rangle \{ ('+' | '-') \langle \text{exp} \rangle \}$

$\langle \text{ter} \rangle ::= \langle \text{fac} \rangle [ ('*' | '/') \langle \text{ter} \rangle ]$

$\langle \text{fac} \rangle ::= '( \langle \text{exp} \rangle )' \mid \text{Num}$

## Capítol 3 Anàlisi Sintàctica

3.7 Crear les gramàtiques que permetin aplicar la tècnica d'anàlisi descendent d'un símbol per davant sense tornada enrera, i que expressi, amb l'arbre sintàctic generat l'estructura dels següents llenguatges:

a) Expressions del llenguatge C amb números i variables com operands i les següents operacions: +, -, \*, /, ++(prefix), --(prefix), ++(postfix), --(postfix), (), =

b) Instruccions de Pascal, considerant que està definit el símbol no terminal <Expressió>. Aquest llenguatge es centra en tractar les expressions de control de flux de Pascal (IF, THEN, ELSE, FOR, WHILE, REPEAT, UNTIL, CASE).

## Capítol 3 Anàlisi Sintàctica

### 3.1 Definir les gramàtiques per als següents llenguatges:

(a)  $l_1 = \{ a^n b^n \mid n > 0 \}$  Lliure de context:      (e)  $l_5 = \{ a^n b^n c^n \mid n > 0 \}$  Sensitiu al context:

$$S \rightarrow aSb$$

$$S \rightarrow ab$$

$$S \rightarrow aSBC$$

(b)  $l_2 = \{ a^n b^m \mid m > n \geq 0 \}$  Lliure de context:

$$S \rightarrow abC$$

$$S \rightarrow Sb$$

$$bB \rightarrow bb$$

$$S \rightarrow b$$

$$bC \rightarrow bc$$

$$S \rightarrow aSb$$

$$CB \rightarrow BC$$

(c)  $l_3 = \{ a^n b a^m \mid m, n > 0 \}$  Regular:

$$cC \rightarrow cc$$

$$S \rightarrow aS$$

$$S \rightarrow aB$$

$$B \rightarrow bC$$

$$C \rightarrow aC$$

$$C \rightarrow a$$

(d)  $l_4 = \{ a^n b a^n \mid n > 0 \}$  Lliure de Context:

$$S \rightarrow aCa$$

$$C \rightarrow aCa$$

$$C \rightarrow b$$

## Capítol 3 Anàlisi sintàctica

3.2 Expressar la gramàtica en BNF dels següents llenguatges:

(a) Expressions amb operands números i variables. Les operacions possibles son:  
+, -, \*, /, (), -(unari).

$$\langle \text{exp} \rangle ::= \langle \text{ter} \rangle \{ ('+' | '-' ) \langle \text{ter} \rangle \}$$
$$\langle \text{ter} \rangle ::= \langle \text{fac} \rangle \{ ('*' | '/' ) \langle \text{fac} \rangle \}$$
$$\langle \text{fac} \rangle ::= \text{Num} | \text{Ident} | '-' \langle \text{fac} \rangle | '(' \langle \text{exp} \rangle ')'$$

b) Afegir a les expressions anteriors la possibilitat de fer crides a funcions amb paràmetres:

$$\langle \text{exp} \rangle ::= \langle \text{ter} \rangle \{ ('+' | '-' ) \langle \text{ter} \rangle \}$$
$$\langle \text{ter} \rangle ::= \langle \text{fac} \rangle \{ ('*' | '/' ) \langle \text{fac} \rangle \}$$
$$\langle \text{fac} \rangle ::= '-' (\text{Num} | \text{Ident} [ '(' [ \langle \text{exp} \rangle \{ ',' \langle \text{exp} \rangle \} ] ')' ] | '(' \langle \text{exp} \rangle ')')$$

## Problema 3.2(cont.)

3.2(c) Afegir a les expressions anteriors la possibilitat de arrays i el factorial (n!):

$\langle \text{exp} \rangle ::= \langle \text{ter} \rangle \{ (+|-) \langle \text{ter} \rangle \}$

$\langle \text{ter} \rangle ::= \langle \text{fac1} \rangle \{ (*|/) \langle \text{fac1} \rangle \}$

$\langle \text{fac1} \rangle ::= '-' \langle \text{fac1} \rangle | \langle \text{fac2} \rangle$

Si una funció NO pot retornar un array:

$\langle \text{fac2} \rangle ::= (\text{Num} | \text{Ident} [ ( ' ( [ \langle \text{exp} \rangle \{ , \} \langle \text{exp} \rangle ] ' ) ' ) | ( [ ' \langle \text{exp} \rangle \{ , \} \langle \text{exp} \rangle ' ] ' ) ] | ' ( \langle \text{exp} \rangle ' ) ' ) \{ ' ! ' \}$

Si una funció SI pot retornar un array:

$\langle \text{fac2} \rangle ::= (\text{Num} | \text{Ident} [ ( ' ( [ \langle \text{exp} \rangle \{ , \} \langle \text{exp} \rangle ] ' ) ' ) | ( [ ' \langle \text{exp} \rangle \{ , \} \langle \text{exp} \rangle ' ] ' ) ] | ' ( \langle \text{exp} \rangle ' ) ' ) \{ ' ! ' \}$

## Problema 3.3

3.3 Estudiar l'arbre sintàctic generat per les gramàtiques anteriors per a les següents expressions. Si la gramàtica es ambigua (més d'un arbre possible) o si l'arbre sintàctic generat no correspon amb l'estructura de l'expressió, corregir la gramàtica.

a)  $1+2*(a-b)-5$

b)  $-(a+b)*c$

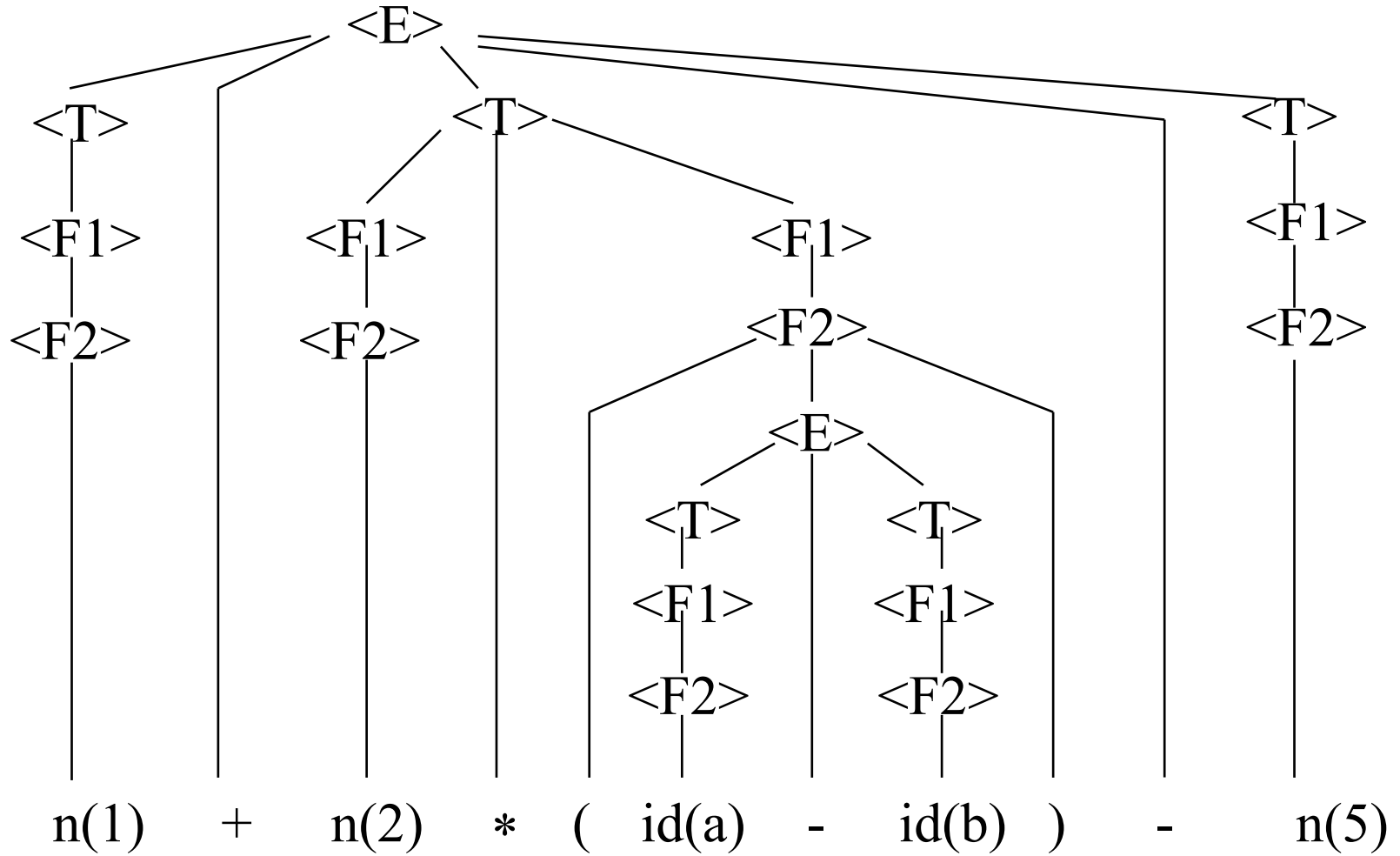
c)  $\text{fib}(n-1)+\text{fib}(n-2)$

d)  $\text{array}[1,3*a][2]$

e)  $5+3!$

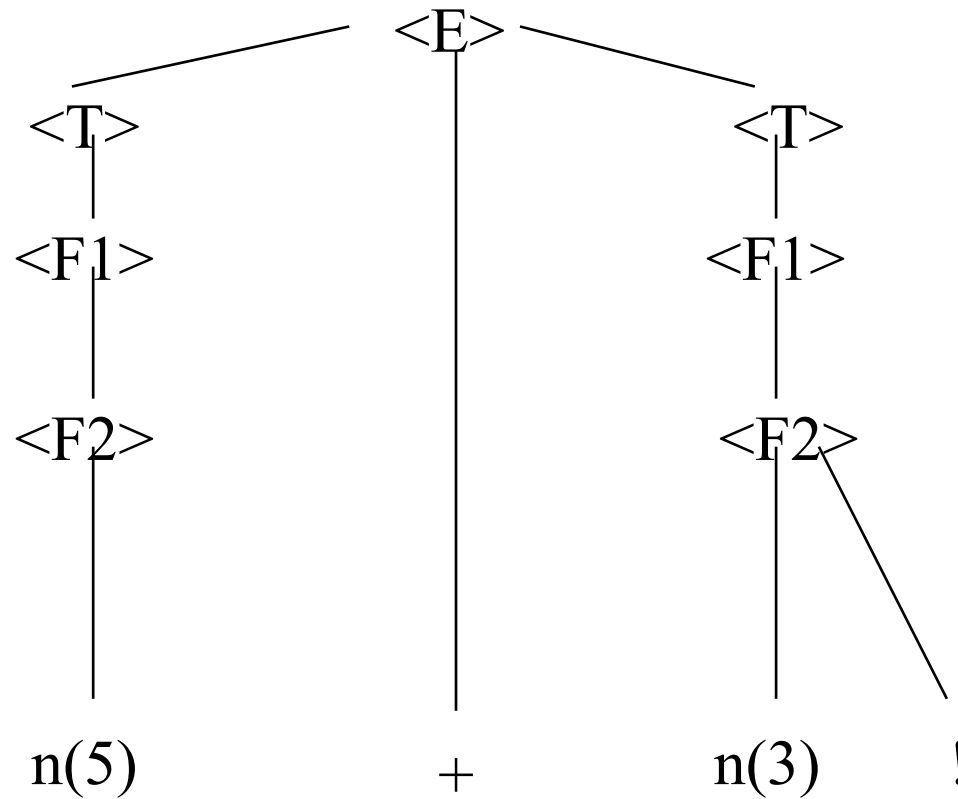
## Problema 3.3(cont)

a)  $1+2*(a-b)-5$



## Problema 3.3(cont)

e) 5+3!



## Problema 3.4

3.4 Considerar la següent sintaxi

$\langle I \rangle ::= \langle A \rangle$

$\langle A \rangle ::= \langle B \rangle \mid \text{If } \langle A \rangle \text{ Then } \langle A \rangle \text{ Else } \langle A \rangle$

$\langle B \rangle ::= \langle C \rangle \mid \langle B \rangle \text{ ' + ' } \langle C \rangle \mid \text{ ' + ' } \langle C \rangle$

$\langle C \rangle ::= \langle D \rangle \mid \langle C \rangle \text{ ' * ' } \langle D \rangle \mid \text{ ' * ' } \langle D \rangle$

$\langle D \rangle ::= \text{ ' x ' } \mid \text{ ' ( ' } \langle A \rangle \text{ ' ) ' } \mid \text{ ' - ' } \langle D \rangle$

Quins símbols son terminals i quins son no terminals?

No\_Terminals = { $\langle I \rangle, \langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle$ }

Terminals = {if, then, else, +, \*, '(', ')', x}

Construir els arbres sintàctics per a les frases següents:

a)  $x+x$

b)  $(x+x) * (+-x)$

c)  $(x*-+x)$

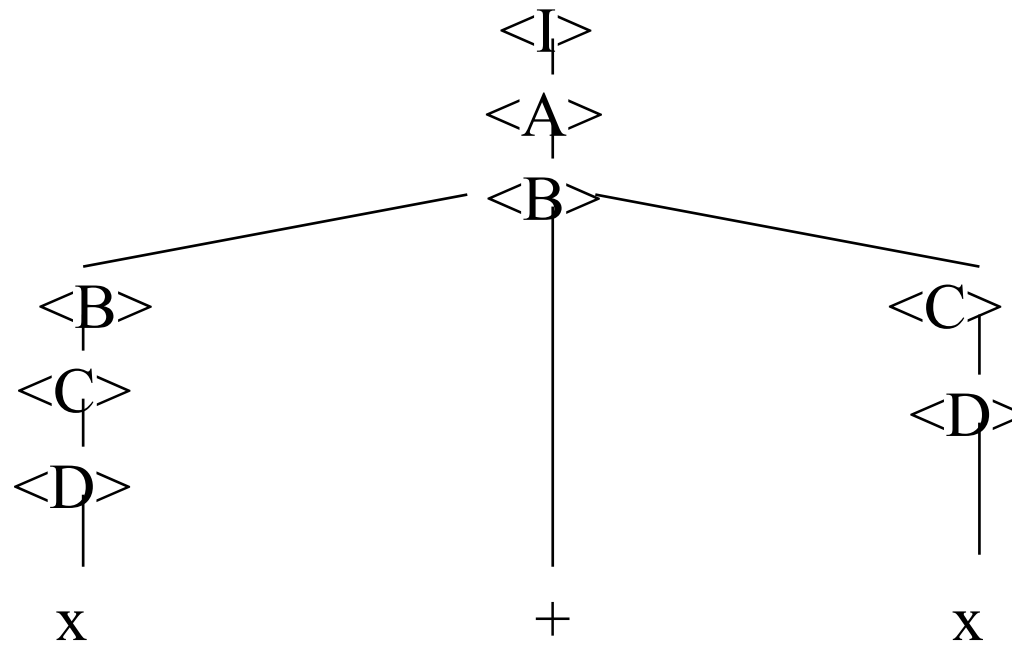
d) if  $x+x$  then  $x*x$  else  $-x$

e) if  $x$  then if  $-x$  then  $x$  else  $x+x$  else  $x*x$

f) if  $-x$  then  $x$  else if  $x+x$  else  $x$

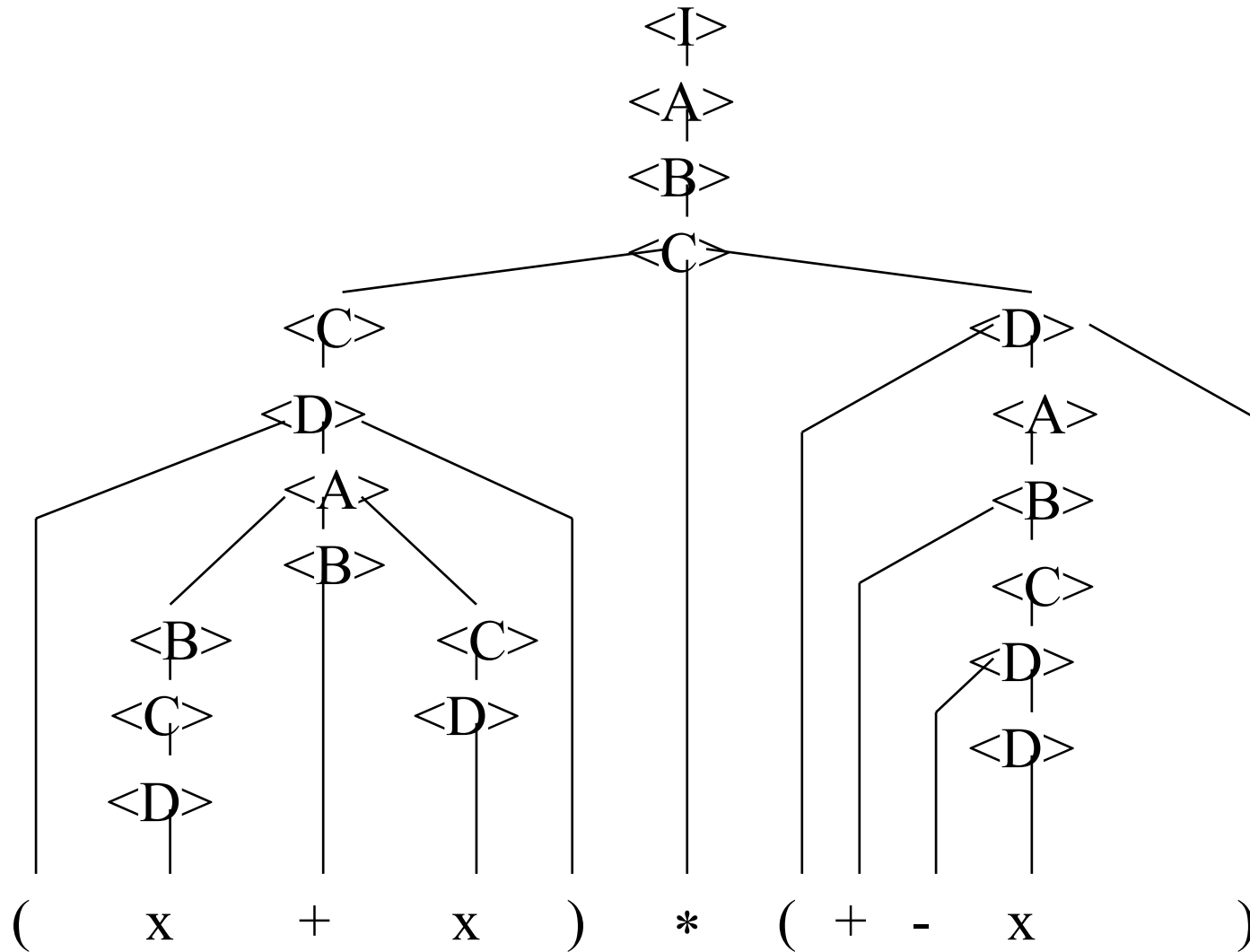
## Problema 3.4(cont)

a)  $x+x$



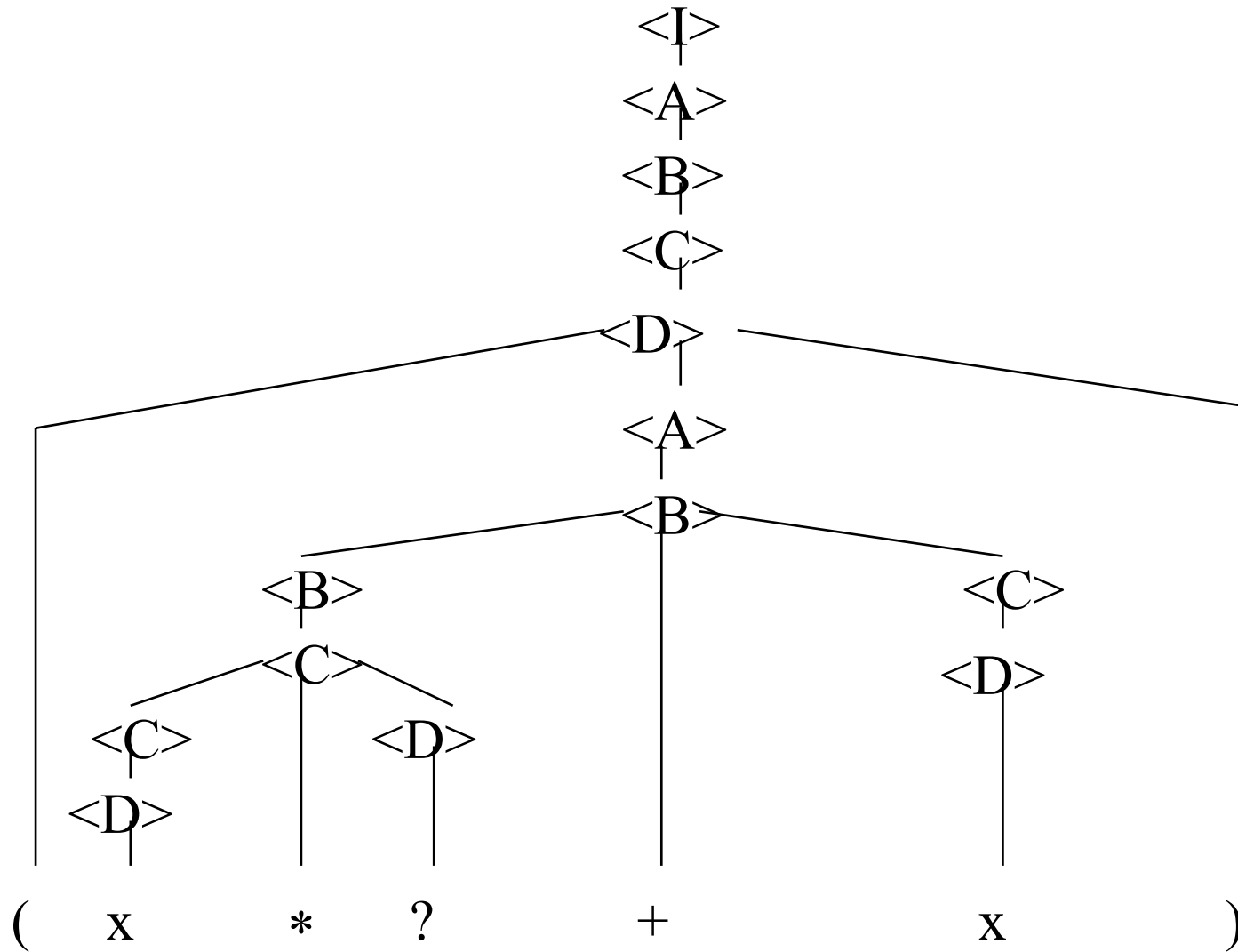
## Problema 3.4(cont)

b)  $(x+x) * (+-x)$



## Problema 3.4(cont)

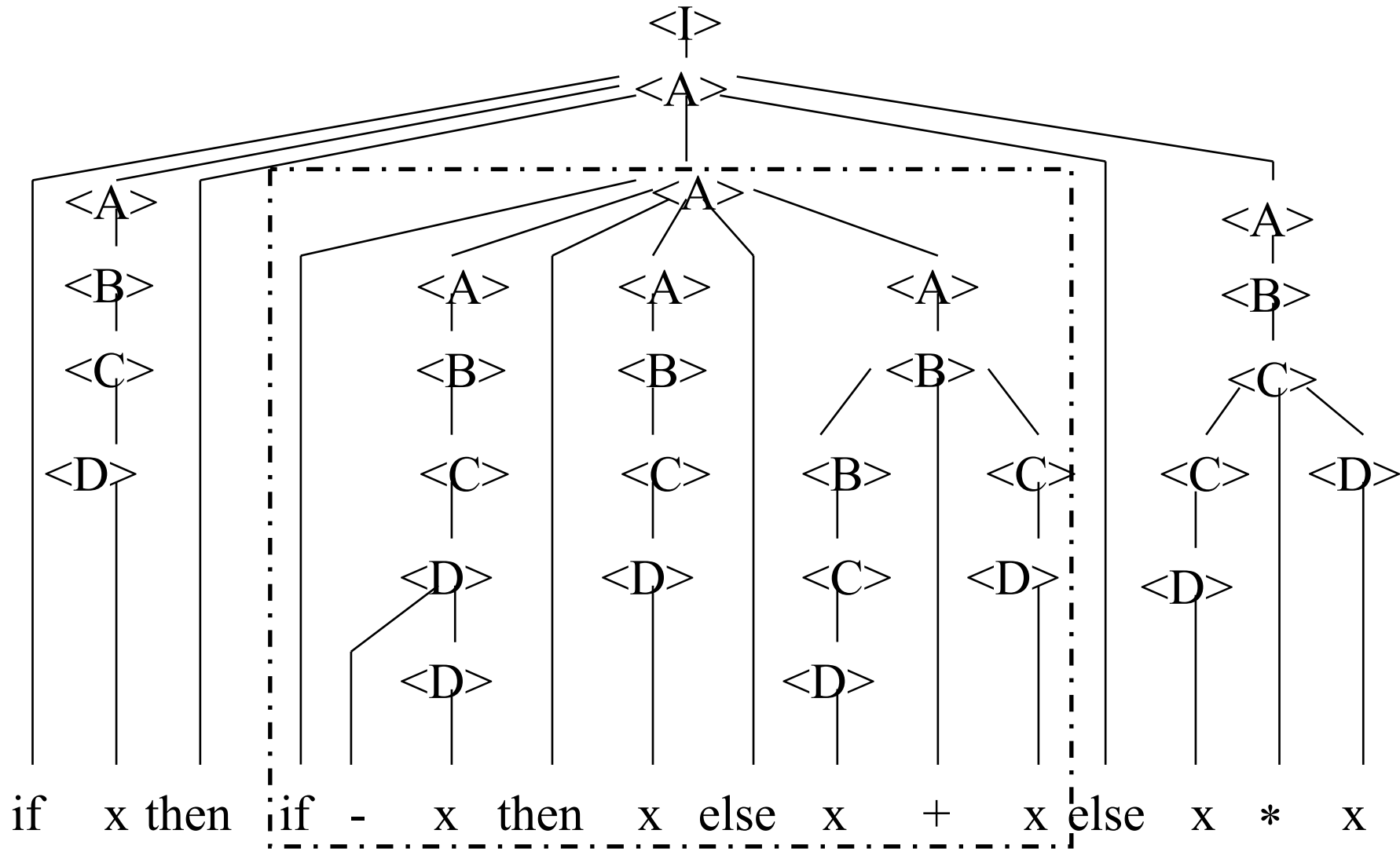
c)  $(x^* - +x)$





## Problema 3.4(cont)

e) if x then if -x then x else x+x else x\*x



## Problema 3.5

4.5 Calcular els conjunts de símbols primers i següents dels símbols terminals i no terminals de la gramàtica de l'exercici anterior.

$$\text{prim}(\langle I \rangle) = \text{prim}(\langle A \rangle) = \{\text{if}, +, *, \text{x}, '(', -\}$$

$$\text{prim}(\langle A \rangle) = \{\text{if}\} \cup \text{prim}(\langle B \rangle) = \{\text{if}, +, *, \text{x}, '(', -\}$$

$$\text{prim}(\langle B \rangle) = \{+\} \cup \text{prim}(\langle C \rangle) = \{+, *, \text{x}, '(', -\}$$

$$\text{prim}(\langle C \rangle) = \{*\} \cup \text{prim}(\langle D \rangle) = \{*, \text{x}, '(', -\}$$

$$\text{prim}(\langle D \rangle) = \{\text{x}, '(', -\}$$

$$\text{seg}(\langle I \rangle) = \{\lambda\}$$

$$\begin{aligned} \text{seg}(\langle A \rangle) &= \{\text{then}, \text{else}, ')'\} \cup \text{seg}(\langle I \rangle) = \\ &= \{\text{then}, \text{else}, ')', \lambda\} \end{aligned}$$

$$\text{seg}(\langle B \rangle) = \text{seg}(\langle A \rangle) \cup \{+\} = \{\text{then}, \text{else}, ')', \lambda, +\}$$

$$\begin{aligned} \text{seg}(\langle C \rangle) &= \text{seg}(\langle B \rangle) \cup \{*\} = \\ &= \{\text{then}, \text{else}, ')', \lambda, +, *\} \end{aligned}$$

$$\text{seg}(\langle D \rangle) = \text{seg}(\langle C \rangle) = \{\text{then}, \text{else}, ')', +, \lambda, *\}$$

## Problema 3.6

3.6 Calcular els conjunts de símbols primers i següents dels símbols no terminals, i verificar que les gramàtiques següents son LL(1).

a)  $\langle I \rangle ::= \langle A \rangle$

$\langle A \rangle ::= \langle B \rangle \text{ '+' } \langle A \rangle \mid \langle D \rangle \langle C \rangle$

$\langle B \rangle ::= \langle D \rangle \mid \langle D \rangle \text{ '*' } \langle B \rangle$

$\langle D \rangle ::= \text{'x'} \mid \text{'(' } \langle C \rangle \text{'}'$

$\langle C \rangle ::= \text{'+' } \text{'x'} \mid \text{'-' } \text{'x'}$

Càlcul dels primers:

$\text{prim}(\langle I \rangle) = \text{prim}(\langle A \rangle) = \{\text{x}, \text{'('}\}$

$\text{prim}(\langle A \rangle) = \text{prim}(\langle B \rangle) \cup \text{prim}(\langle D \rangle) = \{\text{x}, \text{'('}\}$

$\text{prim}(\langle B \rangle) = \text{prim}(\langle D \rangle) = \{\text{x}, \text{'('}\}$

$\text{prim}(\langle D \rangle) = \{\text{x}, \text{'('}\}$

$\text{prim}(\langle C \rangle) = \{+, -\}$

## Problema 3.6(cont)

3.6(a) (cont.)

Càlcul dels següents:

$$\text{seg}(\langle I \rangle) = \{\lambda\}$$

$$\text{seg}(\langle A \rangle) = \text{seg}(\langle I \rangle) = \{\lambda\}$$

$$\text{seg}(\langle B \rangle) = \{+\}$$

$$\text{seg}(\langle D \rangle) = \{*\} \cup \text{prim}(\langle C \rangle) \cup \text{seg}(\langle B \rangle) = \{*, +, -\}$$

$$\text{seg}(\langle C \rangle) = \{ '\} \cup \text{seg}(\langle A \rangle) = \{ '\}, \lambda\}$$

Ara mirem si és LL(1):

$$1. \text{ Si } A \rightarrow \beta_1 \dots A \rightarrow \beta_n \Rightarrow \text{primers}(A \rightarrow \beta_i) \cap \text{primers}(A \rightarrow \beta_j) = \phi$$

$$\forall i, j \ i \neq j \ 1 \leq i, j \leq n$$

## Problema 3.6(a)(cont)

Comprovació de si és LL(1):

$\langle A \rangle ::= \langle B \rangle '+' \langle A \rangle \mid \langle D \rangle \langle C \rangle$ $A \rightarrow B '+' A$ $A \rightarrow D C$
--

$$\begin{aligned} \text{prim}(A \rightarrow B '+' A) \cap \text{prim}(A \rightarrow D C) &= \\ &= \text{prim}(B) \cap \text{prim}(D) = \text{prim}(D) = \\ &= \{x, '('\} \neq \emptyset \end{aligned}$$

$\langle B \rangle ::= \langle D \rangle \mid \langle D \rangle '*' \langle B \rangle$ $B \rightarrow D$ $B \rightarrow D '*' B$
--

$$\begin{aligned} \text{prim}(B \rightarrow D '*' B) \cap \text{prim}(B \rightarrow D) &= \\ &= \{x, '('\} \neq \emptyset \end{aligned}$$

$\langle D \rangle ::= 'x' \mid '(' C ')'$ $D \rightarrow x$ $D \rightarrow '(' C ')'$
--

$$\begin{aligned} \text{prim}(D \rightarrow x) \cap \text{prim}(D \rightarrow '(' C ')') &= \\ &= \{x\} \cap \{ '(' \} = \emptyset \end{aligned}$$

$\langle C \rangle ::= '+' x' \mid '-' x'$
--

$$\begin{aligned} \text{prim}('+' x') \cap \text{prim}('-' x') &= \\ &= \{ '+' \} \cap \{ '-' \} = \emptyset \end{aligned}$$

**La gramàtica No és LL(1)**

## Problema 3.6(cont)

b)  $\langle \text{exp} \rangle ::= \langle \text{ter} \rangle \{ ('+' | '-') \langle \text{exp} \rangle \}$   
 $\langle \text{ter} \rangle ::= \langle \text{fac} \rangle [ ('*' | '/') \langle \text{ter} \rangle ]$   
 $\langle \text{fac} \rangle ::= '(' \langle \text{exp} \rangle ')'$  | Num

Càlcul dels primers:

$\text{prim}(\langle \text{exp} \rangle) = \text{prim}(\langle \text{ter} \rangle) = \{ '(', \text{Num} \}$

$\text{prim}(\langle \text{ter} \rangle) = \text{prim}(\langle \text{fac} \rangle) = \{ '(', \text{Num} \}$

$\text{prim}(\langle \text{fac} \rangle) = \{ '(', \text{Num} \}$

Càlcul dels següents:

$\text{seg}(\langle \text{exp} \rangle) = \{ '+', '- ', ')', \lambda \}$

$\text{seg}(\langle \text{ter} \rangle) = \text{prim}(\{ ('+' | '-') \langle \text{exp} \rangle \}) \cup \text{seg}(\langle \text{exp} \rangle) = \{ '+', '- ', ')', \lambda \}$

$\text{seg}(\langle \text{fac} \rangle) = \text{prim}([ ('*' | '/') \langle \text{ter} \rangle ]) \cup \text{seg}(\langle \text{ter} \rangle) = \{ '* ', '/ ', '+', '- ', ')', \lambda \}$

## Problema 3.6(b)(cont)

És LL(1)?:

$\langle \text{exp} \rangle ::= \langle \text{ter} \rangle \{ ('+' | '-' ) \langle \text{exp} \rangle \}$   
 $E \rightarrow T E1$   
 $E1 \rightarrow E2 E E1$   
 $E1 \rightarrow \lambda$   
 $E2 \rightarrow '+'$   
 $E2 \rightarrow '-'$

$\langle \text{ter} \rangle ::= \langle \text{fac} \rangle [ ('*' | '/') \langle \text{ter} \rangle ]$   
 $T \rightarrow F T1$   
 $T1 \rightarrow '*' T$   
 $T1 \rightarrow '/' T$   
 $T1 \rightarrow \lambda$

$\langle \text{fac} \rangle ::= (' \langle \text{exp} \rangle ' | \text{Num})$   
 $F \rightarrow '(' E '('$   
 $F \rightarrow \text{num}$

És LL(1)?:

$\text{prim}(E1 \rightarrow E2 E E1) \cap \text{prim}(E1 \rightarrow \lambda) = \{+, -\}$

$\cap \text{seg}(E1) = \{+, -\} \neq \phi$

$\text{Seg}(E1) = \text{Seg}(E) = \{\lambda\} \cup \text{Prim}(E1) = \{+, -, \lambda\}$

$\text{prim}(E2 \rightarrow '+' ) \cap \text{prim}(E2 \rightarrow '-' ) = \phi$

$\langle \text{ter} \rangle ::= \langle \text{fac} \rangle [ ('*' | '/') \langle \text{ter} \rangle ]$

$\text{Prim}(T1 \rightarrow '*' T) \cap \text{prim}(T1 \rightarrow '/' T) = \phi$

$\text{Prim}(T1 \rightarrow '*' T) \cap \text{prim}(T1 \rightarrow \lambda) = \phi$

$\text{Prim}(T1 \rightarrow \lambda) \cap \text{prim}(T1 \rightarrow '/' T) = \phi$

$\text{Prim}(F \rightarrow '(' E '(') \cap \text{prim}(F \rightarrow \text{num}) = \phi$

**La gramàtica No és LL(1)**

## Problema 3.7

3.7 Crear les gramàtiques que permetin aplicar la tècnica d'anàlisi descendent d'un símbol per davant sense tornada enrera, i que expressi, amb l'arbre sintàctic generat l'estructura dels següents llenguatges:

a) Expressions del llenguatge C amb números i variables com operands i les següents operacions: +, -, \*, /, ++(prefix), --(prefix), ++(postfix), --(postfix), (), =

$$\langle E \rangle ::= \langle T \rangle \{ ('+' | '-') \langle T \rangle \}$$
$$\langle T \rangle ::= \langle F \rangle \{ ('*' | '/') \langle F \rangle \}$$
$$\langle F \rangle ::= '-' \langle F \rangle \mid \langle F1 \rangle$$
$$\langle F1 \rangle ::= \text{Num} \mid$$
$$('++' | '--') \text{Id} ['=' \langle E \rangle] \mid$$
$$\text{Id} [('++' | '--')] ['=' \langle E \rangle] \mid$$
$$'(\langle E \rangle)'$$

## Problema 3.7(a)(cont)

3.7(a) (cont.) Opció tenint en compte que una assignació es pot fer a una expressió que ens estigui indicant una adreça de memòria.

Tingueu en compte que hem de fer comprovacions semàntiques per veure que estem assignant a una posició concreta de memòria, i que apliquem les operacions de (++) i (--) també a una posició concreta de memòria.

$$\begin{aligned} \langle A \rangle ::= & \langle E \rangle \text{ [ ' = ' } \langle A \rangle ] \\ \langle E \rangle ::= & \langle T \rangle \{ ( \text{' + ' | ' - ' } ) \langle T \rangle \} \\ \langle T \rangle ::= & \langle F \rangle \{ ( \text{' * ' | ' / ' } ) \langle F \rangle \} \\ \langle F \rangle ::= & \text{' - ' } \langle F \rangle \mid ( \text{' -- ' | ' ++ ' } ) \langle F1 \rangle \mid \\ & \langle F1 \rangle [ ( \text{' ++ ' | ' -- ' } ) ] \\ \langle F1 \rangle ::= & \text{Num} \mid \\ & \text{Id} \mid \\ & \text{' ( ' } \langle E \rangle \text{' ) ' } \end{aligned}$$

## Problema 3.7

3.7(b) Instruccions de Pascal, considerant que està definit el símbol no terminal <Expressió>. Aquest llenguatge es centra en tractar les expressions de control de flux de Pascal (IF, THEN, ELSE, FOR, WHILE, REPEAT, UNTIL, CASE).

```
<I> ::= If <E> Then <I> [Else <I>] |  
      For Id := <E> (To|Downto) <E> Do <I> |  
      While <E> Do <I> |  
      Repeat <I> { ';' <I> } Until <E> |  
      Case <E> Of  
          [      Const { ',' Const } ':' <I>  
            { ';' Const { ';' Const } ':' <I> }  
          ]  
      End |  
      Begin <I> { ';' <I> } End
```