

Grammar 4

<<Grammar4.ppt>>

【例一】寫出下列形式語法 (Formal Grammar) 之類別 (Type) 與其所定義之語言 $L(G)$ 。

$$N = \{ A, B, \Sigma \} \quad T = \{ 0, 1 \}$$

$$P = \{ \Sigma \rightarrow AB$$

$$A \rightarrow 1A$$

$$A \rightarrow 1$$

$$B \rightarrow B0$$

$$B \rightarrow 0 \}$$

【解】上述形式語法屬於type 2語法，亦即上下文自由式 (context-free) 語法；其所定義之語言 $L(G) = \{ \text{string of the form } 1^m 0^n \text{ for } m, n \geq 1 \}$ 。

【說明】令 N 表示非終端符號 (nonterminal symbol) 的集合， T 表示終端符號 (terminal symbol) 的集合， λ 表示空字串 (empty string)。Type 2 (即context-free) 語法的產生規則 (productions) 如下：

$$A \rightarrow \beta \quad \text{其中 } \beta \in (N \cup T)^* - \lambda ; A \in N$$

註： $(N \cup T)^* - \lambda$ 代表所有由集合 N 及 T 中有限個符號組成的字串之集合，但此集合不包括空字串。

注意：**type 2**語法之產生規則的左半部限制只能是單一的非終端符號。

【例二】 Let G be the grammar with the following productions

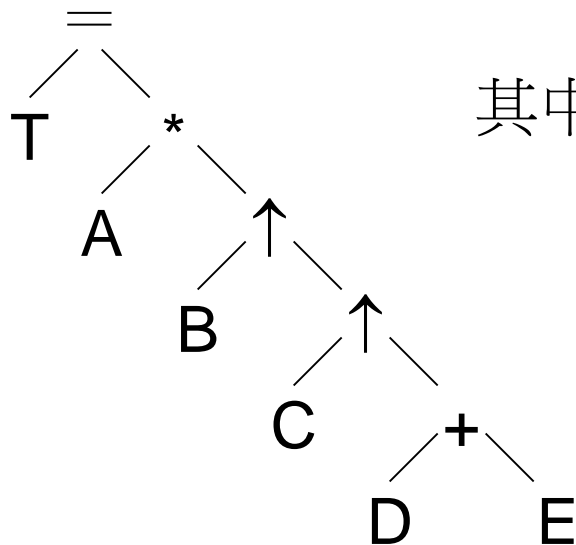
$$\begin{array}{ll}
 S \rightarrow CD & Ab \rightarrow bA \\
 C \rightarrow aCA & Ba \rightarrow aB \\
 C \rightarrow bCB & Bb \rightarrow bB \\
 AD \rightarrow aD & C \rightarrow e \\
 BD \rightarrow bD & D \rightarrow e \\
 Aa \rightarrow aA &
 \end{array}$$

Where a capital letter represents a nonterminal and a lower-case letter represents a terminal.

Does this grammar form a “ CONTEXT-FREE ” language ? [yes or no only]

【解】 No.

【說明】” **CONTEXT-FREE** “ 語言之產生規則 (productions) 的左邊部分限制只能是單一的非終端符號 (nonterminal symbol)，而上述文法違反這個限制，因此不是 “ **CONTEXT-FREE** “ 語言。



其中連續的 \uparrow 採 “由右向左” 運算。

3. 討論：

對於運算優先順序相同的連續運算符號而言(請參考BNF範例)：

(1) 從 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{term} \rangle \mid \langle \text{exp} \rangle - \langle \text{term} \rangle \mid \langle \text{term} \rangle$ 得知：

+ 及 - 採 “從左向右” 運算，亦即屬於 “向左遞迴 (left recursion)”。

(2) 從 $\langle \text{term} \rangle ::= \langle \text{term} \rangle * \langle \text{part} \rangle \mid \langle \text{term} \rangle / \langle \text{part} \rangle \mid \langle \text{part} \rangle$ 得知：

* 及 / 採 “從左向右” 運算，亦即屬於 “向左遞迴”。

(3) 從 $\langle \text{part} \rangle ::= \langle \text{symbol} \rangle \uparrow \langle \text{part} \rangle \mid \langle \text{symbol} \rangle$ 得知：

連續的 \uparrow 採 “從右向左” 運算，亦即屬於 “向右遞迴

(right recursion)”。

【例三】設有福傳 (FORTRAN) 程式如下：

```
REAL FUNCTION MEAN ( A, B, C )  
MEAN = ( 2 * A + 5 * B - 7 * C ) / 3  
RETURN  
END
```

針對此FORTRAN，寫出其BNF定義。

【解】 $\langle \text{function} \rangle ::= \langle \text{fun} \rangle \langle \text{body} \rangle \langle \text{end} \rangle$
 $\langle \text{fun} \rangle ::= \langle \text{type} \rangle \text{FUNCTION } \langle \text{idn} \rangle (\langle \text{var} \rangle)$
 $\langle \text{end} \rangle ::= \text{END}$
 $\langle \text{type} \rangle ::= \text{REAL} \mid \text{INTEGER}$
 $\langle \text{var} \rangle ::= \langle \text{idn} \rangle \mid \langle \text{idn} \rangle , \langle \text{var} \rangle$
 $\langle \text{body} \rangle ::= \langle \text{stmnt} \rangle \mid \langle \text{body} \rangle \langle \text{stmnt} \rangle$
 $\langle \text{stmnt} \rangle ::= \langle \text{assign} \rangle \mid \text{RETURN}$
 $\langle \text{assign} \rangle ::= \langle \text{idn} \rangle = \langle \text{exp} \rangle$
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{term} \rangle \mid \langle \text{exp} \rangle - \langle \text{term} \rangle \mid \langle \text{term} \rangle$
 $\langle \text{term} \rangle ::= \langle \text{term} \rangle * \langle \text{symbol} \rangle \mid \langle \text{term} \rangle / \langle \text{symbol} \rangle \mid \langle \text{symbol} \rangle$
 $\langle \text{symbol} \rangle ::= \langle \text{literal} \rangle \mid \langle \text{idn} \rangle \mid (\langle \text{exp} \rangle)$
 $\langle \text{idn} \rangle ::= \langle \text{letter} \rangle \mid \langle \text{letter} \rangle [\langle \text{letter-digit} \rangle]$
 $\langle \text{letter-digit} \rangle ::= \langle \text{letter} \rangle \mid \langle \text{digit} \rangle$

$\langle \text{letter} \rangle ::= A \mid B \mid C \mid \dots \mid Z$

$\langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\langle \text{literal} \rangle ::= [\langle \text{digit} \rangle]$

註： $[\langle \text{digit} \rangle]$ 表示包含 $\langle \text{digit} \rangle$ 的個數在1到15之間。

【例四】 假設 $T = \{ +, *, \uparrow, (,), A, B, C \}$ ， $L \subset T^*$ 為由 T 中 symbol 所組成的所有 arithmetic expression 的語言 (Language)，試寫一描述 L 之 Grammar。
($A \uparrow B$ 代表 A 之 B 次方，亦即 A^B)

【解】 以BNF描述 L 如下：

$$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{term} \rangle \mid \langle \text{term} \rangle$$

$$\langle \text{term} \rangle ::= \langle \text{term} \rangle * \langle \text{sterm} \rangle \mid \langle \text{sterm} \rangle$$

$$\langle \text{sterm} \rangle ::= \langle \text{symbol} \rangle \uparrow \langle \text{sterm} \rangle \mid \langle \text{symbol} \rangle$$

$$\langle \text{symbol} \rangle ::= A \mid B \mid C \mid (\langle \text{exp} \rangle)$$

- 【例五】
- What is an unambiguous grammar ?
 - Write an unambiguous grammar to generate valid Fortran arithmetic expression involving the operators $+$, $-$, $*$, unary $-$ (negation) , and parentheses. Use the terminal d to represent operands.
 - Draw the parse tree for the expression $d + d * d + d + d$
 - Write out the leftmost derivation for the above expression.

【解】 a. 若一文法 (grammar) 對所有的指述 (statement) 展開僅有一種導出方式時，則此類文法稱為語意明確的文法 (unambiguous grammar) 。

b. 以BNF (BACKUS – NAUR FORM) 寫一語意明確的文法，
描述產生規則如下：

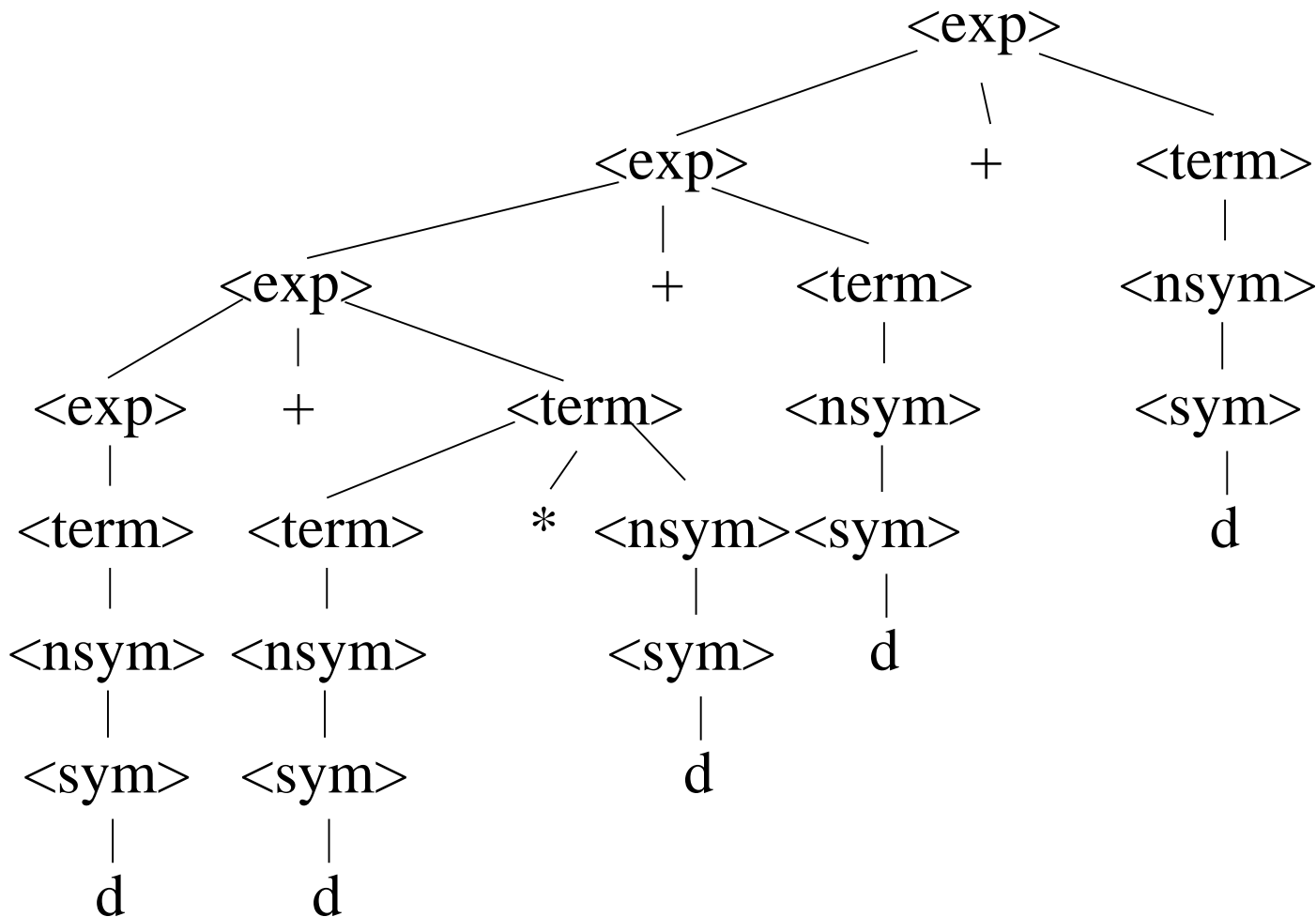
$\langle \text{sym} \rangle ::= d \mid (\langle \text{exp} \rangle) \dots\dots\dots(1)$

$\langle \text{nsym} \rangle ::= \langle \text{sym} \rangle \mid -\langle \text{sym} \rangle \dots\dots\dots(2)$

$\langle \text{term} \rangle ::= \langle \text{term} \rangle * \langle \text{nsym} \rangle \mid \langle \text{nsym} \rangle \dots\dots\dots(3)$

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle + \langle \text{term} \rangle \mid \langle \text{exp} \rangle - \langle \text{term} \rangle \mid \langle \text{term} \rangle \dots(4)$

c. 畫表示式 $d + d * d + d + d$ 對應的分析樹 (parse tree) 如下：



d. 寫最左導出 (left derivation) 表示 $d + d * d + d + d$ 的過程如下：

$$\begin{aligned}
 \langle \text{exp} \rangle &::= \langle \text{exp} \rangle + \langle \text{term} \rangle \quad (\text{利用(4)展開}) \\
 &::= \langle \text{exp} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle \quad (\text{利用(4)展開}) \\
 &::= \langle \text{exp} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle \quad (\text{利用(4)展開}) \\
 &::= \langle \text{term} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle \quad (\text{利用(4)展開}) \\
 &::= \langle \text{nsym} \rangle + \langle \text{term} \rangle * \langle \text{nsym} \rangle + \langle \text{nsym} \rangle + \langle \text{nsym} \rangle \\
 &\quad (\text{利用(3)展開}) \\
 &::= \langle \text{nsym} \rangle + \langle \text{nsym} \rangle * \langle \text{nsym} \rangle + \langle \text{nsym} \rangle + \langle \text{nsym} \rangle \\
 &\quad (\text{利用(3)展開}) \\
 &::= \langle \text{sym} \rangle + \langle \text{sym} \rangle * \langle \text{sym} \rangle + \langle \text{sym} \rangle + \langle \text{sym} \rangle \quad (\text{利用(2)展開}) \\
 &::= d + d * d + d + d \quad (\text{利用(1)展開})
 \end{aligned}$$

語法圖

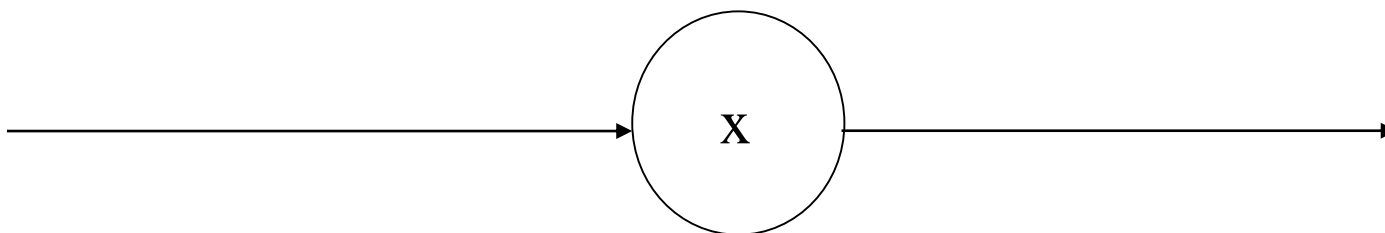
規則 G_0

設任一非終端符號 $\langle A \rangle$ ，在造句程序集中存在一組造句程序與之對應。亦即

$$\langle A \rangle ::= \xi_1 \mid \xi_2 \mid \dots \mid \xi_n$$

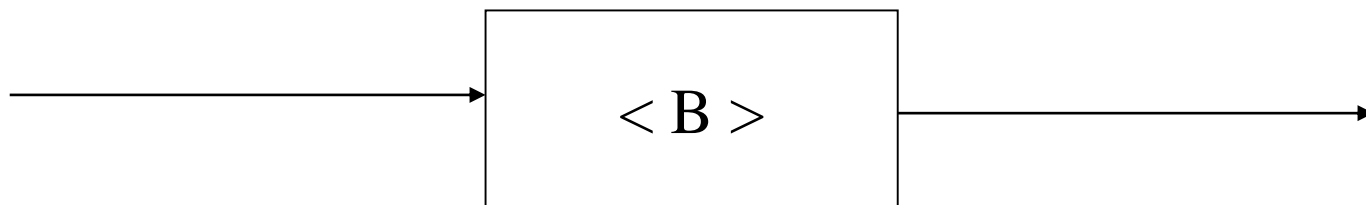
規則 G_1

對屬於符號串 ξ_i 中任一終端符號 x ，其語法圖是以一圓圈內加註該終端符號 x 來表示。即



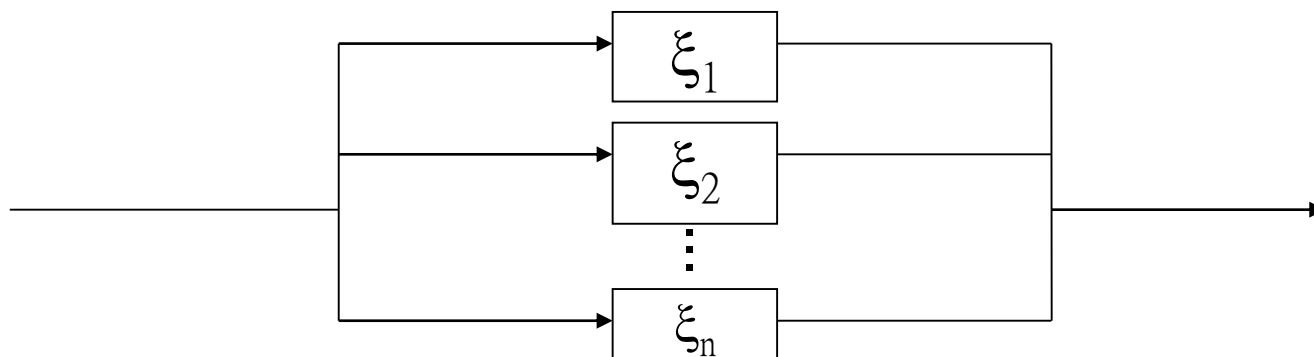
規則 G_2

對屬於符號串 ξ_i 中任一非終端符號 $\langle B \rangle$ ，其語法圖是以一方框內加註該非終端符號 $\langle B \rangle$ 來表示。即



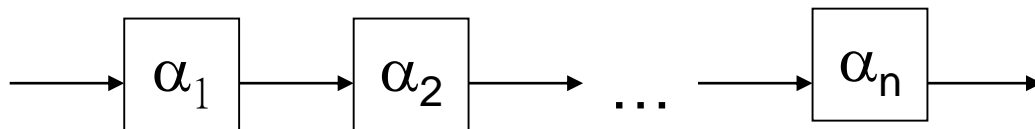
規則 G_3

若 $\langle A \rangle ::= \xi_1 \mid \xi_2 \mid \dots \mid \xi_n$ 則其相對應之語法圖為



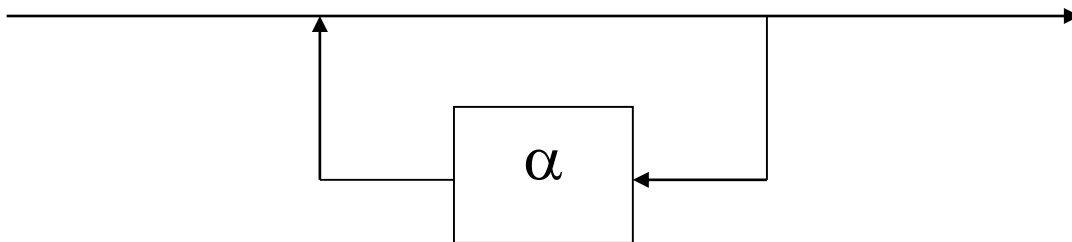
規則 G_4

若 $\xi = \alpha_1 \alpha_2 \dots \alpha_n$ 則其相對應之語法圖為



規則 G_5

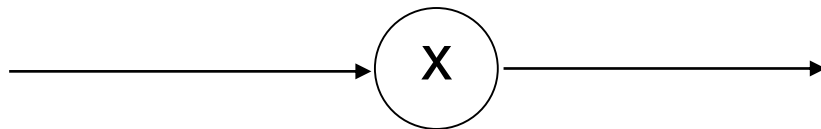
若 $\xi = \{ \alpha \}$ 則其相對應的語法圖為



語法圖轉換為程式

規則 T_1

若終端符號之語法圖為



則其轉換敘述為

IF CH = 'x' THEN Next Char ELSE ERROR ;

規則 T_2

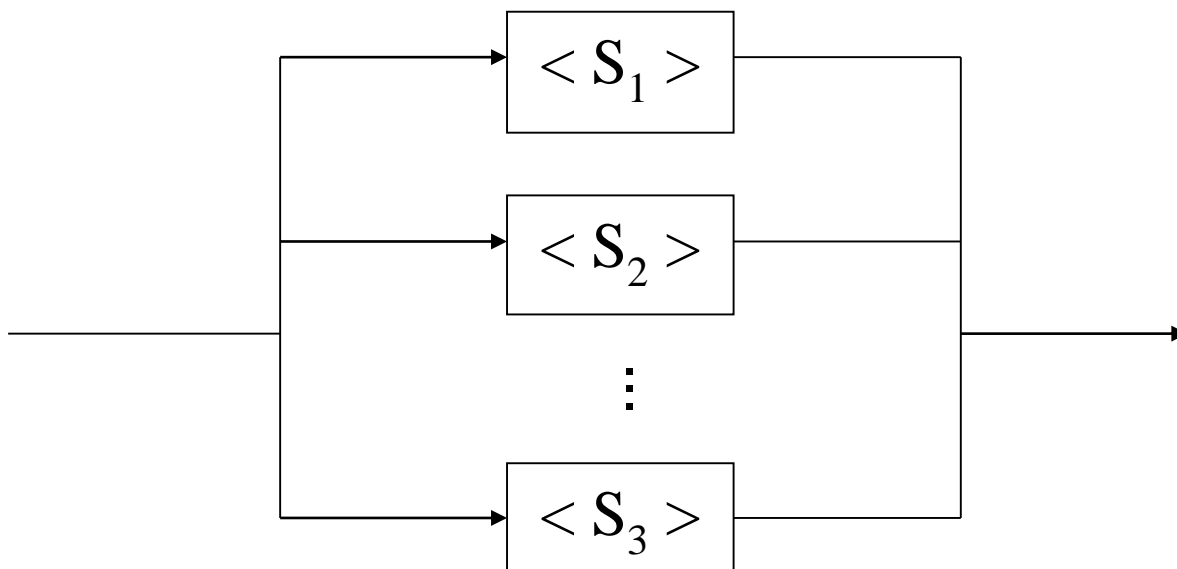
若非終端符號 $\langle A \rangle$ 之語法圖為



則其轉換敘述為
 A ;

規則 T_3

若選擇式語法圖為



則其轉換敘述為

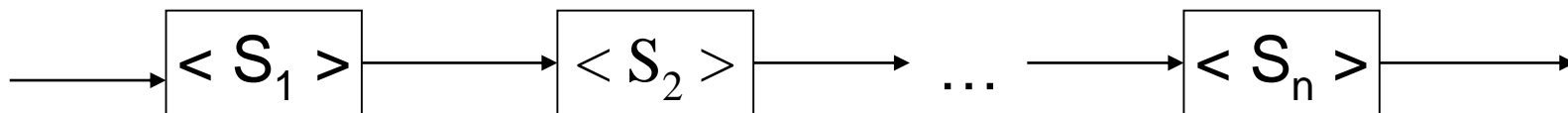
```
CASE CH OF
  start ( < S1 > ) : S1 ;
  start ( < S2 > ) : S2 ;
  ⋮
  start ( < Sn > ) : Sn ;
END ;
```

或寫成

```
IF CH = start ( < S1 > ) THEN S1 ELSE
IF CH = start ( < S2 > ) THEN S2 ELSE
  ⋮
IF CH = start ( < Sn > ) THEN Sn ELSE
  ERROR ;
```

規則 T_4

若循序式語法圖為

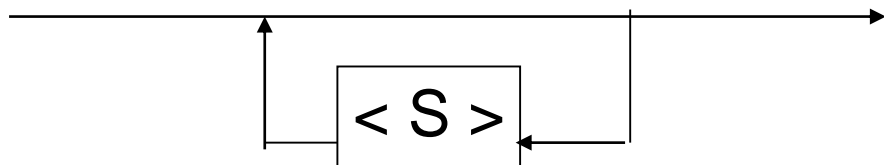


則其轉換敘述為

```

BEGIN
  ...
  S1 ;
  S2 ;
  :
  Sn ;
END ;
  
```

規則 T_5
若重複式語法圖為



則其轉換敘述為

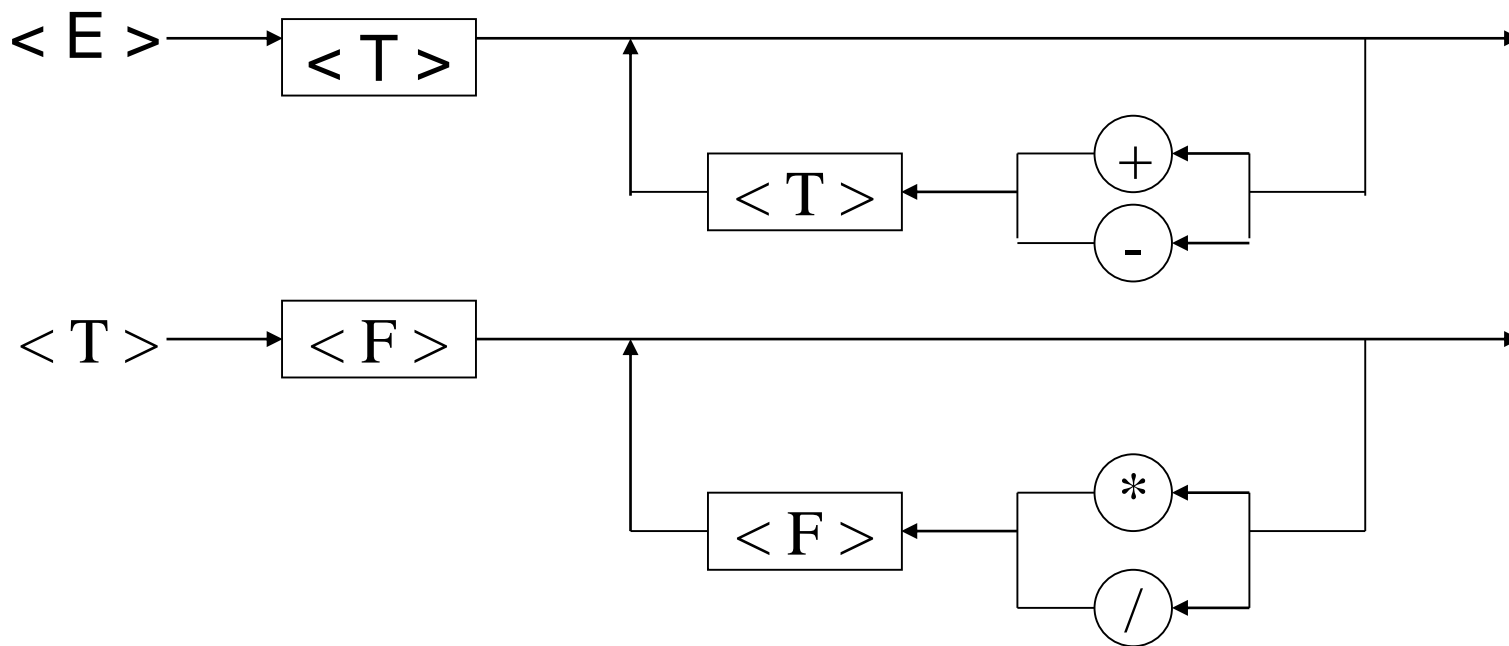
WHILE CH = start (< S >) DO S ;

【例五】語言E以EBNF定義為

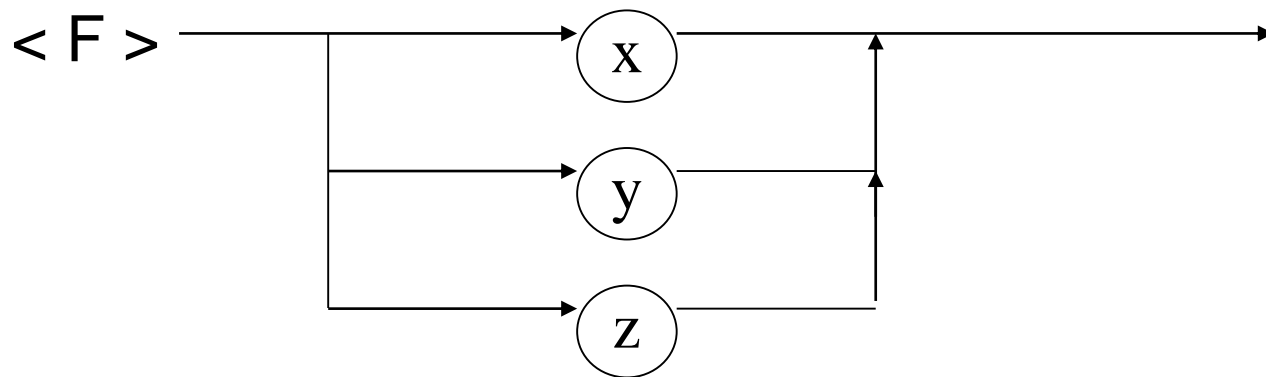
$$P_1 \quad \langle E \rangle ::= \langle T \rangle \{ \backslash + \mid - \backslash \langle T \rangle \}$$
$$P_2 \quad \langle T \rangle ::= \langle F \rangle \{ \backslash * \mid / \backslash \langle F \rangle \}$$
$$P_3 \quad \langle F \rangle ::= x \mid y \mid z$$

請製作相對應之語法圖，並根據語法圖設計程式。

【解】 P_1, P_2 依 G_4, G_3, G_5 得相對應語法圖為



P_3 依 P_4 得



若把 $\langle F \rangle$ 代入 $\langle T \rangle$ ，把 $\langle T \rangle$ 代入 $\langle E \rangle$ 得 $\langle E \rangle$ 之合併語法圖

