



# Introd2

<<Introd2.ppt>>

## 二. 語法分析階段(**Syntax Analysis phase**)

### (一) 功能：

辨認語言的結構，供解釋階段呼叫對應的動作副程式 (**action routine**)產生中間形式（或矩陣）用。

### (二) 資料基底：

1. 輸入—齊—符號表（**uniform symbol table**）。

說明：本符號表於語彙分析階段建立，用來取代原始程式，使得往後階段的處理簡化。

2. 中間的儲存體—堆疊（**stack**）。

說明：堆疊可視為一有序串列（**order list**），其插入及刪除都在該串列的特定一端進行，亦即其作業方式採用後進先出（**Last In First Out, LIFO**），也就是最後加入者最先被刪除。

### 三. 解釋階段（**Interpretation phase**）

#### （一）功能：

呼叫動作副程式（**action routine**），執行下列工作：

1. 產生原始程式的中間形式。
2. 將識別字的所有資料屬性加入識別字表中。
3. 將中間矩陣的矩陣行之暫時計算結果的所有資訊（包括矩陣行代號及所有屬性）逐項加入暫時儲存表（可視為識別字表的一部份）中。

#### （二）資料基底：

1. 齊一符號表。
2. 堆疊—存放目前正被語法分析及解釋階段處理的齊一符號。
3. 識別字表。說明：
  - （1）名稱於語彙分析階段填入。
  - （2）資料屬性於解釋階段填入。
  - （3）位址於位置分配階段填入。

4. 矩陣—程式的最初中間形式，矩陣項目（**entry**）的格式如下：

齊一符號   齊一符號   齊一符號   鏈結   鏈結

運算符號	運算元1	運算元2	向前指標	向後指標
------	------	------	------	------

其中向前指標存放前面一個矩陣項目的註標（**index**）。

向後指標存放後面一個矩陣項目的註標。

說明：

- (1) 為了使最佳化階段能方便地增加或刪除矩陣項目，因此每一個矩陣項目增加兩個鏈結欄，一為向前指標。
- (2) 程式指令可分為算述指述（**arithmetic statement**）、非算數指述（**nonarithmetic statement**）及不執行指述（**nonexecutable statement**）三類，本階段算術及非算術指述將產生間矩陣，而不執行指述只把有關資訊加入識別字表中，但不產生中間矩陣；分述如下頁：

— 算術指述—以堆疊為中間儲存體，按照算術式運算的優先順序逐步建立中間矩陣。

例如：算術式  $A = B - C * D / (E + F)$  對應的中間矩陣如下：

矩陣 行數	運算符號	運算元1	運算元2	向前指標	向後指標
1	*	C	D	0	2
2	+	E	F	1	3
3	/	M1	M2	2	4
4	-	B	M3	3	5
5	=	A	M4	4	0

其中M1表示矩陣的第一項計算結果，M2、M3、及M4類似。

註：產生對應算術式之中間矩陣的作法是首先利用堆疊為中間儲存體，產生算術式的後置表示式（**postfix**），然後輸入該後置表示式，又利用堆疊為中間儲存體，產生中間矩陣。

— 非算術式指述—只**DO**、**IF...THEN...ELSE...**、**GO TO**等 非算術式指述，分別依格式填入矩陣中；其中可能有些指述（例如**IF**指述）將佔用矩陣一個以上的項目。

例如：指令**RETURN (COS)**；及**END**；用中間矩陣表示如下：

矩陣 行數	運算符號	運算元1	運算元2	向前指標	向後指標
1	RETURN	COS			
2	END				

— 不執行指述—譬如PL/1中**DECLARE**指令或**FORTRAN**中**DIMENSION**、**INTEGER**及**REAL**等指令屬於不執行指述，本階段只把不執行指述所定義的識別字之屬性加入識別字表中，而不產生中間矩陣。

(3) 矩陣的運算元包括四種型態的齊一符號：**IDN**、**LIT**、**TRM**及**MTX**；以符號**MTX<sub>n</sub>**表示矩陣第**n**項的計算結果，將它存放在暫時儲存表（**temporary storage table**）（可視為識別字表的一部份）中，而符號的位址在位置分配階段才得到。

5. 暫時儲存表（可視為識別字表的一部份）--於本階段建立，本表包括所有型態為**MTX**的符號（指矩陣的中間計算結果）及其屬性，而符號的位址在位址分配階段才得到。

### (三) 演算法：

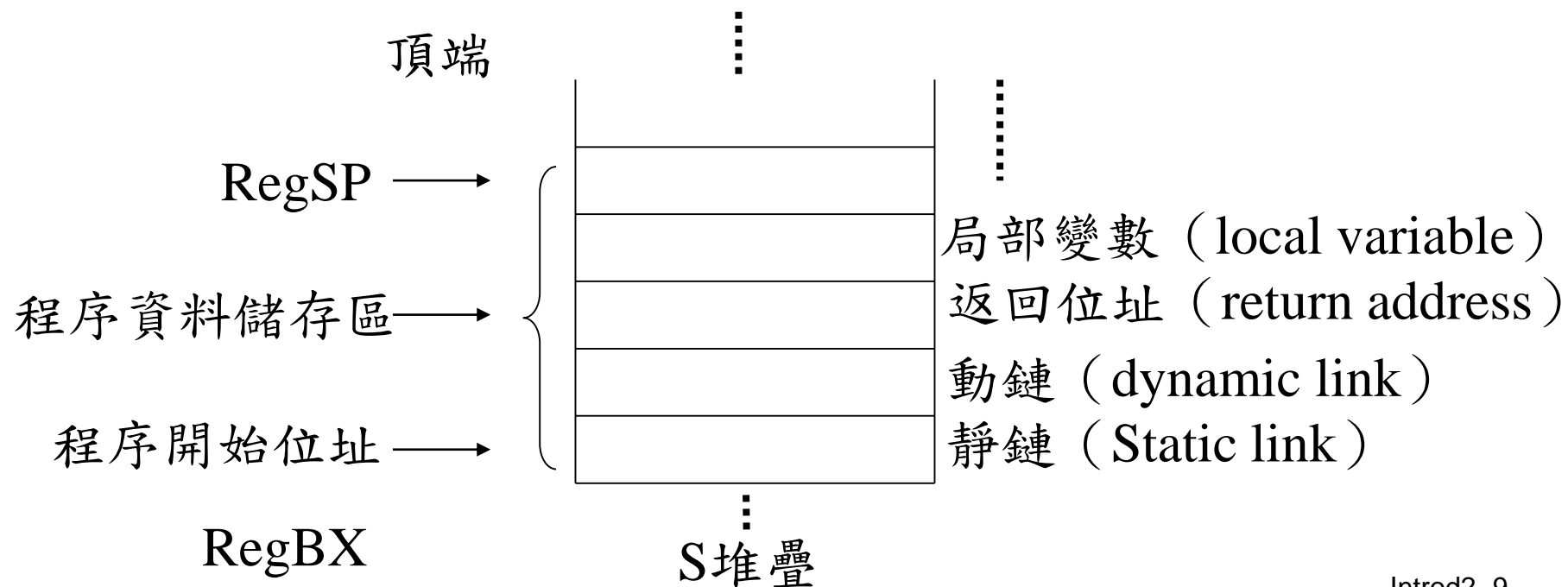
本階段依照語法分析階段辨認出來的語言結構分別呼叫對應的動作副程式，反覆作業，直到齊一符號表的符號全部輸入而且都予以處理為止。

	7		
IdTop →	6	GCD	PROC
	5	Y	VARIABLE
	4	X	VARIABLE
	3	N	CONSTANT
	2	M	CONSTANT
	1	TEST52	PROG
		NAME	KIND

註標    識別名稱                      屬性



識別字堆疊（**IdStack**），以LangX語言之程式**PROGRAM TEST52**；**CONST** **M=12**，**N=9**；**VAR** **X**，**Y**；**PROCEDURE** **GCD**；...為例，所建立之識別字堆疊。第一欄為識別名稱**NAME**，第二欄為屬性，只有四種，及程式**PROG**，程序**PROC**，常數**CONSTANT**，以及變數**VARIABLE**。堆疊頂端註標為**IdTop**。



每一程序之基本資料（動鏈、靜鏈、返回位址）及局部變數儲存於**S**堆疊之相對位置。**RegSP**為**S**堆疊頂端註標。

## LangX指令一覽表

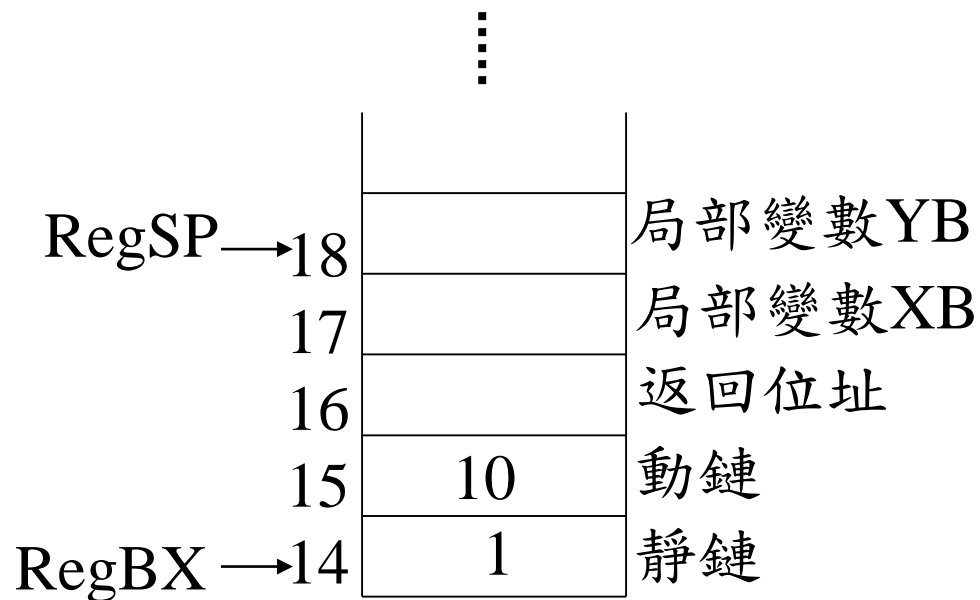
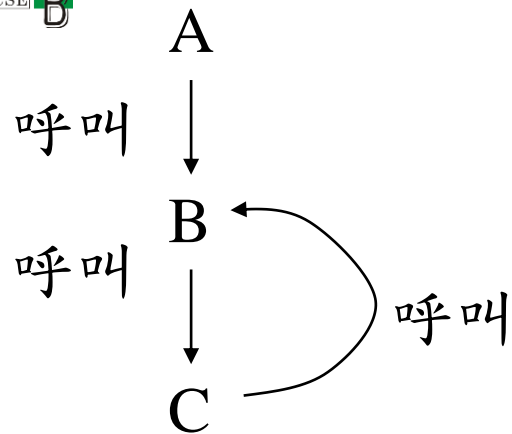
編號	指令 (instruction)			說明
	運算碼F	層次差L	位移A	
1	Allocate	0	A	A為配置記憶體之數量。
2	PushC	0	A	A為疊入之常數值。
3	PushV	0	A	疊入 (L, A) 之變數
4	PopV	0	A	疊出至位址為 (L, A) 之變數。
5	Call	0	A	呼叫位址為 (L, A) 之程序。
6	Jump	0	A	跳躍至 (0, A) 之位址。

(接下頁)

7	JumpC	0	A	依S堆疊頂端元素值若為偽則跳躍至位址 (0, A)。
8	Return	0	0	程序被呼叫完畢，轉回呼叫程序。
9	Negative	0	0	S堆疊頂端元素改變符號。
10	Addition	0	0	疊出S堆疊頂端兩元素，次頂端元素加 (+) 頂端元素，其結果再疊入。
11	Subtract	0	0	同上，不同的是相減 (-)。
12	Multiply	0	0	同上，不同的是相減 (*)。
13	Divide	0	0	同上，不同的是相減 (/)。

(接下頁)

14	EQ	0	0	疊出S堆疊頂端兩元素，比較次頂端元素是否等於(=)頂端元素，再將結果疊入。
15	NEQ	0	0	同上，不同的是比較不等於( $\neq$ )。
16	LT	0	0	同上，不同的是比較小於( $<$ )。
17	LEQ	0	0	同上，不同的是比較小於等於( $\leq$ )。
18	GT	0	0	同上，不同的是比較大於( $>$ )。
19	GEQ	0	0	同上，不同的是比較大於等於( $\geq$ )。



註標 S堆疊

## 程序B開始位址

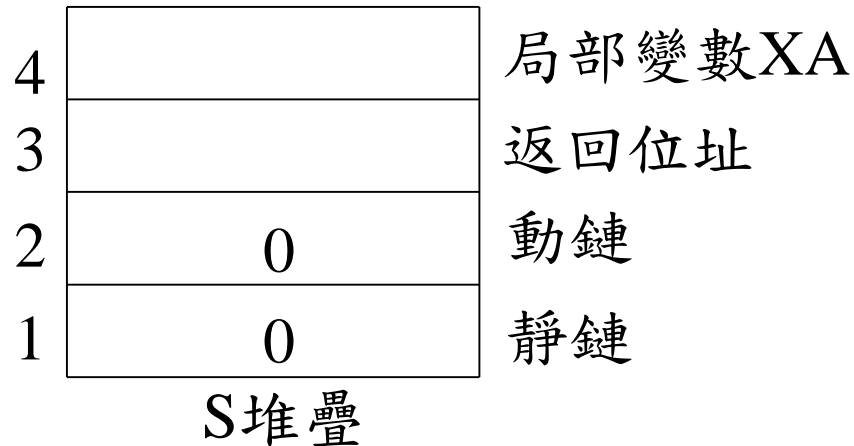
13		局部變數XC
12		返回位址
11	5	動鏈
10	5	靜鏈

## 程序C開始位址

9		局部變數YB
8		局部變數XB
7		返回位址
6	1	動鏈
5	1	靜鏈

註標                  S堆疊

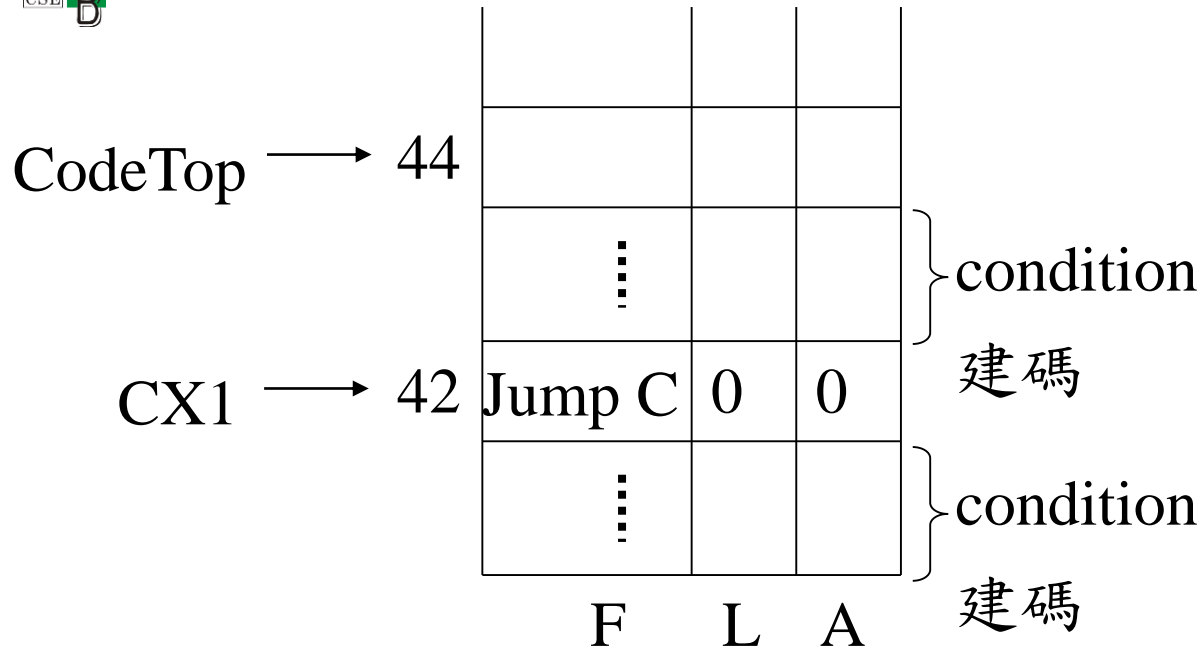
程序B開始位址



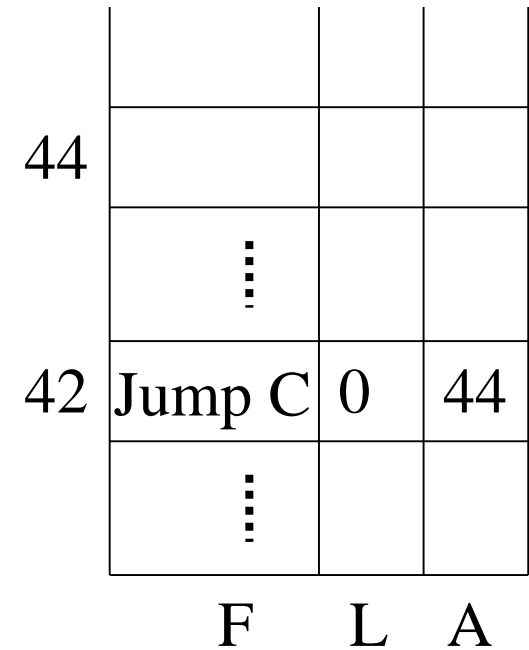
程序A開始位址

註標

依例提（圖5.36）所配置在S堆疊中各程序之資料儲存區。圖中左邊有箭頭之線為動鏈，已是呼叫之關係，右邊有箭頭之線為靜鏈，以示各程序之結構層次。注意最頂端程序B之動鏈為註標值10，而靜鏈為1。



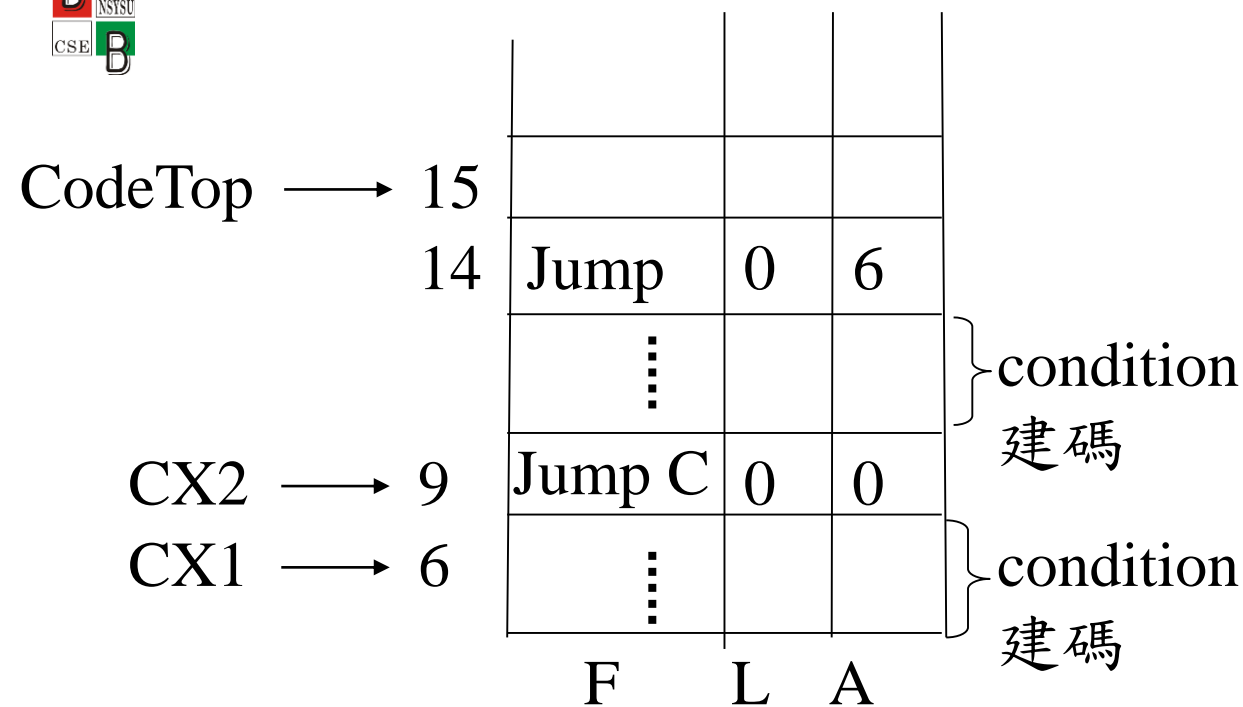
(a) condition建碼之後



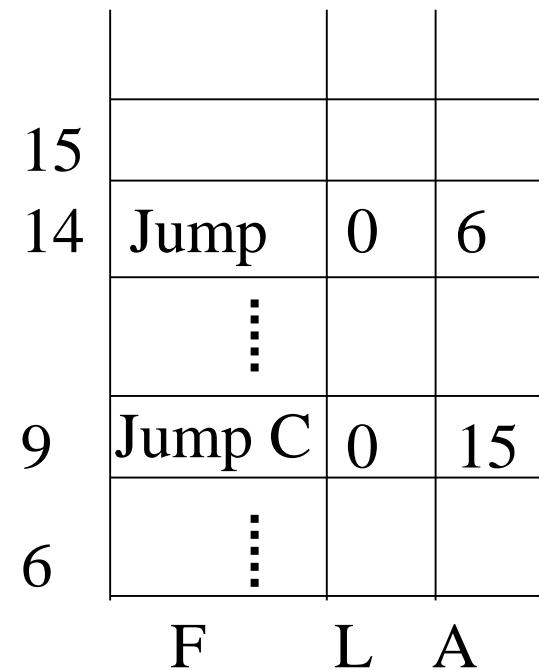
(b) statement建碼之後

敘述IF condition THEN statement經編譯程式建碼時，存在P碼指令堆疊CODE之情形。(a)是目的位址還不知道時，A欄為0。(b)是目的位址已知道為44（以TEST 53建碼為例）時填入CX 1（為42）處之A欄。





(a) condition建碼後，指令JumpC應跳躍至CodeTop之指令，但這時還未知，故A欄先設為0。而JumpC指令之位址設為CX2。



(b) statement建碼後，指令Jump應跳至condition建碼開始處CX1，因其為已知(6)。這時WHILE之下一敘述建碼的位址為已知（例如15），這時才填入CX2處之A欄。

以原始程式**TEST 53**中之敘述**WHILE  $X > Y$  DO  $X := X - Y$** ；為例，說明**WHILE**敘述建碼之情形。Condition為 $X > Y$ ，statement為 $X := X - Y$ ；(a)為JumpC之目的位址未知。(b)為JumpC之目的位址已知之情形。