

編譯程式簡介

編譯程式簡介

- 編譯程式的七個階段中，語彙分析(Lexical analysis)、語法分析(syntax analysis)、解釋(Interpretation)及與機器無關的最佳化(Machine independent optimization)等前面四個階段是“與機器無關而與語言有關”

編譯程式簡介

- 編譯程式的七個階段中，儲存位置的分配 (Storage assignment)、數碼產生 (Code generation) 及組合並輸出 (Assembly and output) 等後面三個階段是“與機器有關而與語言無關”。

編譯程式簡介

■ 解釋階段(Interpretation phase)

■ 功能：

依據語法分析階段辨認出來的語言結構之類型（包括算數式、非算數式及不能執行指述等），分別呼叫對應的動作副程式產生中間形式（或矩陣），供後面階段使用；同時將識別字的資料屬性加入識別字表中。

■ 在編譯的過程中使用“中間形式”的優點：

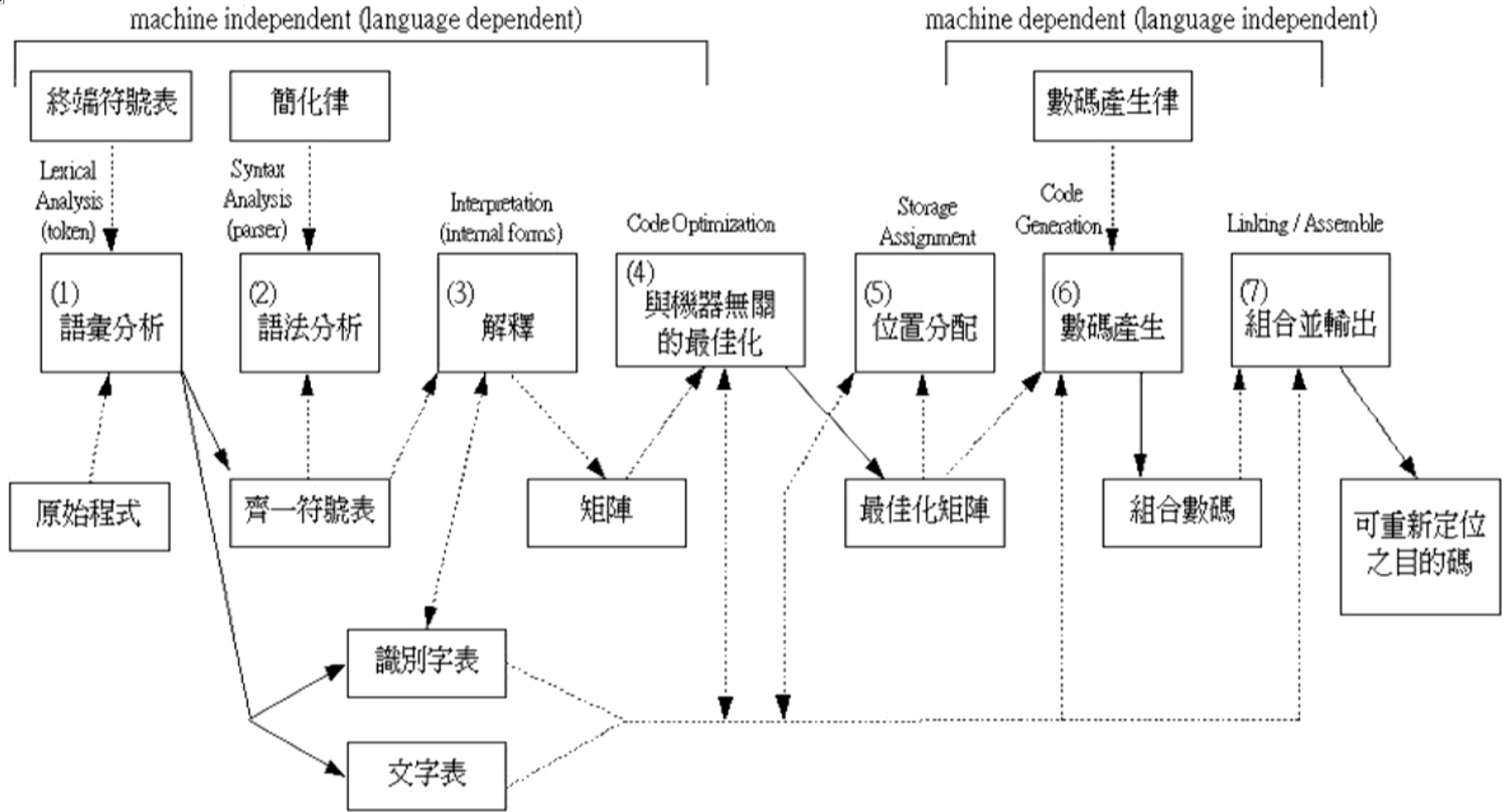
- 使得目的碼(object code)容易地達成最佳化。
- 使編譯的過程能在邏輯上劃分為“與機器無關”階段及“與機器有關”階段。

編譯程式簡介

- 與機器無關的最佳化階段(Machine-Independent Optimization phase)
 - 功能：
輸入中間矩陣，運用與機器無關的最佳化技巧，產生與機器無關的最佳化矩陣，並且更新識別字表及文字表的相關部分。

編譯程式簡介

- 簡介四個“與機器無關的最佳化”技巧如下：
 - 刪除共同的副式子(common subexpression)。
 - 將常數(constants)間的運算先行計算。
 - 布林式子(Boolean expression)的最佳化。
 - 將迴路(loop)中不變的計算式子移出迴路外。



註：實線表示建立資料基底(data bases)
虛線表示引用或更新資料基底

編譯程式的結構圖

Example 1

- Suppose the various functions of a compiler can be classified into the following six phases:

- (A) Code generation phase
- (B) Code optimization phase
- (C) Interpretation phase
- (D) Lexical phase
- (E) Storage assignment phase
- (F) Syntax phase

Based on INPUT/OUTPUT relationship, give the processing sequence of these phases in a typical compiler.

- Ans : (D)(F)(C)(B)(E)(A)

Example 2

- During a compilation, the identifier table will be modified by
 - (a) Lexical Analysis
 - (b) Syntax Analysis
 - (c) Interpretation
 - (d) Storage Assignment
- Ans : (a)(c)(d)

Example 2 說明

- 下列三個階段會修改identifier table的內容：
 1. Lexical Analysis Phase—identifier table於本階段建立，只填identifier的名稱到名稱欄中。
 2. Interpretation Phase—本階段加入identifier的所有屬性(attributes)到identifier table中。
 3. Storage Assignment Phase—本階段分配儲存體給各identifier，而且將分配給各identifier的位址加入identifier table中。

Example 3

- What are the advantages of using an intermediate language in a compiler?
List the advantages and disadvantages of using an intermediate language in a compiler.

Example 3

Ans :

1. 在編譯過程中採用中間的語言 (intermediate language)之要點如下：
 1. 使得目的數碼(object code)之最佳化容易地達成。
 2. 使編譯的過程能在邏輯上劃分為“與機器無關”階段及“與機器有關”階段。

Example 3

註：

1. “與機器無關的” 階段包括語彙分析(Lexical analysis)、語法分析(syntax analysis)、解釋(Interpretation)及與機器無關的最佳化(Machine independent optimization)等階段
2. “與機器有關的” 階段包括位置分配(Storage assignment)、數碼產生(Code generation)及組合並輸出(Assembly and output)等階段。 。

Example 3

2. 在編譯的過程中採用中間語言之缺點如下：
- 在編譯的過程中採用中間的語言，將使得反覆處理的次數(passes)增多，因此編譯的速率降低。

11	WHILE	WhileSymbol
10	VAR	VarSymbol
9	THEN	ThenSymbol
8	PROGRAM	ProgSymbol
7	PROCEDURE	ProcSymbol
6	IF	IfSymbol
5	END	EndSymbol
4	DO	DoSymbol
3	CONST	ConstSymbol
2	CALL	CallSymbol
1	BEGIN	BeginSymbol
0		IDENTIFIER

註 WORD WordSymbol
 標 字組 字組符號

LangX 語言中之字組符號表。註標
 1~11 為保留字之字組及符號。註
 標 0 為變數或常數識別字之字組及
 符號。

其他	NUL
;	Semicolon
.	PERIOD
,	COMMA
=	EQUAL
)	RightParenthesis
(LeftParenthesis
/	SLASH
*	TIMES
—	MINUS
+	PLUS

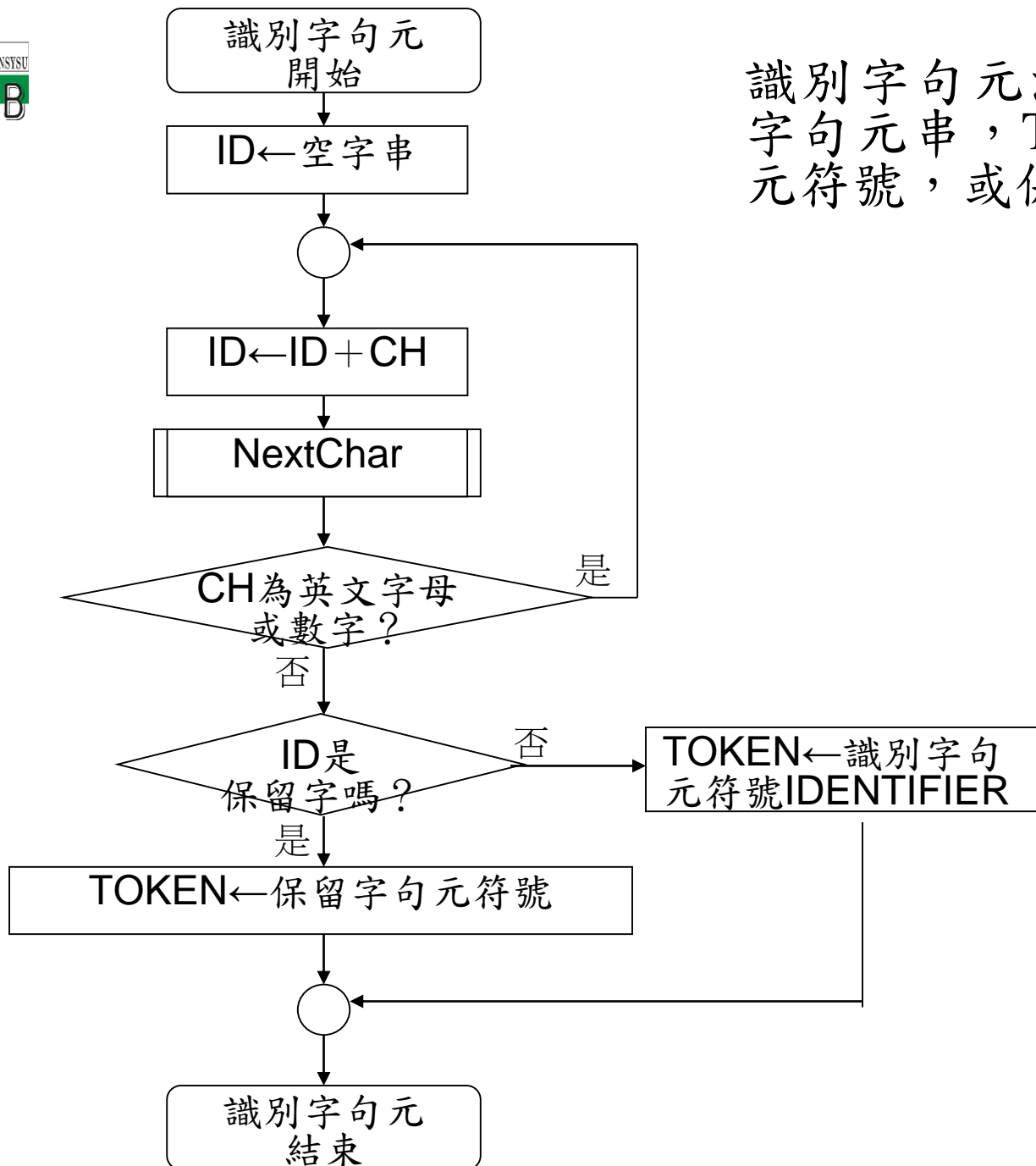
CH
字元

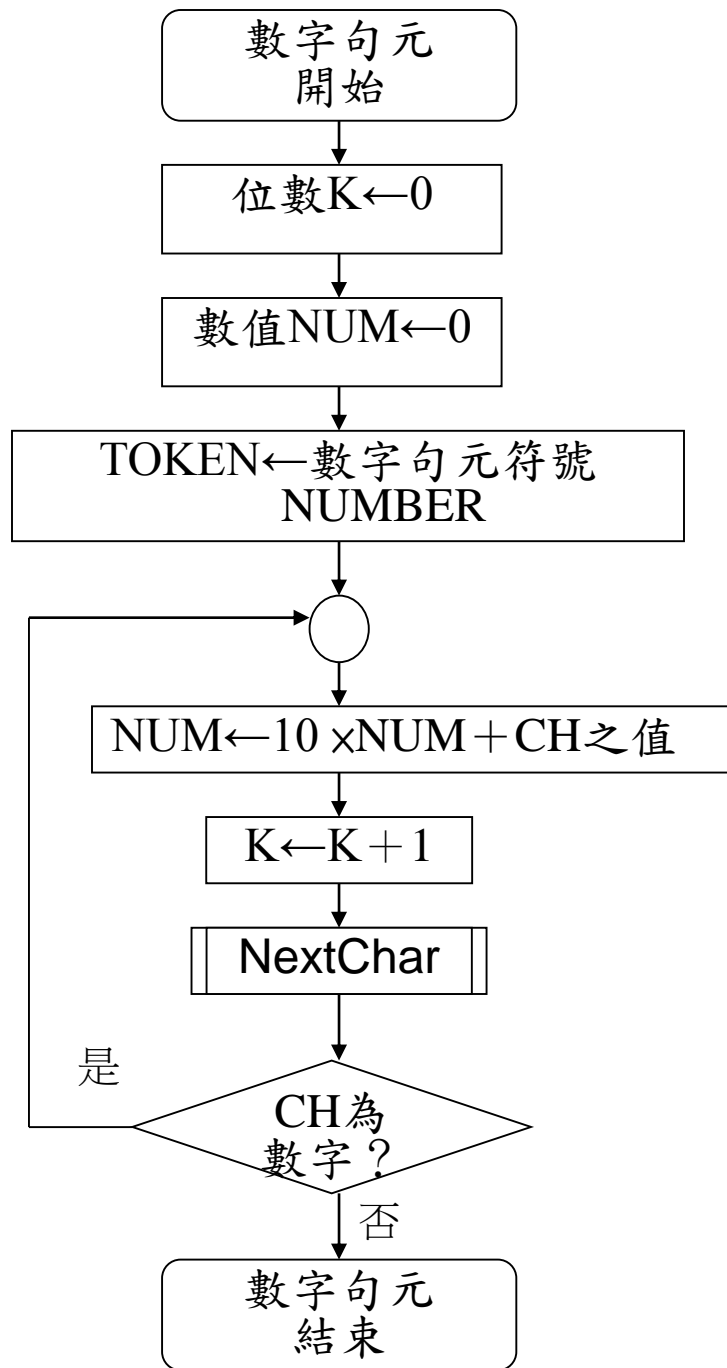
CharSymbol
字元符號

數字	NUMBER
>	GREATER
>=	GreaterEqual
<	Less
< >	NotEqual
<=	LessEqual
: =	BECOMES

句元 句元符號
單一字元所不能判斷之
符號表。

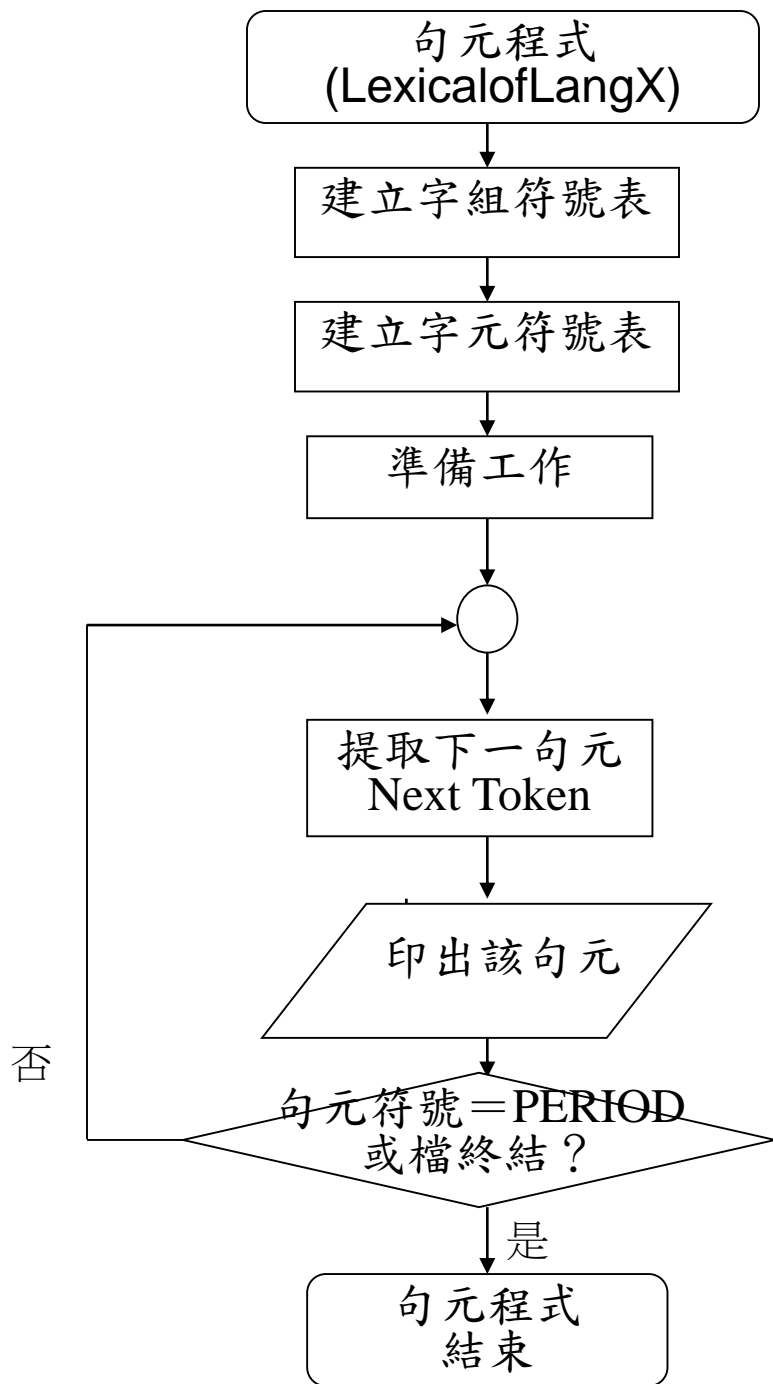
識別字句元流程圖。ID為識別字句元串，TOKEN為識別字句元符號，或保留字句元符號。





數字句元流程圖。K為數字之位數，NUM為數字之值。CH為下一個字元。TOKEN為數字句元符號。

句元程式 (語彙分析 Lexical of LangX) 之主要流程圖。



■ 語彙規則的BNF表示法

$$\langle idn \rangle ::= \langle letter \rangle | \langle letter \rangle \langle letter_digit \rangle_1^{30}$$

$$\langle letter_digit \rangle ::= \langle letter \rangle | \langle digit \rangle$$

$$\langle letter \rangle ::= A | B | \dots | Z$$

$$\langle digit \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

$$\langle lit \rangle ::= \langle digit \rangle_1^{15}$$

$$\begin{aligned} \langle trm \rangle ::= & \langle OP \rangle | DECLARE | END | PROCEDURE | RETURN | i | j | \\ & | GF | C | STATIC | FIXED | BINARY | \phi \end{aligned}$$

$$\langle OP \rangle ::= + | - | * | \div | ** | =$$

舉例說明：下圖包含一原始程式、終端符號及原始程式經語彙分析階段後產生之齊一符號表、識別字及文字表。

LBL	:	PROCEDURE	(RATE	,	START	,	FINISH)	;	
		DECLARE	(COST	,	RATE	,	START	,		
		FINISH)	FIXED	BINARY	(31)	STATIC	;	
		COST	=	RATE	*	(START	-	FINISH)	+
		3	*	RATE	*	(START	-	FINISH	-	50
)	;								
		RETURN	(COST)	;					
END	;										

(a)經分割為字彙的原始程式例子



終端符號表 (Terminal Symbol)

分隔
字元 其他

符號

1	:	Yes	
2	;	Yes	
3	(Yes	
4)	Yes	
5	,	Yes	
6		Yes	
7	PROCEDURE		
8	DECLARE		
9	RETURN		
10	END		
11	+	Yes	
12	-	Yes	
13	*	Yes	
...	

齊一符號表 (Uniform Symbol)

類別 指標

IDN	1
TRM	1
TRM	7
TRM	3
IDN	2
TRM	5
IDN	3
TRM	5
IDN	4
TRM	4
TRM	2
TRM	8
TRM	3
...	...

LBL
:
PROCEDURE
(
RATE
,
START
,
FINISH
)
;
DECLARE
(

識別字表

名稱	屬性	位址
LBL	解釋階段填入	位置分配階段填入
RATE		
START		
FINISH		
COST		

文字表

文字	基底	尺度	準確度	其他	位址
31	DECIMAL	FIXED	2		位置分配 階段填入
3	DECIMAL	FIXED	1		
50	DECIMAL	FIXED	2		