

## LL(1) Parser 實作

- ★ 使用自己熟悉的程式語言實作
  - ★ Recursive-descent LL(1) 或 table-driven LL(1) 皆可
  - ★ 指定某個 CFG 之後，可重複輸入不同的 input token strings
  - ★ Demo：8 個 CFG，各 5 個 input token strings
- 
- ★ 輸入
    - ☆ CFG：txt 文字檔，可於執行時輸入檔名
    - ☆ Input token string：可用 txt 文字檔輸入，或直接輸入
  - ★ 輸出
    - ☆ Parsing table：non-terminal 和 terminal symbols 依 CFG 中出現順序排序
    - ☆ Applying rules 順序
    - ☆ Accept or error
    - ☆ Stack 變化狀況 (optional)
  - ★ CFG
    - ☆ 「>」代表  $\rightarrow$  「lamda」代表  $\lambda$
    - ☆ 有編號
    - ☆ Symbol 之間用空白鍵間隔
  - ★ Input token string
    - ☆ Token 長度不限制為單一字元
    - ☆ Input token string 長度不限制

1 Start  $\rightarrow$  E \$  
 2 E  $\rightarrow$  Prefix ( E )  
 3       | v Tail  
 4 Prefix  $\rightarrow$  f  
 5       |  $\lambda$   
 6 Tail  $\rightarrow$  + E  
 7       |  $\lambda$

	(	)	v	f	+	\$
Start	1		1	1		
E	2		3	2		
Prefix	5			4		
Tail		7			6	7

1 Start > E \$  
 2 E > Prefix ( E )  
 3 | v Tail  
 4 Prefix > f  
 5 | lamda  
 6 Tail > + E  
 7 | lamda

( v + f ( v ) ) \$   1   2   5   3   6   2   4   3   7  
 v \$   1   3   7  
 f v + ( ( v ) ) \$   1   2   4   Error

1 S  $\rightarrow$  A B c \$  
 2 A  $\rightarrow$  a  
 3       |  $\lambda$   
 4 B  $\rightarrow$  b  
 5       |  $\lambda$

	c	a	b	\$
S	1	1	1	
A	3	2	3	
B	5		4	

1 S > A B c \$  
 2 A > a  
 3 | lamda  
 4 B > b  
 5 | lamda

a b c \$   1   2   4  
 b c \$   1   3   4  
 a b \$   1   2   4   Error