**LL(1) Parser實作**

**分組 & demo方式**

* 最多5人一組
* 110/1/5 (二) 3:30pm各組派代表來demo (地點再通知)
* 110/1/6 (三) 課堂時間，各組現場抽2位同學demo及講解程式碼

**實作要求**

* 使用自己熟悉的程式語言實作
* Recursive-decent LL(1) parser：Chap2及Chap5範例CFG
* Table-driven LL(1) parser：直接輸入CFG
* 輸入
* 指定某個CFG後，可以重複輸入不同的token string
* CFG (for table-driven parser)
* txt文字檔，可於執行時輸入檔名 (請**勿**將檔名寫在程式中)
* 有編號
* > 代表 → L代表 λ
* Symbols之間用**空白鍵**間隔
* Token string
* 可由txt文字檔輸入，或直接輸入
* Token string長度不限制，token長度不限制為單一字元
* Tokens之間用**空白鍵**間隔
* 輸出
* Parsing table (for table-driven parser)
* Non-terminal依CFG中出現順序排序
* Terminal依英文字母順序排序 ($放最後)
* Applying rules順序
* Accept or Error(Expected *token*) or Error(*Non-terminal* vs. *Terminal*)

1 S → A C $

2 C → c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | q | $ |
| S | 1 | 1 | 1 |  | 1 | 1 |
| C |  |  | 2 | 3 |  | 3 |
| A | 4 | 5 | 5 |  | 5 | 5 |
| B |  | 6 | 7 | 7 | 7 | 7 |
| Q |  |  | 9 |  | 8 | 9 |

3 | λ

4 A → a B C d

5 | B Q

6 B → b B

7 | λ

8 Q → q

9 | λ

1 S > A C $

2 C > c

3 | L

4 A > a B C d

5 | B Q

6 B > b B

7 | L

8 Q > q

9 | L

**a b b d c $**

1 4 6 6 7 3 2 Accept

**c $**

1 5 7 9 2 Accept

**c c $**

1 5 7 9 2 Error(Expected $)

**a b a d c $**

1 4 6 Error(B vs. a)