

Compiler Laboratory #2

NTUCSIE Compiler 2011 (03/07)

本實驗目的是計算 first-set, follow-set 以及 predict-set。

輸入檔案：敘述 grammar 的文字檔，參見附檔之 lab_2/*.g。檔案格式規範如下：

- 當 terminal 是 token class 時，表示方式為 id；id 的規範同實驗一。
- 當 terminal 是 lexeme 時，表示方式為 string；string 的規範亦同實驗一。
- Nonterminal 的表示方式亦為 id。
- 對一個 nonterminal，其 production rules 必須寫在一起。格式如下：
nonterminal : rule_1 | rule_2 | ... | rule_n ;
其中 : | ; 是 meta-characters。
- right-hand-side 為 λ 時，則省略 rule_i。
- 整個 grammar 第一條 production rule 的 nonterminal 即為 start symbol。
- 依照上述規則，一個 id 可能為 nonterminal 或是 token class。當一個 id 曾經出現在 production rule 的 left-hand-side，則視其為 nonterminal；否則視其為 token class。
- 可以加入 whitespace 與註解，規範同實驗一。
- 本實驗的輸入不需要考慮不合理、不合法的 grammar 寫法。

輸出檔案：輸出結果假如可以用表格表示，使用方便性會較佳；如下列 excel：



	A	B	C	D	E
1	LHS	RHS	FIRST	FOLLOW	PREDICT
2	E	Prefix (E)	f v ((eof))	f (
3		v Tail			v
4	Prefix	f	(null) f	(f
5		(null)			(
6	Tail	+ E	(null) +	(eof))	+
7		(null)			(eof))
8					

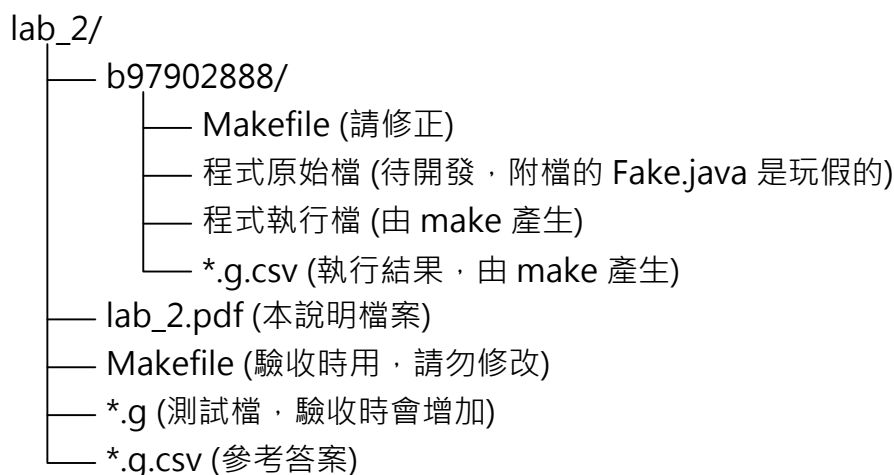
為達成此一目的，本實驗輸出為 csv 檔，參見附檔之 lab_2/*.g.csv。詳細規範如下：

- 輸出檔案名稱即輸入檔案名稱加上 .csv。
- Csv 檔內各欄位均用 quoted field 方式儲存，即前後都加雙引號。
- 第一列必須是標題。
- 第二列以下是各個 production rule，次序同輸入檔案。
- 一個 nonterminal 的第二個以上的 production rules，不用寫出 LHS。
- 一個 nonterminal 僅須在其第一個 production rule 那列，寫出其 first-set 以及 follow-set。
- 每個 production rule，都須寫出其 predict-set。
- Lexeme 不用印出左右雙引號。
- λ 與 $\$$ 分別印成 (null) 與 (eof)。
- 同一格有兩個以上的 symbols，中間用一個空白隔開。整格前後亦加一個空白。
- FIRST, FOLLOW, PREDICT 那三種 sets，元素必須排序。印出的次序為：特殊符號 (λ 與 $\$$)、token classes (依一般 ASCII 字串次序排列)、lexemes (依一般 ASCII 字串次序排列)。比如，{ "if", int_number, float_number, "+", "*", $\$$ } 印出：
(eof) float_number int_number * + if

開發工具：語言不拘，工具不拘，只要寫入 Makefile 即可。建議善用實驗一 scanner 的成果。

執行平台：217 工作站實驗室之 linux α 。

執行環境：解壓縮附檔，即可得下列目錄結構。



執行步驟：

1. 於 lab_2/b97902888/ 目錄底下開發程式。必要時可以建立子目錄。

2. 修改 lab_2/b97902888/Makefile。

必須搭配驗收用之 lab_2/Makefile。搭配細節請見附檔兩個 Makefile 內之註解。

步驟 3~6 是驗收步驟。

3. cd lab_2

4. make clean STU=b97902888

效果必須清掉 lab_2/b97902888/ 底下之程式執行檔以及執行結果。

5. make judge STU=b97902888 GRAM=grammarF.g

效果包含：

(1) 產生程式執行檔。執行檔擺放位置不得超出目錄 lab_2/b97902888/ 之範圍。

(2) 產生執行結果 lab_2/b97902888/grammarF.g.csv。請留意目錄以及檔名。

(3) diff 執行結果 lab_2/b97902888/grammarF.g.csv 與參考答案 lab_2/grammarF.g.csv。若沒有比對出任何差異，則通過。

6. make judge STU=b97902888 GRAM=grammarE4.g

一次 make judge 餵入一份測試檔。

附檔提供的測試檔僅是範例。驗收時，實驗要求還是以本文件的規範為準。建議應多測試自創之測試檔。

7. cd lab_2/b97902888

8. make run GRAM=自創測試檔

效果為 5(1) 與 5(2)，不含 5(3) 的 diff。

結果繳交：請上傳一個 rar 或是 zip 檔到 ceiba 系統，結構必須為：

```
lab_2/
├── b97902888/
│   ├── Makefile
│   ├── 程式原始檔
│   └── IWantToSay.txt (optional)
```

解壓縮後，第一層必須是 lab_2。第二層必須是你的學號 (不要 888)，英文字母小寫。請參考附檔之結構。只需上傳程式原始檔 (不要 Fake.java) 以及 lab_2/b97902888/Makefile，其他檔案不得上傳。假如有話要交代助教，可以多加 IWantToSay.txt。

上傳後，建議自行下載一次，以確認繳交的是正確的版本。

評分原則：

1. 繳交截止時間為 4/16 週六中午 12:00。寬限時間為 1 小時。

2. 週六 13:00~週日 12:00 算遲交 1 天。週日 12:00~週一 12:00 算遲交 2 天。以下依此類推，除第 1 天外，均以 12:00 為分界點。
3. 遲交 1~7 日，每日扣得分 10%。第 8 日起，以 0 分計算。
4. 繳交格式不對，扣得分 20%。
5. 結果正確性與完整性佔 90%；部分通過則斟酌略減。
6. Program readability/maintainability 佔 10%。