1112919 丁柏凱 系統程式期末專案 Personal Work

我們作業的流程是:

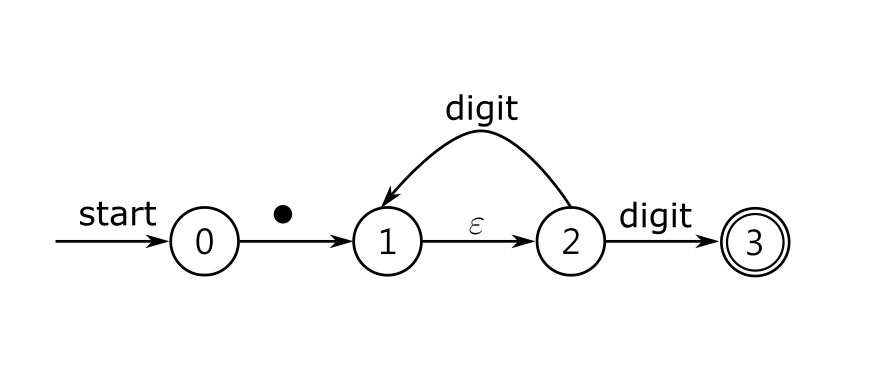
Regular Expression --> NFA (Dotted Item Method) --> DFA

--> Python Program

我主要負責處理從 NFA 圖形轉成 DFA 的 Transition table，以及將 DFA 的轉換應用到 Python Program 這兩部分。

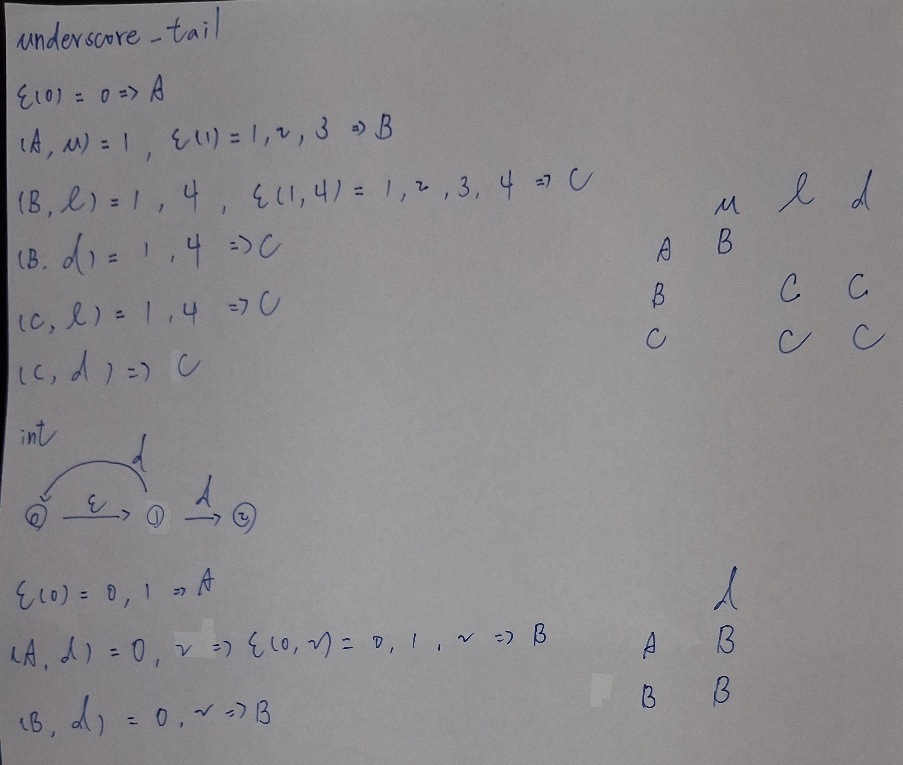
1. 將 NFA 圖形轉成 DFA 的 Transition table

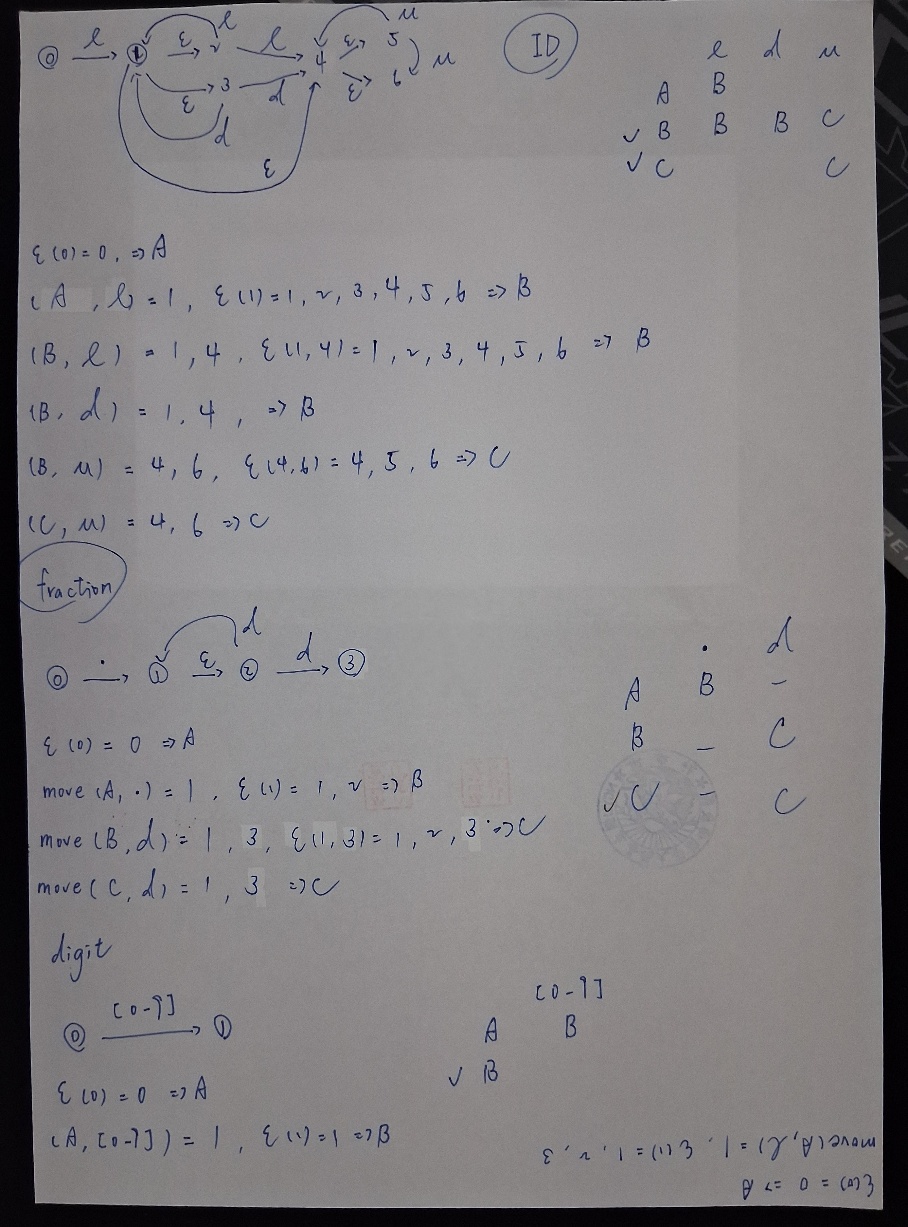
其他組員使用 Dotted Item 的方法，把正規表示式畫成 NFA 圖形，為了要放入報告中，我們使用了 inkscape 這個軟體，將手繪的 NFA 圖形轉成電子檔案，如下圖所示:

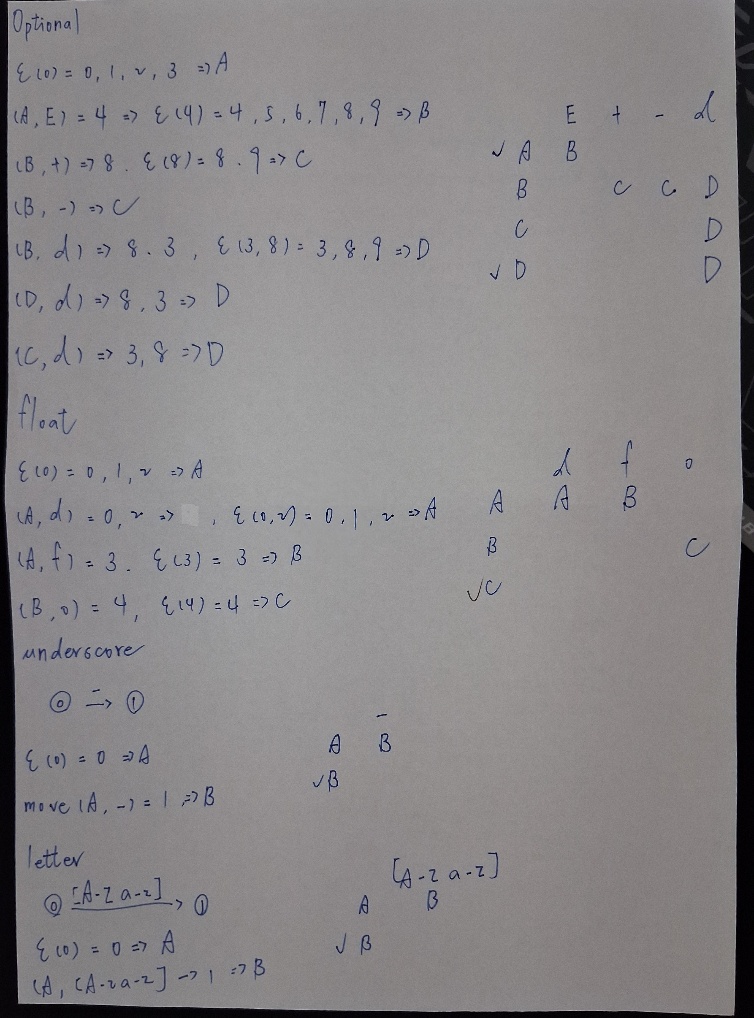


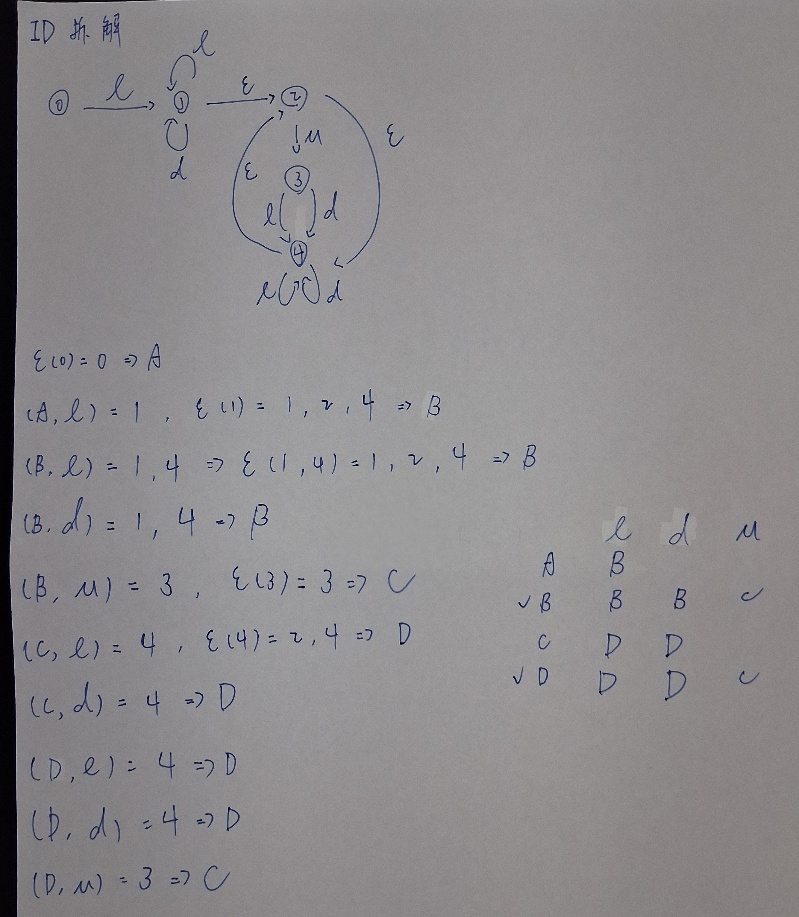
不過只有我會用這個軟體，所以這部分是由我先教其他組員怎麼使用 inkscape，交給他去畫圖，完成後我再進行檢查和修正。

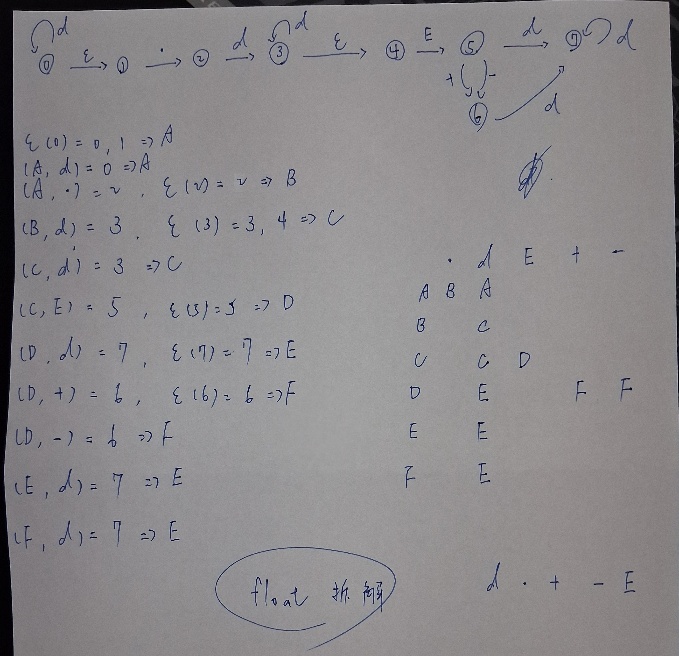
我依據 NFA 圖形，先用手寫的方式，計算出每一個的 DFA Transition table:











完成之後，也因為是要放入報告中，所以需要轉成電子文件，我畫的 NFA --> DFA 就由我自己來完成。

(Regular Expression --> NFA 是由其他組員完成)

我使用 Markdown + LaTeX 來完成，像是其中一個部分:

|  |  |
| --- | --- |
|  |  |
| Markdown | PDF |

(上傳檔案有附上完整的 .md 和 PDF 檔案)

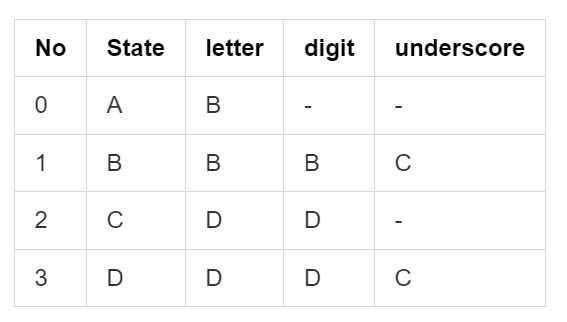
1. 將 DFA 的轉換應用到 Python Program

我們的程式是用 Python 來實作，全部都是我自己一個人寫的。(上傳檔案有附上程式)

輸入是讀取一個檔案，所以要在 index.py 同目錄下放一個 input.txt 作為輸入測資。

一開始想說，是不是全部的規則都要能夠被判斷出來，像是 underscore\_tail 這個規則，如果輸入 \_AErwev6，應該要被判斷成是 underscore\_tail，但是後面在處理 float 的時候，有遇到像是 fraction 完全被 float 覆蓋、輸入 E10 要被判斷成 optional\_exponent 還是 id 等等問題，後來才發現題目其實有要求要判別出哪幾個…

要如何將 DFA 的過程，套入到程式中呢?我是用 python 中的 dictionary。舉個例子來說明:

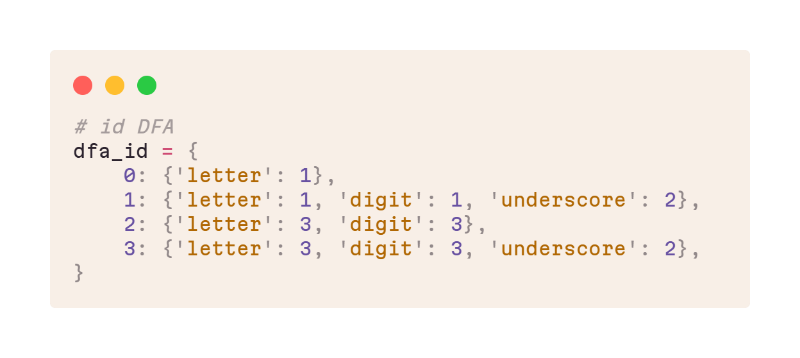


先把原本寫的 A~D 個狀態，標為 0~4

從原本初始狀態 A，也就是 0，如果讀了 letter，就會跑到狀態 B，也就是狀態 1，那程式中就寫上，狀態 0，讀取 letter，就切換到狀態 1。

那在狀態 B，也就是 1，如果讀取了 letter 或是 digit，還是會留在狀態 B，讀取 underscore 會轉換到狀態 C，那程式中就寫上，狀態 1，讀取 letter，切換到狀態 1，讀取 digit，切換到狀態 1，讀取 underscore，切換到狀態 2。後面依此類推。

我們就可以在 dictionary 中寫下這樣的狀態轉移:



再來還需要決定一下合法的結束狀態:

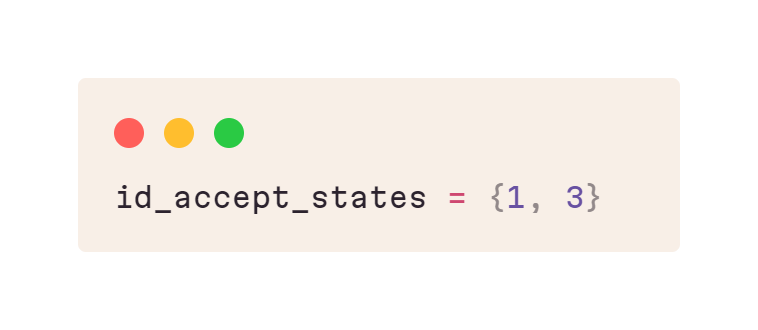
狀態 A 是初始狀態，當然不能當作是結束狀態。

而狀態 B，是從 A 讀了英文字母之後，或是 B 自己讀了字元或數字，也就是這串輸入的最後是英文字母或是數字，因此狀態 B 是可以作為結束狀態的。

狀態 C，是狀態 B 或 D 讀取了底線而來，也就是這串輸入的最後是底線，那根據規則，id 的結尾不可能是底線，所以狀態 C 不能當作結束狀態。

狀態 D，是由狀態 C 讀了英文字母或數字而來，也就是底線後面跟了英文字母或數字，或是從狀態 D 再讀英文字母或數字，因此這串輸入的最後面是英文字母或是數字，可以當作結束狀態。

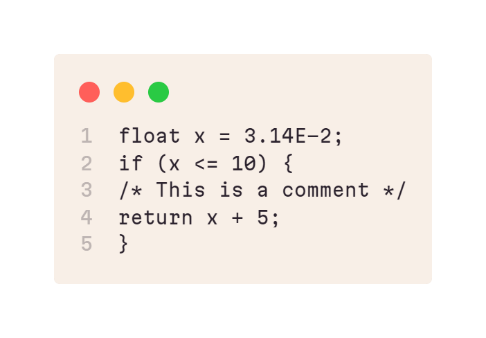
id 這個 DFA，可以做為結束狀態的有 B 和 D，也就是狀態 1 和 3，因此我們把合法的結束狀態記錄下來:



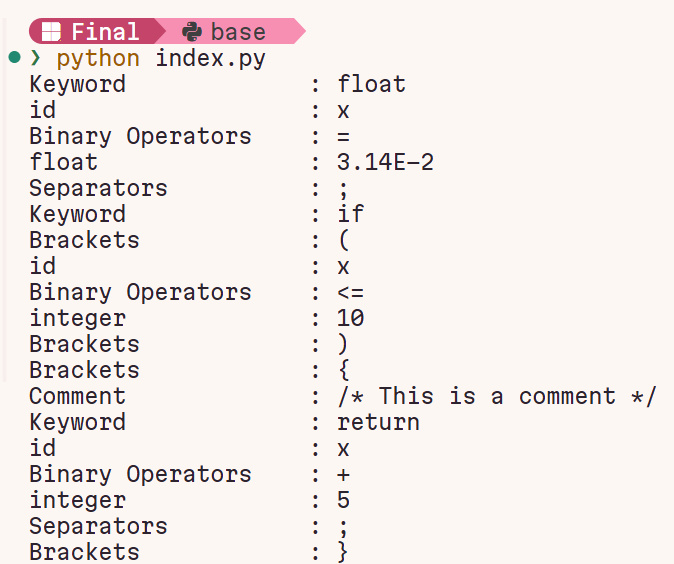
後面幾種也都是一樣的做法。

測試一下執行結果

輸入:



可以得到這樣的輸出:



Token 都有被判別出來，是沒有問題的。