

一、 個人簡歷表

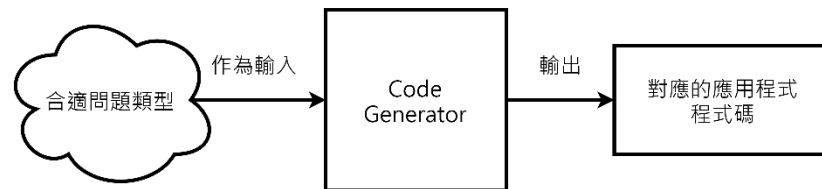
|        |   |    |   |      |   |                          |
|--------|---|----|---|------|---|--------------------------|
| 就讀大學   | 國立台中教育大學 資訊工程學系   |    |   |      |  |                          |
| 姓名     | 林仲鎧   | 性別 | 男 | 聯絡電話 |   | 0911841542<br>04-8810146 |
| 學習歷程   | <ul style="list-style-type: none"> <li>➤ 基礎計算機與網路知識，應用數學能力</li> <li>➤ C 與 C++ 程式撰寫</li> <li>➤ 利用 JAVA 撰寫之 Assembler</li> <li>➤ 微處理機系統操作及寫入(MSP430)，嵌入式系統實作遠端網路監控一設置之空氣偵測裝置</li> <li>➤ 軟體工程實務之基本能力</li> <li>➤ Python 與其他擴充功能之程式撰寫(e.g. 爬蟲)</li> <li>➤ 利用 CLIPS 開發之系所排課系統</li> <li>➤ 英文與日文基本聽說讀寫能力</li> </ul> |    |   |      |   |                          |
| 程式基本能力 | C、C++、MATLAB、JAVA、Python、CLIPS 開發工具   |    |   |      |   |                          |
| 經歷     | <ul style="list-style-type: none"> <li>➤ 大專程式能力檢定(CPE) 三題</li> <li>➤ 2017 軟體工程課程 助教</li> <li>➤ 2015、2016 國防部暑期二階段軍事訓練役結訓，第二階段任學員長職務</li> <li>➤ 2017 畢業專題: CLIPS 的 rule-based 系所排課系統</li> </ul>  |    |   |      |   |                          |
| 個人興趣   | 桌球、棋類策略遊戲、電視電影與日本動畫   |    |   |      |   |                          |

## 二、 畢業專題

### (一) 以規則基(Rule-based)方式開發之系所排課系統

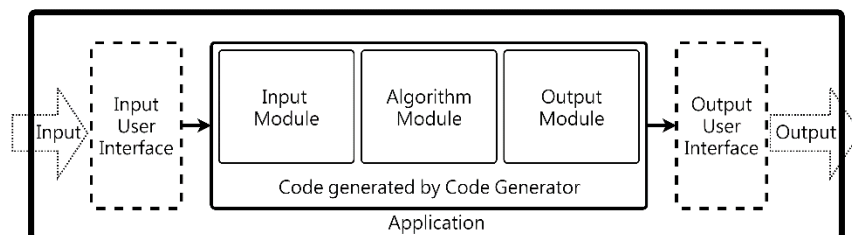
研究發想源於系辦每學期都要排定系上的課表，這複雜的工作一直以來都是由系辦人員手動完成，然而撰寫一個系統來自動化完成該工作是比较有效率的做法。

由於此類 Time-tabling 問題適合用於描述物件間的事實、限制、規則與互動關係，因此規則基(rule-based)的程式語言較符合此計畫的需求。C Language Integrated Production System (CLIPS) 屬於 rule-based 類型的編程語言，該類語言的運作方式主要為建立事實 (fact)與規則



(rule)，然後不斷推斷現時事實符合的規則並執行規則定義的處理流程及建立新的事實或規則。因此開發過程中只需定義相關的限制與事實，規則基(rule-based)便能以相較於一般循序式(procedural-based)快速的方式產生結果。

本專題利用規則基(Rule-based)方式開發系所排課系統，研究範圍包含輸入資料的內容、輸入資料的簡化、排課規則的制定、排課機制的演算法、效能加速。說明如下：



1. 輸入資料的內容：
  - a. 課程資料：包含課程名稱、課程時段、必選修類型、任課教授、開課年級、喜好資訊(時段上的喜好、受其他課程影響的喜好)
  - b. 教授資料：包含姓名、職位、研究領域、喜好資訊(時段上的喜好、受其他課程影響的喜好)
  - c. 教室資料：包含名稱、地點、管理單位、設備
2. 輸入資料的簡化：

利用 Code Generator 作輸入資料簡化，透過輸入自定義格式(自然語言形式)之輸入資料，利用自動化程式碼生成，產生機器能夠閱讀的格式(程式碼形式)，大幅提高系統的可閱讀性。
3. 排課規則的制定：
  - a. 基本限制：自然上無法違反的限制。

Ex: 一個老師不能同時出現在多個時段
  - b. 可自定義限制：可自由定義的限制
    - i. 彈性限制：可滿足、可不滿足

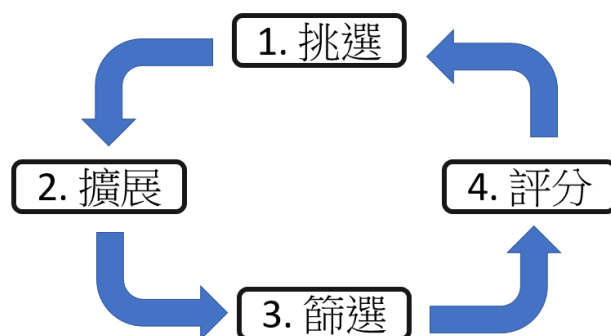
Ex: 每個老師會有自己喜好的時段
    - ii. 硬性限制：不可違反的限制

Ex: 學校固定通識課、班會的時段，系上排課時必須迴避
4. 排課機制的演算法：
  - a. 土法煉鋼、基本機制：

將課程視為球、時段視為箱子，因此可看成複數相異球丟入複數相異箱子中的排列組解法。
  - b. 評分機制：

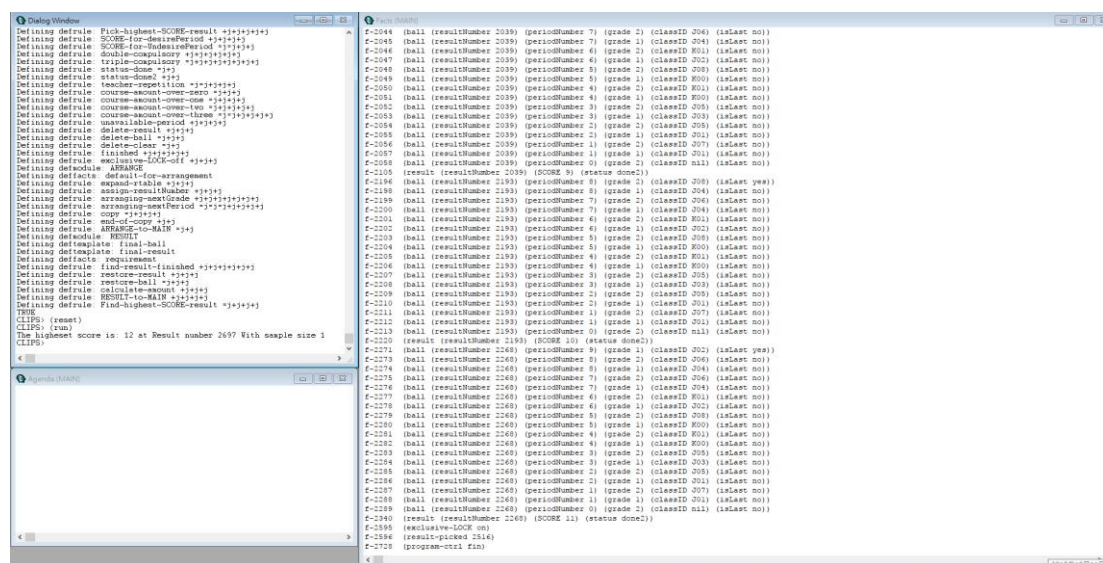
由於排列組合的結果數量過於龐大，無法在有效時間內完成排課，因此需要評分機制，為每個節點評分，最後輸出分數較高的課表結果。一方面能解決處理時間過長的問題，另一方面也能提供較符合排課規則的結果。

c. 建立搜尋樹：利用 **branch-and-bound strategy** 演算法，在進行排課時，利用「選擇分數較高者」作為 branch 條件，同時利用「違反排課規則」作為 bound 條件，建立搜尋樹找到最佳課表(即評分最高者)



### 5. 效能加速：

經過分析資料後發現，利用本系統一層一層篩選違反規則的結果，存在一個缺點：當搜尋樹運行進度越深，且結果樣本數到達一定巨大數量時，將產生大量違反規則之結果，而系統不斷進行「擴展結果→判斷違反規則→刪除結果」的動作，導致效能低落，因此採取以下解決方法：新增 branch 條件：當搜尋樹中的其中一個結果多次違反規則時，立即分支到其他結果。



圖九 排課系統運作畫面