元 智 大 學

資 訊 工 程 系 (所)

碩 士 論 文 計 畫 書

程式迴圈結構與語意之分析比較與視覺化機制

Visualization and Comparison Mechanisms of Program Loop Structure and Semantics

研究生：黃智鐸

指導教授：周志岳 博士

曾淑芬 博士

中華民國一○六年三月

# 目錄

[目錄 ii](#_Toc476253117)

[圖目錄 iii](#_Toc476253118)

[表目錄 iv](#_Toc476253119)

[摘要 v](#_Toc476253120)

[第一章 緒論 1](#_Toc476253121)

[1-1. 研究背景 1](#_Toc476253122)

[1-2. 研究動機 2](#_Toc476253123)

[1-3. 研究目的 3](#_Toc476253124)

[第二章 文獻探討 4](#_Toc476253125)

[2-1. 初學者在學習程式設計需具備能力 4](#_Toc476253126)

[2-2. 學習程式設計的困難 4](#_Toc476253127)

[2-3. 程式迴圈結構 5](#_Toc476253128)

[2-4. 靜態分析 5](#_Toc476253129)

[2-5. SIPLeS-II系統 6](#_Toc476253130)

[2-6. AnalyesC系統-利用分析程式碼來分析程式 6](#_Toc476253131)

[2-7. 程式流程結構分析與比較之視覺化機制 6](#_Toc476253132)

[第三章 系統設計 7](#_Toc476253133)

[3-1. 系統簡介 7](#_Toc476253134)

[3-2. 系統設計方法 7](#_Toc476253135)

[3-2-1. 基模介紹 10](#_Toc476253136)

[3-3. 系統架構 11](#_Toc476253137)

[3-3-1. 流程結構分析機制 12](#_Toc476253138)

[3-3-2. 宣告變數剖析機制 13](#_Toc476253139)

[3-3-3. 策略與要素分析比對機制 14](#_Toc476253140)

[3-3-4. 迴圈策略與語意比較機制 16](#_Toc476253141)

[3-3-5. 視覺化呈現機制 17](#_Toc476253142)

[第四章 評估方法 18](#_Toc476253143)

[第五章 預期成果 19](#_Toc476253144)

[參考文獻 20](#_Toc476253145)

# 圖目錄

[圖表 1 迴圈策略-計數型控制包含之資料 1](#_Toc476250235)

[圖表 2 迴圈策略-哨兵控制包含之資料 2](#_Toc476250236)

[圖表 3 基模比對 8](#_Toc476250237)

[圖表 4 比較策略與語意基模 9](#_Toc476250238)

[圖表 5範例程式 10](#_Toc476250239)

[圖表 6 整體系統架構圖流程結構分析機制 11](#_Toc476250240)

[圖表 7 流程結構分析機制 12](#_Toc476250241)

[圖表 8 宣告變數剖析機制 13](#_Toc476250242)

[圖表 9 策略與要素比對機制 15](#_Toc476250243)

[圖表 10 迴圈策略與語意比較機制 16](#_Toc476250244)

[圖表 11 視覺化呈現機制 17](#_Toc476250245)

# 表目錄

[表格 1 評估學生程式類型與結構表 17](#_Toc475756335)

# 摘要

對於初學者來說學習程式大多都不具備程式規劃、程式撰寫以及程式偵測能力，導致於學生無法找出程式問題的解題策略，或者是無法將解題策略寫成程式碼，甚至沒辦法將程式碼可能會執行的結果預測出來並且偵錯。而初學者對於迴圈的困難往往是不了解迴圈該使用何種的解題策略，對於迴圈的語法撰寫不清楚

，以及對於迴圈語法測試的困難，這些都是初學者學習寫程式很大的阻力。

現有的系統大多是對於特定的程式語言來開發或是對於特定寫法的程式來開發，利用分析程式碼來轉換相依圖來進行分析程式碼，鮮少針對初學者來開發並且利用視覺化來幫助學生直接了解目前程式碼的結構以及語意狀態。

本研究的目的在於研發一套程式「程式迴圈結構與語意之分析比較與視覺化機制」，透過分析迴圈結構和語意並且視覺化呈現，來幫助初學者學習理解迴圈，克服初學者在學習程式所遭遇到的總總困難，以幫助老師快速的能夠自動分析出學生程式碼的結構。

本研究將針對元智大學98學年度至103學年度資訊工程學系選修「程式設計(一)」知學生作業來進行評估，將以人工的方式來判斷開發的系統之正確行。並且在106程式設計(一)課程中讓學生可以實際利用每週作業所上傳作業來評估系統。

關鍵字:基模比對、迴圈結構分析、迴圈語意分析、迴圈語意比較

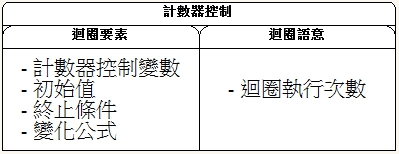
# 緒論

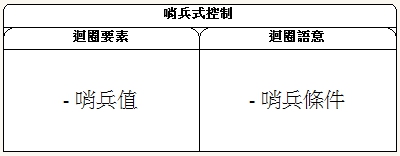
## 研究背景

學習程式語言往往需要具備程式規劃、程式撰寫、程式測試的能力 (duBoulay 1986)。對初學者來說欠缺程式規劃能力會造成學習者找不到適合解決問題的策略。欠缺程式撰寫能力則是無法將學習者找到的策略轉化成程式碼(Robins, A., Rountree, J., & Rountree, N. 2003)，欠缺程式測試能力而是學習者無法預測出程式執行的結果或者是無法找出程式碼中的錯誤 (Thompson 2006) 。

初學者在學習程式語法的迴圈，經常找不到適合的迴圈解題策略，如計數器控制迴圈和哨兵式控制迴圈，對於迴圈兩種策略使用時機不清楚，也對於語法撰寫上設定正確的起始、終止條件不了解，導致於迴圈執行的次數有錯誤，或者是不會設定哨兵式控制迴圈的哨兵值，可能造成無限迴圈或者是錯誤的跳脫條件，甚至對於迴圈的語法測試上有困難，不清楚自己本身的迴圈預期會跑的次數。

實際程式語言中迴圈的撰寫方式有三種為分別為For、While、Do While。在迴圈的結構中主要分類出兩種策略：計數器控制迴圈、哨兵式控制迴圈 如圖表(1)、圖表(2)所示。而在這兩種策略中，計數器控制型迴圈的要素主要包含: 計數器控制變數、起始值、終止值、變化公式，語意包含了迴圈執行次數。哨兵型控制迴圈的要素主要:哨兵值。語意包含了哨兵條件。



圖表 1 迴圈策略-計數型控制包含之資料

圖表 2 迴圈策略-哨兵控制包含之資料

目前學者們利用「程式碼分析程式結構」，SIPLeS-II利用將程式碼轉換成增強型物件導向程式相依圖(Xu, S., & Chee, Y. S., 1999)，來分析程式碼的語意。AnalyesC也是利用將程式碼轉換成相依圖後進行語義的分析(Li, G., Wu, W., Sun, Y., Wang, J., & Lai, T. 2007)，結構分析則是分析程式中的控制結構、函式的呼叫、資料結構相似度檢查。程式流程結構分析與比較之視覺化機制(葉時廷, 2016)，是利用Antlr所撰寫的BNF將學生的程式碼轉換成一顆剖析樹，來分析學生程式碼的流程架構。

## 研究動機

現有利用程式碼分析程式結構的研究中，SIPLeS-II只針對SmallTalk語言來做開發。AnalyesC只提出如何使用程式碼分析程式結構的概念。「程式流程結構分析與比較之視覺化機制」只針對程式流程結構以及程式流程比較的視覺化呈現

，並沒有呈現出更細部經過分析後的程式碼資訊。因此本次研究是希望藉由程式碼分析以及程式碼結構視覺化的方式分析迴圈結構與語意，來找出學習者使用迴圈的策略，並且分析出迴圈的執行次數。利用程式碼分析之後得到的資訊來比較學習者跟學習者之間或是學習者跟老師之間使用迴圈的次數以及策略，再使用程式碼視覺化(Code Visualization)的方式將迴圈的語意視覺化呈現 (Myers,B.A 1990) ，來提供更多資訊讓學習者更完整的了解迴圈使用的策。

## 研究目的

在迴圈結構中包含了迴圈解題策略與迴圈結構要素，而語意包含了迴圈執行次數與哨兵條件。本次研究目的希望研發一套「程式迴圈結構與語意之分析比較與視覺化機制」，是希望偵測迴圈結構中的要素將其建立成基模(Schema)，利用比對基模來分析學生迴圈策略以及迴圈語意，能夠利用比對後所分析之資訊幫助老師自動分析正確的學生程式碼結構，並且幫助學生快速地了解迴圈流程結構。

本次研發系統包含四種機制：

* 1. 宣告變數剖析機制

分析出宣告的變數並且提取出個變數的初始直與各變數使用之狀態。

* 1. 策略與要素分析比對機制

利用比對學生迴圈結構要素，如哨兵行控制迴圈裡的哨兵直，計數型控制迴圈的計數型控制變數、起始值、終止值、變化公式來分析出學生使用的迴圈結構的解題策略是屬於哨兵型控制迴圈或是計數型控制迴圈，並且偵測出計數型控制迴圈所執行的次數和哨兵式控制迴圈的哨兵值。

* 1. 迴圈策略與語意比較機制

利用策略與要素分析比對機制分析出的學生的迴圈策略以及迴圈語意(執行次數或哨兵值)，來跟其他學生或是老師做比較，讓學生跟老師可以知道，兩邊程式使用的迴圈策略以及迴圈語意之解題差異。

* 1. 視覺化機制

將程式中的迴圈資訊以及利用程式迴圈結構與語意分析機制所分析出的迴圈策略以及迴圈語意和程式迴圈結構與語意比較機制所比較完之差異透過視覺化機制呈現出來。

# 文獻探討

1. 初學者在學習程式設計需具備能力

初學者大多在學習程式設計都會遇到很多困難，它牽涉到很多領域的技能。而Robin等學者在(2003)研究中就提到初學者在學習程式設計以前需要先具備解題策略評估、偵錯、修改能力。像程式規劃的能力，才能規劃寫程式的方法或設計演算法。而撰寫時候更是牽涉到需要偵錯和修改的能力，像是如何預想程式應該會是如何執行，初學者必須要具備能夠預測出程式碼執行的狀況並且能夠利用預想能力來判斷程式錯誤的地方。

1. 學習程式設計的困難

在學習程式設計時會遇到的困難處，duBoulay學者在(1986)的研究論文中有提到，學生在學習程式語言的文法會出現因為不清楚程式碼指令下達以後會讓程式執行之間有什麼反應，所以對初學者來說學習設計程式以及程式偵錯與測試的能力是困難的。也會因為程式碼的抽象(Milne & Rows,2005)讓初學者很難理解程式碼的概念導致寫出來的程式碼跟原本所規劃想表達的不盡相同(Robin,Rountree,& Rountree,2003)。

初學者對於迴圈的困難處，Thompson等學者在(2006)的研究中就提出學生無法對於迴圈目前的設置看出迴圈執行狀況，問題為:他利用for迴圈使用遞增遞減的方法找出陣列中最大的值，學生無法利用現有的for迴圈所設置的因素找出答案，很顯然的學生無法利用從for迴圈的程式碼上面觀察出來迴圈實際會執行的狀況。

1. 程式迴圈結構

程式迴圈的類型主要分為三種類型For、While、Do While。For迴圈的結構主要包含了三種要素為:初始值、終止條件、變化公式。初始值主要用來計算迴圈執行次數所使用的變數。終止條件是用來判斷離開迴圈條件的欄位。變化公式主要是用來控制迴圈變數的一個欄位，他會在迴圈條件成立時並且執行完畢後，進行更新動作的欄位，目的就是要讓迴圈繼續執行直到終止條件成立。

While迴圈的結構的要素為:條件式。While迴圈執行動作主要是當While要素的條件式成立時，執行While迴圈本體，則不成立時，跳脫While迴圈。

Do While迴圈的結構跟While類似但是Do While迴圈他在進入While(條件式)判斷前會先做一次迴圈本體的敘述，也就是說他會先做一次程式碼再去判斷While迴圈成不成立，成立才繼續做迴圈本體敘述，不成立就跳脫迴圈，不再做迴圈本體敘述。

通常For迴圈大多使用的迴圈策略主要是計數型控制迴圈，因為For迴圈主要是用來處理已知執行次數資料量，用起來較為簡單，While迴圈與Do While主要則兩種策略都使用為:使用策略計數型控制迴圈和哨兵式控制迴圈，因為While跟Do While比較常在處理未知執行次數資料量，所以比較容易使用到哨兵值來控制迴圈的跳脫。

1. 靜態分析

目前的電腦輔助分析學生程式碼系統的相關研究目前大多使用到靜態分析以及動態分析。動態分析主要是利用比對程式執行後輸入與輸出的結果，來進行分析出程式的正確性與否，只能看結果不能看過程。而靜態分析的好處就是能夠針對程式碼的結構進行分析與評估，利用分析程式碼的分析與評估可以運用在比較學生跟學生之間的程式碼架構，藉由此方法可以知道學生之間程式碼是否有抄襲。

1. SIPLeS-II系統

國立新加坡大學電腦學院的SONG-WEN,XU等學者(1999)研製出來的系統主要用來分析學生程式，並且用來查找其中錯誤並提供錯誤提示，從而提供了一個更為智慧化的程式設計學習環境。其系統採用了一種技術名為物件導向程是依賴關係圖(AOPDG)的結構。方法是利用把SmallTalk語言的程式碼轉換成另一種中介語言的結構後，並且利用判斷相依圖來進行診斷與評分，好處是能夠更嚴謹的偵測出程式的錯誤與容易實踐，但此系統只針對SmallTalk程式語言做開發，不適合學習C++語言的初學者。

1. AnalyesC系統-利用分析程式碼來分析程式

廣東工業大學Wu, W., Li等學者(2007)研發的系統－AnalyseC，該系統主要分為兩種層次進行程式分析為:結構與語意層次。結構層次分析主要從程式度量的角度進行，透過分析程式中的控制結構、函式呼叫和使用的資料結構，檢查程式之間的結構相似性。語意層次分析主要使用編譯器相關的技術，對程式進行保留語義的標準化轉換，得到可以進行比較的程式語義表現形式－程式依賴圖進行程式评价分析。

1. 程式流程結構分析與比較之視覺化機制

元智大學資訊工程系葉時廷(2016)研發的系統，該系統主要是使用ANTLR來撰寫BNF規則，利用ANTLR剖析學生程式碼，並且轉換成BNF規則的剖析樹，藉由剖析樹得知學生所撰寫的程式流程架構，將剖析出來的程式流程架構畫出程式流程圖。也可以利用程式碼比較機制來比較學生跟學生之間或者是學生跟老師之間的程式流成差異，並且將學生程式碼流程結構和利用程式碼比較機制比較後的流程差異利用視覺化呈現出來。

# 系統設計

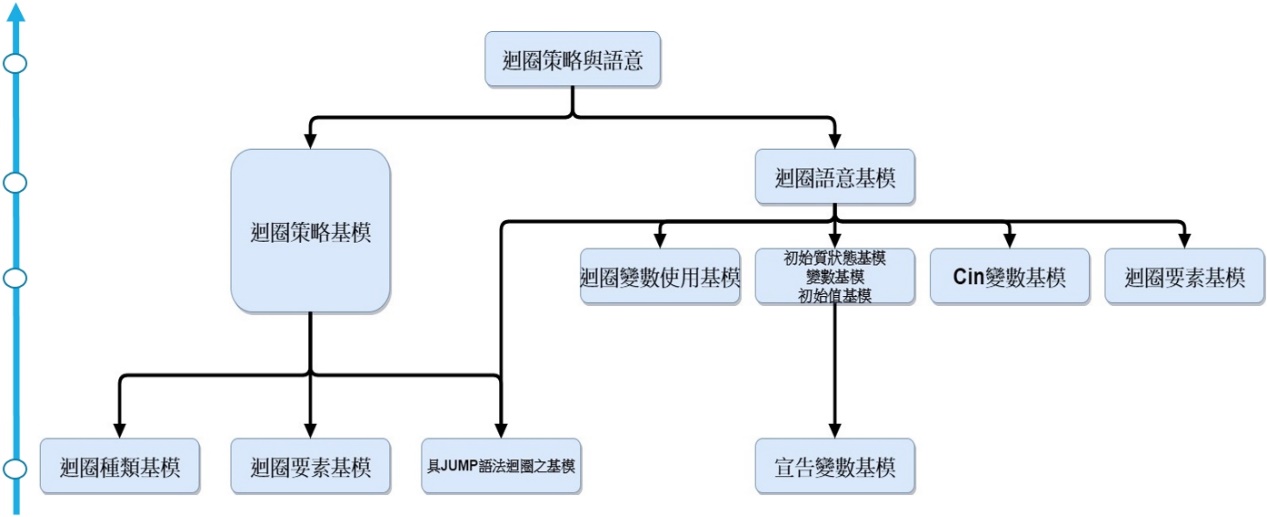
1. 系統簡介

本次系統研發主要研發「程式迴圈結構與語意之分析比較與視覺化機制」，主要是使用ANTLR所剖析出的BNF規則剖析樹所剖析的迴圈程式碼語意資料，將資料利用迴圈語意分析機制來分析迴圈的使用策略、迴圈的執行次數，並且使用分析出來的策略與語意來和學生之間或是和老師之間比較，藉由比較了解不同策略的程式碼之間的差異，最後將結果以視覺化方式呈現給學生及教師。

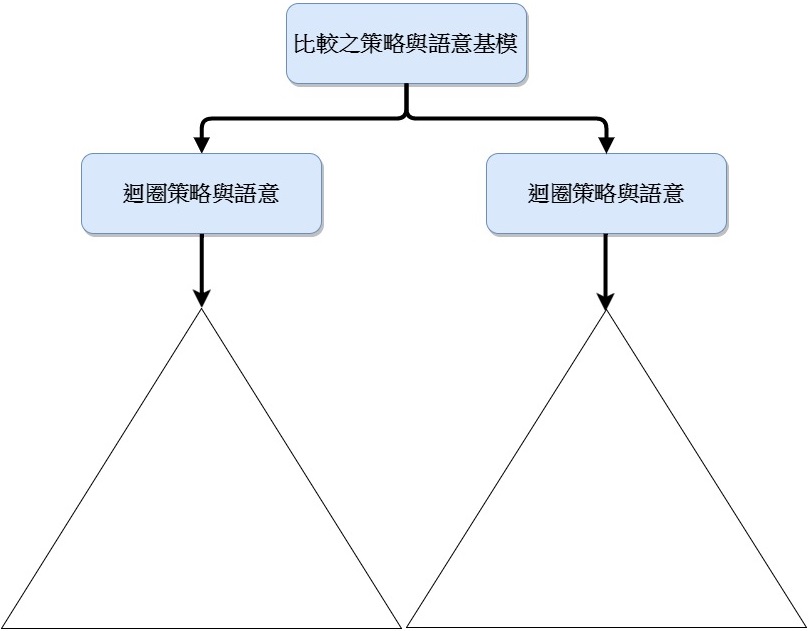
1. 系統設計方法

本研究之系統是利用建立基模(Schema)進行基模比對(Schema Matching)來分析迴圈策略和迴圈語意。主要建立了八種基模:迴圈種類基模、迴圈要素基模、宣告變數基模(其中包含始值狀態基模、變數基模、初始值基模)、Cin變數基模(其中包含Cin變數使用狀態基模)、迴圈策略基模、具哨兵值迴圈基模、迴圈語意基模。藉由同時比對三種基模資訊:迴圈種類基模、迴圈要素基模和具JUMP語法迴圈之基模分析迴圈策略，來產生出迴圈策略基模。比對五種基模資訊:宣告變數基模、Cin變數基模、迴圈要素基模、迴圈變數使用基模、具哨兵值迴圈基模分析迴圈語意並且產生出迴圈語意基模。

其中系統的比對方法是採用Bottom-Up方法來建立基模進行比對，先將迴圈策略以及迴圈語意的所需分析的基本資訊之基模建立出來，藉由最底層資訊來往上分析以及建立上層資訊，由下往上最後得到整個所要分析的迴圈策略與語意，如圖表(3)所示。並將單一程式比對出來所產生的迴圈策略與語意基模與另一個單一程式比對所產生的迴圈策略與語意進行比較，如圖表(4)所示。



圖表 3 基模比對

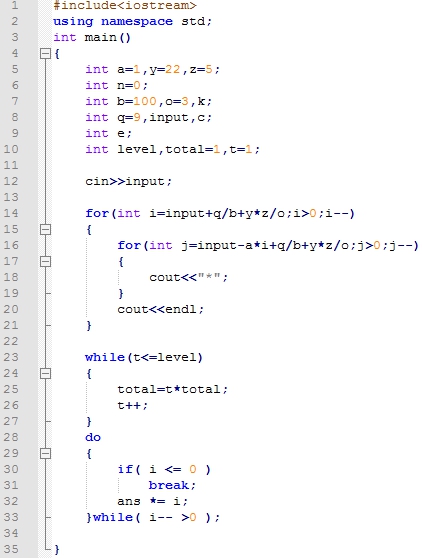


基模比對

基模比對

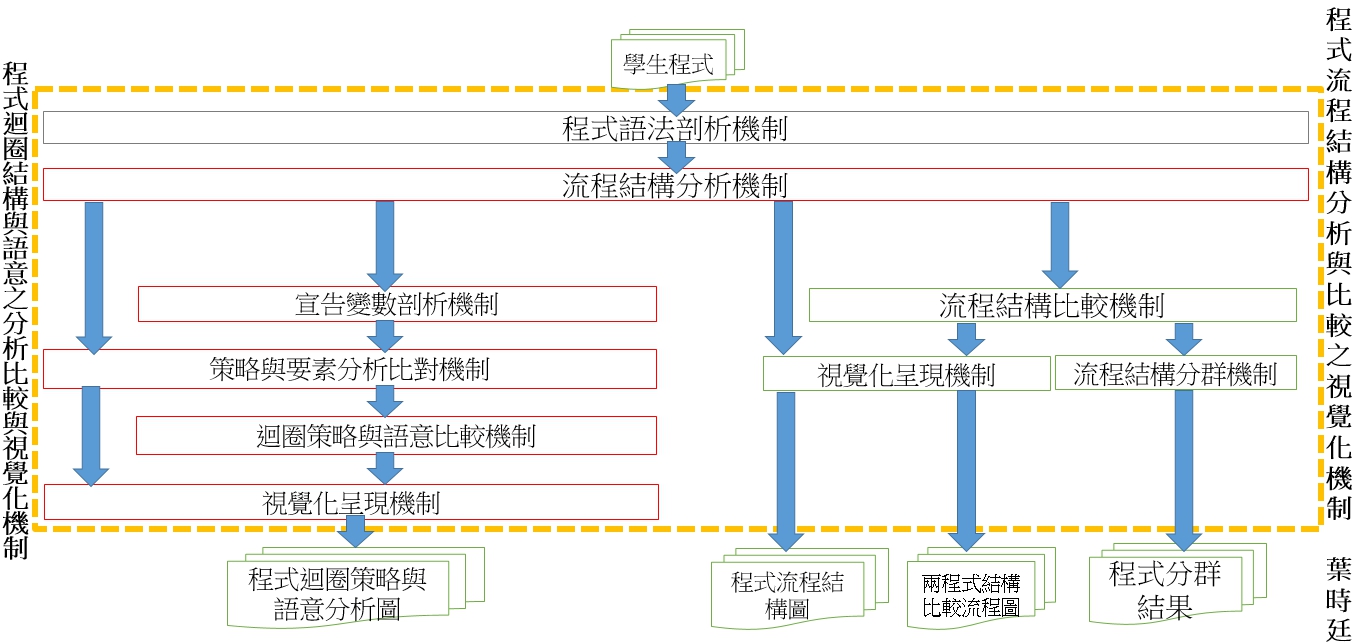
圖表 4 比較策略與語意基模

### 基模介紹



圖表 5範例程式

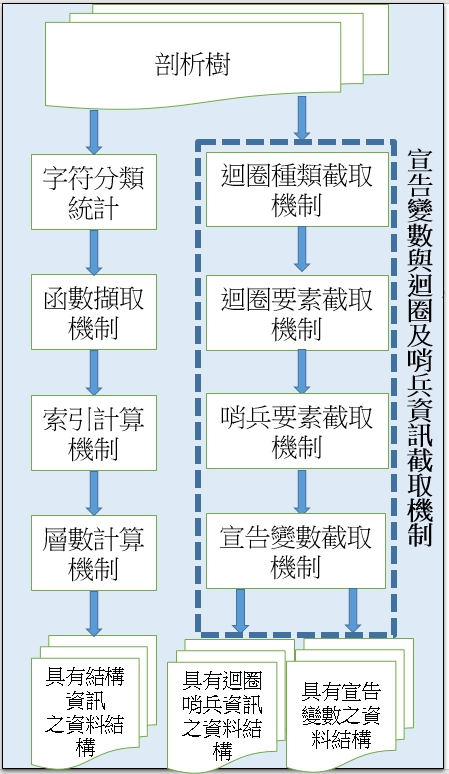
如圖表(5)所示之範例程式中總共使用三種迴圈撰寫方法分別為第十四行、第六行的For迴圈，第二十三行的While迴圈以及第二十八行到第三十三行的Do-While迴圈。上述迴圈撰寫方式可以分別將他的迴圈種類以及迴圈要素建立成兩種基模為迴圈種類基模以及迴圈要素基模，而在迴圈要素基模For迴圈主要記下了三種要素分別為起始、終止、變化公式。While迴圈記了一種要素為條件運算式。Do While迴圈記了本體、條件運算式。具JUMP語法迴圈之基模主要是偵測出迴圈裡面的JUMP語法，作為哨兵策略判斷條件之基模。第五行到第十行的變數宣告範圍主要是將他的宣告變數建立成宣告變數基模，並且利用將其切割分類建立成另外三種附屬的基模為:初始值狀態基模、變數基模、初始值基模。第十二行的Cin可以將其使用的變數建成Cin變數基模，並且比對宣告變數基模來產生出Cin變數使用狀態基模。迴圈變數使用基模主要是將迴圈要素切割，將其要素中的變數提取出來並且利用比對宣告變數來建立的一個屬於迴圈變數狀態類型的基模，其目的是要用來比對上層的語意所需要的基模。迴圈策略基模和迴圈語意基模屬於系統的比對最上層產生之基模，主要是記錄著分析迴圈結構後產生出的策略以及分析語意為分析後之最終結果。

1. 系統架構

圖表 6 整體系統架構圖流程結構分析機制

### 流程結構分析機制

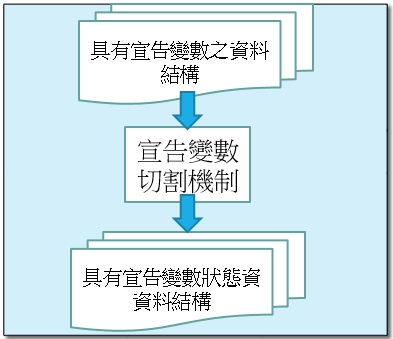
在「程式迴圈結構與語意之分析比較與視覺化機制」(葉時廷 2016)，流程結構分析機制是將剖析樹所剖析出來的程式結構對字符分類、函式擷取、索引計算以及層數計算，這些主要是在處理程式碼的整理架構。本研究中的流程結構分析機制新增了宣告變數與迴圈及哨兵資訊截取機制，主要是將剖析樹所剖析的程式碼迴圈部分，截取迴圈種類、迴圈要素、哨兵要素、以及宣告變數和Cin變數，利用四種的擷取機制來截取上述的資料，並且將資料建立成基模(Schema)為五種基模:迴圈種類基模、迴圈要素基模、具JUMP語法迴圈之基模、以及宣告變數基模和Cin變數基模。



圖表 7 流程結構分析機制

### 宣告變數剖析機制

利用宣告變數切割機制將上一個程序所產生的具有宣告變數之資料結構進行切割，目的是將宣告變數由完整的語句如:int a = 10;的結構拆解出來，可以偵測出是否有初始值之狀態:True，以及拆解出變數:a，初始值為10的資訊並且將這些建立成三種基模為:初始值狀態基模、變數基模、初始值基模。



圖表 8 宣告變數剖析機制

### 策略與要素分析比對機制

1. 迴圈分類機制

在「程式迴圈結構與語意之分析比較與視覺化機制」(葉時廷 2016)中的程式語法剖析機制中將迴圈主要三種撰寫方式的For分類成為兩種為For\_declare和For\_initial這兩種差別主要是差別在For\_declare是將控制的變數宣告成區域的變數，For\_initial是將控制的變數宣告成為全域變數。而迴圈分類機制是將不同的迴圈撰寫方式分類出來並且將資訊傳入策略分析機制。

1. 策略分析機制

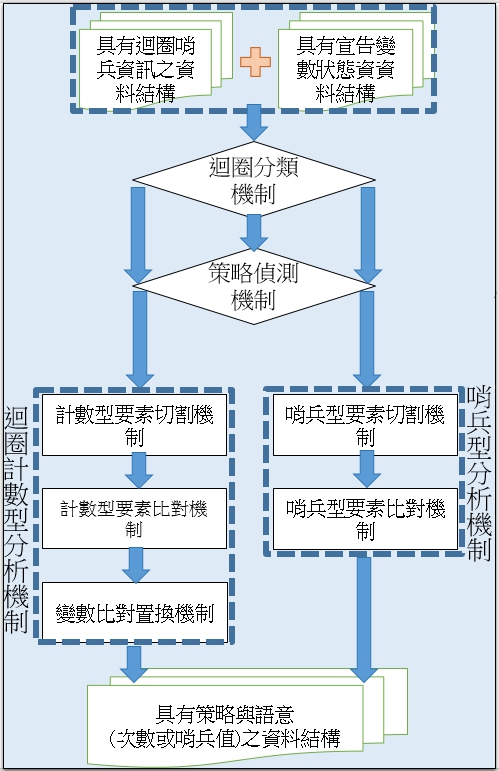
主要分析不同迴圈的策略，首先利用迴圈要素基模和迴圈種類基模比對具JUMP語法迴圈之基模和Cin變數基模，來檢察迴圈要素和迴圈種類基模是不是有JUMP指令或是Cin控制跳出迴圈的存在，藉由比對這兩種基模來更新迴圈變數使用狀態基模和判斷迴圈策略是屬於哪一種，來建立迴圈策略基模以及迴圈變數使用狀態基模。

1. 迴圈計數型分析機制

包含了三種小機制:計數型要素切割機制、計數型要素比對機制、變數比對置換機制。計數型要素切割機制，主要目是將要素切割成運算符號和變數，其用意是利用運算符號如:<、>、<=、>=、++、--，來判斷變數的變化方式，藉由得之變化方式來進行計數型要素比對機制，得之變化方式之後，利用變數比對宣告變數基模、Cin變數基模、變數使用狀態基模來判斷變數的狀況與資料，將迴圈的次數分析出來，進行變數置換來置換以初始的變數，並且重組切割要素的語句，來分析迴圈次數。

1. 哨兵型分析機制

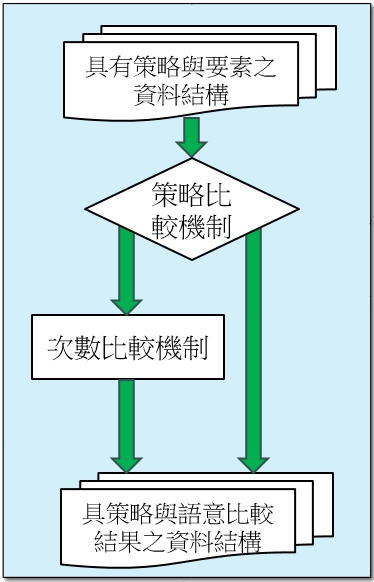
切割迴圈要素切割後將要素切割成變數、變數型態、比較運算符號(!=、==、<=、>=)。並且比對具有JUMP語法迴圈之基模和Cin變數基模，來判斷哨兵策略是使用Cin控制或是使用JUMP指令控制跳脫。並且在比對完後後建立一個迴圈策略基模。



圖表 9 策略與要素比對機制

### 迴圈策略與語意比較機制

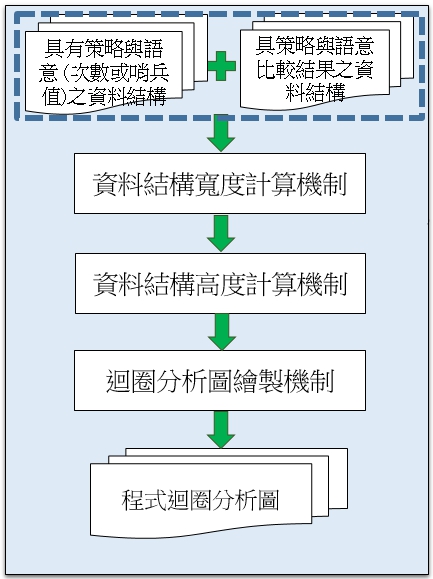
在迴圈策略與語意比較中主要分為兩種機制:為策略比較機制、次數比較機制，其功能比對兩個不一樣程式所分析後建立的迴圈策略基模與迴圈語意基模。先比對策略同樣策略才比語意，藉由比對的方法可以分析出兩種程式使用迴圈的策略，與使用策略之語意之間的差別。



圖表 10 迴圈策略與語意比較機制

### 視覺化呈現機制

此機制是將策略與語意跟策略與語意比較的資料進行視覺化，進而可以計算出資料其資料結構的寬度、高度並且利用jGraphx開源套件將資料利用繪製的圖片呈現出來。



圖表 11 視覺化呈現機制

# 評估方法

本次研究是以元智大學資工系98學年度至103學年度的程式設計(一)課程學生作業評估「程式迴圈結構與語意之分析比較與視覺化機制」，以及105學年度程式設計(一)課程每周作業評估。依目前已收集四種類型之作業，為星號三角形156個、閏年146、階層463、質數143個、共908個學生程式，來評估程式是否可以正確的偵測出迴圈策略與迴圈語意。藉由本系統能夠評估出程式迴圈策略正確、計數型迴圈之次數正確在轉由人工判斷策略與計數型次數是否一致，並且統計其正確率。並且利用問卷評估「程式迴圈結構與語意之分析比較與視覺化機制」，對於105學年度程式設計(一)課程之學生幫助性。

表格 1 評估學生程式類型與結構表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 迴圈 | | | 分支 |
| 評估程式 | FOR | WHILE | DO WHILE | IF |
| 星號三角形 | ● |  |  |  |
| 閏年判斷 |  |  |  | ● |
| 階層計算 | ● | ● | ● |  |
| 質數判斷 | ● |  |  |  |

# 預期成果

本研究開發目標於發明一套「程式迴圈結構與語意之分析比較與視覺化機制」，其功能能夠自動化的分析學生的程式迴圈結構，將學生迴圈結構分析出來並且找出學生迴圈的策略與迴圈語意，並且可以比較任意兩份同類型不同作者寫的程式碼，將其策略比對出來，自動分析出他使用的策略與語意找到寫出正確寫法跟錯誤寫法之間策略與語意之差別。並且利用視覺化將分析結果利用圖片顯示與系統的右側幫助使用者理解撰寫之程式邏輯以及理解迴圈的抽象概念，讓使用者能夠清楚的知道當前撰寫程式的邏輯是否跟規劃的有所一樣。

# 參考文獻

1. 李广强. (2007). 基于转换的程序分析技术的研究与应用，广东工业大学碩士論文.
2. 葉時廷.(2016). 程式流程結構分析與比較之視覺化機制，元智大學資訊工程研究所碩士論文.
3. Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73.
4. Li, G., Wu, W., Sun, Y., Wang, J., & Lai, T. (2007, February). Transformation-based assessment for C programs. In *Signal Processing and Its Applications*, 2007. ISSPA 2007. 9th International Symposium on (pp. 1-4). IEEE.
5. Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7(1), 55-66.
6. Myers, B. (1980). Displaying data structures for interactive debugging.
7. Myers, B. A. (1990). Taxonomies of visual programming and program visualization. *Journal of Visual Languages & Computing*, 1(1), 97-123.
8. Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
9. Thompson, E., Whalley, J., Lister, R. & Simon, B. (2006).Code classification as a learning and assessment exercise for novice programmers. In *NACCQ 2006: Proceedings of the* 〖19〗^th *Annual Conference of the National Advisory Committee on computing Qualifications*, pp. 291-298.
10. Xu, S., & Chee, Y. S. (1999). SIPLeS-II: An Automatic Program Diagnoses System for Programming Learning Environments. *Artificial Intelligence in Education–Open Learning Environments: New Technologies to Support Learning Exploration, and Collaboration*, 397-404.