

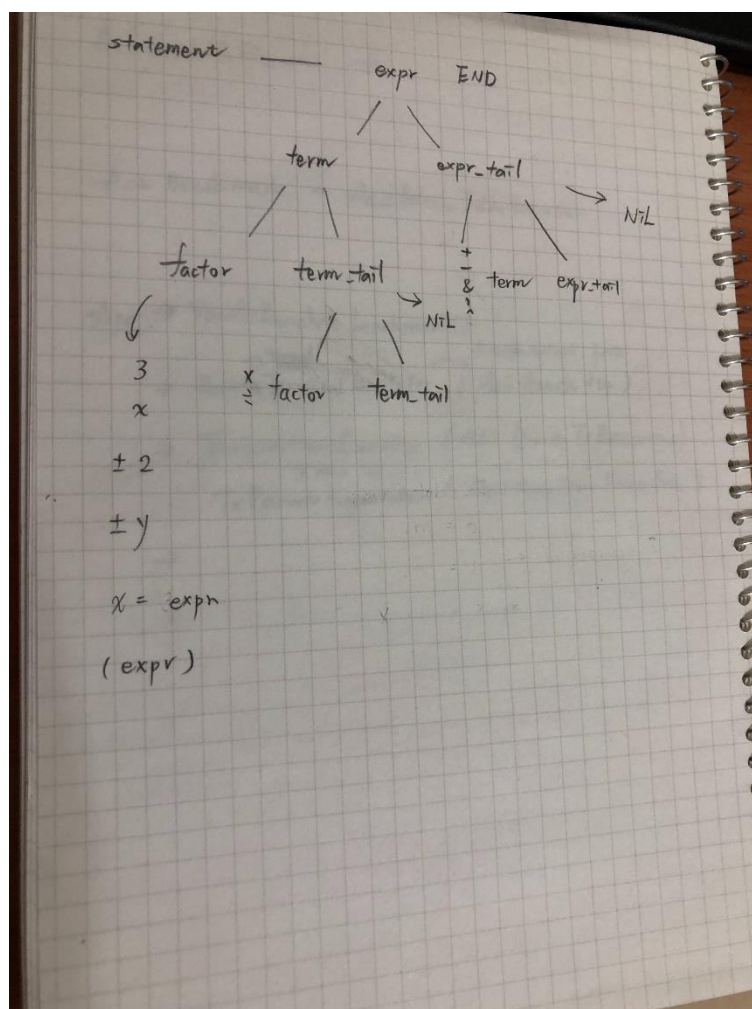
Report

電資 21 106060004 曹家誠

1. 設計思路:

先 trace code，大致上可以分成兩部分，第一部分是建好 syntax tree，第二部分是依據建好的 tree，去 print Assembly code。

第一部分中，大部分內容都已經被完成了，我花了一些時間在研究 statement 等等關係，大致上如下圖。



接下來就是補齊尚未完成的部分，大致上剩下 ORANDXOR 的部分，不會很難只要參考其他 ADDSUB 等就可以完成。

也可以補齊 evaluateTree 來檢查一下結果，到這邊大致上 Tree 已經建構完成。

第二部分才是最花時間的部分，首先先檢查不合法的部分，如果是不符合沒辦法建構成 **Tree** 的狀況，再上一部份建構時就會 **Error**，所以已經 **ok** 了。

而接下來就是要檢查 **Tree** 是正確但是仍然不合法的，像是右邊直接出現沒出現過的變數、或是沒有 '=' 等等(注：直接出現沒出現過的變數會再 **getVal** 創建，所以應該會在那邊拋出錯誤)，透過 **isValid** 檢查，是不是 **Var = XXXX** 這樣的結構，並且透過 **checkTree**，**recursive** 去檢查會不會有多個等於等等狀況。

接下來針對加速的部分，添加 **reconstructTree function**，**recursive** 去將常數之間的運算先做，就不用到時候還要透過 **MOV** 做這些運算。

最後就是將完成的 **Tree Print** 出來成 **Assembly code** 的形式，傳入 **root** 以及 **regNum**，**regNum** 從 3 開始，怕會改到 **xyz** 的部分，其中 **ORANDXOR**、**MULDIV**、**ADDSUB** 都比較簡單，不會有什麼例外狀況，只要 **recursive call** 左右半邊 **child** 就好，最後將兩者依相對的運算子做輸出，而 **ID** 以及 **ASSIGN** 會比較需要處理。

因為這兩者都會有變數，也就是要涉及 **memory(table)** 的操作，所以會比較複雜。**ID** 的部分就是先檢查變數是否再 **memory**，如果是就必須從 **memory** 取，否則可以從 **regsiter** 拿就好(依據是否為前三個 **xyz** 有所不同，**xyz** 就可以從 **r0, r1, r2** 拿)。**ASSIGN** 的部分就是最後看是否為 **xyz**，再看要放到 **memory** 或是 **r0, r1, r2**。

2. 過程中遇到的困難:

首先是 **trace code** 有點麻煩，變數以及 **function** 都沒有 **comment** 要幹嘛的，有的變數名稱也猜不出是拿來幹嘛的，所以 **trace** 了蠻久的。

處理 **Tree** 的部分不難，但是會在 **printAssembly** 時遇到麻煩，不過其實大致上有點像是多試試幾種排列組合，再透過觀察程式表現修改，尤其是 **ID** 以及 **ASSIGN** 的部分，牽涉 **memory** 會有點難控制。

最後就是忘記 **print EXIT 1...**，一直以為什麼沒考慮到。

3. 想對老師 or 助教說的話:

以後 **variable** 以及 **function** 希望都要有 **comment** 看這是在幹嘛的 **function**，可能會比較好 **trace XD**。