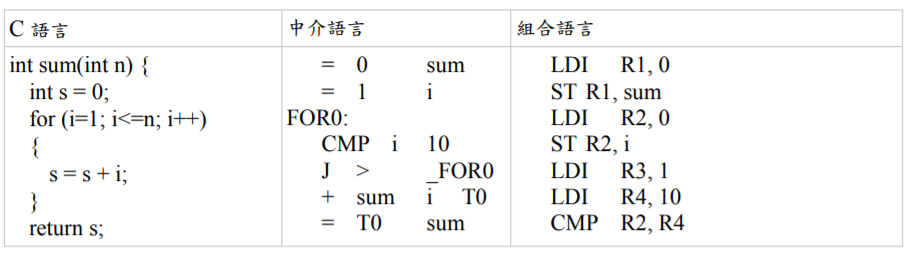
WEEK12

### **自然語言**

1. 自然語言並非刻意「設計出來的」。例如中文、英文。是歷史逐漸演變的。
2. 人造語言由某個人從無到有所創造設計出來的，像是「C 語言、Python、JavaScript、Ruby、Perl 」這種語言通常有非常明確固定的語法，我們可以透過電腦程式去「解釋」這些語言的語法，然後做出對應的動作。
3. 標記語言，像是「HTML、XML、維基語言、Markdown、ReStructuredText」等，這類的語言乃是在自然語言上進行一些標記，以便讓程式可以透過標記進行對應的動作，像是加上粗體、加上超連結、顯示成表格、或者單純只是標記某個區塊的特性，像是 XML 的標記就是如此。

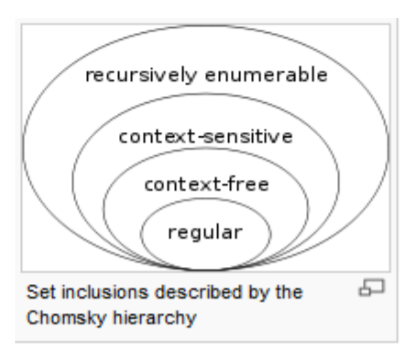
### **人造語言**

人造語言的種類很多，但大部分都是程式類的語言，像是高階語言 (像是 C、 Ruby、Python)、組合語言 (像是 x86、ARM、CPU0 的組合語言)、還有高階語言在 翻譯成組合語言之前通常會經過某種中介語言等等，以下是一些人造語言的範 例。



## **Chomsky Hierarchy (喬姆斯基語言階層）**

* Type1 語言的語法有點限制，因為每個規則的左邊至少要有一個非終端項目 A，但其前後可以連接任意規則，這種語法所能描述的語言稱為「對上下文敏感的語言」 (Context-Sensitive)，因為 可以決定之後到底是否要接 ，所以前後文之間是有關係的，因此才叫做「對上下文敏感的語言」。這種語言在計算理論上可以對應到「線性有界的非決定性圖靈機」，也就是一台「記憶體有限的電腦」。
* Type2 語言的語法限制更大，因為規則左邊只能有一個非終端項目 (以 A 代表)，規則右邊則沒有限制這種語言被稱為「上下文無關的語言」(Context Free) ，在計算理論上可以對應到 「非決定性的堆疊機」(non-deterministic pushdown automaton)。
* Type3 的語法限制是最多的，其規則的左右兩邊都最多只能有一個非終端項目 (以 A, B 表示) ，而且右端的終端項目 (以 a 表示) 只能放在非終端項目 B 的前面。這種語言稱為「正規式」(Regular)，可以用程式設計中常用的「正規表達式」(Regular Expression) 表示，對應到計算理論中的有限狀態機(Finite State Automaton)。



參考資料

<https://zh.wikipedia.org/zh-tw/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80>

<https://blog.csdn.net/Tiger_DAHE/article/details/111871450>

<https://medium.com/%E4%BA%BA%E5%B7%A5%E6%99%BA%E6%85%A7/%E8%AA%9E%E6%B3%95%E7%90%86%E8%AB%96-23bc87126e6>