How When Where Why What Break?

Block Diagrams Constructor + Editor + Simulator

R11921126 黃允誠

What is the problem?

Block Diagrams 比較簡單，但不能處理 modules 間 dependent 的關係，而且如果是非純串並聯的系統，分析計算很複雜。Markov Chain 有更強的表達能力，但圖上的 node 數量成指數暴漲，而且計算更複雜。
各種分析方法都有優劣處，但共通點就是：計算很複雜。
其實我們可以換個思路：
我們的最終目的如果只是知道系統的極限、可靠性、哪邊容易壞、該如何優化架構等答案，我們當然可以坐下來算一大堆的數學，算到整個答案都算出來為止。但我們也可以直接簡單暴力地用同樣的系統進行大量的實驗收集數據，只要實驗夠大量，就可以得到逼近理論值的結果；甚至可以做到比上下界估算這類方法還精準。
問題是我們要怎麼拿到一堆一模一樣的系統？實驗是否會花費不可接受的成本？沒關係，我們只要用程式模擬實驗就好。
首先對於最基礎的元件，例如電晶體、電池、開關等等，當然還是要進行最基礎的實體分析。接著我們就針對以這些基礎元件構成的底層系統元件，如加法器、邏輯閘等等，造出理論上的架構圖後，以程式短時間內直接模擬上億、上兆次生命週期的運作過程取得這些元件的相關數值。最後一層一層往上重複這個過程，我們理論上能以程式模擬得到任意系統的可靠性、瓶頸元件等等分析結果。
所以這個模擬程式只需要做三件事：第一，支援 Block Diagrams 的設計、構建及編輯；第二，支援各種已知參數的設定，例如每個元件的單位時間損壞率、元件之間的交互作用或影響、維修人員的維修效率等等；第三，直接模擬整個程式運作過程中各元件的狀態，並重複模擬後統整實驗結果。

Why is it important?

雖然使用 Block Diagrams，但是這個 approach 是直接進行模擬，理論上程式可以輕鬆設定任何系統相關的特徵，包含元件之間的交互影響、隨時間改變的損壞率和維修率等等，甚至都不需要是一個單一數學式的 function，而可以是任意的規則：例如早中晚各一個不同的常數維修率、A 元件壞掉後 B 元件損壞率 1.5 倍、甚至是線路本身會影響連接的元件等等。以上這些內容當然總是會有辦法用數學計算、統計分析算出來，問題在於分析的複雜度、難度跟成本。如果用這個 approach 就是「以不變應萬變」，理論上任何的分析需求只要開發對應的模擬程式就可以直接統計出結果。

雖然一學期的時間有限，我不一定能完成到甚麼程度，但重點是這個方法的核心蓋念；以及理論上能達到的能力上限：基本上不論是遇到什麼樣的系統可靠性分析問題，但凡你找不到任何一個合適的分析方式，你總是可以用這篇 Proposal 提出的方法作為最終手段得到一個精準度隨程式模擬次數提升的近似結果。
隨著科技進步，人們也應用科技造出愈加複雜的系統。但由於這個方法只模擬系統運作過程中各部分的狀態，它的複雜度遠低於甚至幾乎無關於實際上的系統本身。這個方案在全面 general purpose 的模擬器開發完成之後，就會變成有超高彈性；逼近 100%表達力覆蓋所有可能問題；分析成本又很低；程序還標準化的最終手段，可以視為任意系統總體可靠度分析的 baseline 方案。

What are the most related works?

痾……我不知道……這只是忽然出現在腦中的 idea。反正是分析系統總體可靠度的一種方式……

What's your hypothesis so far?

如前面所述，由於開發時間有限，可能沒辦法做到非常高階的一些細節，我應該會先開發一個最基礎的概念演示 prototype，然後還有時間再慢慢往上加，看最後能做到甚麼程度。當然還是有一些期望的，希望至少能做到處理某一種元件之間 dependent 的問題。
不過如果是說理論上的話，我目前認為這個系統幾乎沒有上限，應該可以一直優化到覆蓋 100%系統架構可能性為止，只是開發成本的問題。

How are/will you evaluate it?

我覺得就盡量做吧，看能寫到甚麼程度，這個實作部分其實就只是花時間開發程式的問題而已。

References

只有課程講課內容跟課程講義，idea 是我自己想的。