正規描述與自動驗證

Formal Description & Automated Verification

王　凡

國立台灣大學 電機工程系

產業升級壓力下的一代

- 大前研一（未來分析家）：*「台灣…優勢只剩下五年。*
- 杜書伍（聯強國際…優勢
- 歐陽明…大陸的數位內…，……

台灣未來產業的優勢在哪裡？
各位五年後的競爭力在哪裡？

Verification（驗證）?

- 找出系統設計中的所有錯誤。
- 確認系統中已經（接近）沒有錯誤。

非常困難！

複雜系統的決勝關鍵！

各位同學的一條生路！

台灣產業的一條生路！

簡介

- 瞭解電腦系統的**formal semantics**
- 學習電腦輔助驗證的理論與製作

瞭解電腦系統的formal semantics

```
divide (a, b) {
  while (a > 0)
    a = a-b;
  if (a == 0) return 1;
  else return 0;
}
```

Is this program
co
How do I know
It checks if a is
divisibl
I doubt it!
What
Well,
sometimes
happens!

瞭解電腦系統的formal semantics

```
divide (a, b) {
  while (a > 0)
    a = a-b;
  if (a == 0) return 1;
  else return 0;
}
```

Seriously, what does “a=a-b;” means ?

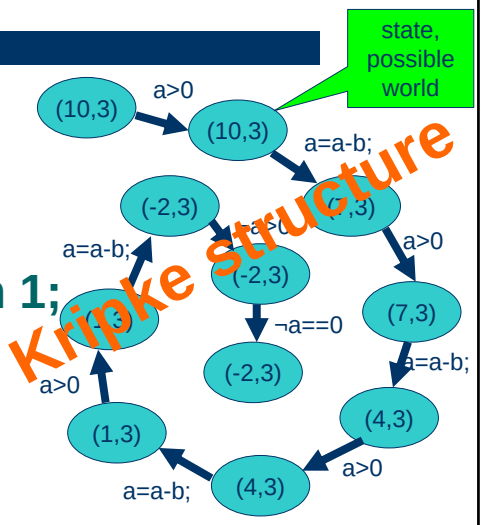
What does this `if’ statement means ?

瞭解電腦系統的formal semantics

- When we say a program is correct, what is the behavior model of the program ?
- What is the mathematics of program behaviors ?

瞭解電腦系統的formal semantics

```
divide (a, b) {
  while (a > 0)
    a = a-b;
  if (a == 0) return 1;
  else return 0;
}
```



瞭解電腦系統的formal semantics - an attempt

①divide (a, b) { p(k) ... state k;
② while (a > 0) ... state k;
③ a = a-b; ... k;
④ if (a == ...) re...
⑤ else ret...
⑥}

First-order arithmetic!
A lot of tools can help you predict the behaviors of the program.

∀k≥0(p(k) ... a(k+1)==a(k)-b(k))

學習電腦輔助驗證的理論與製作

Goedel's incompleteness theorem:

- 任何有限規則系統，都有一個無法證明的事實。

State-space explosion problem ?

- When a and b are both 32 bits long, # states
  $2^{32} \times 2^{32}$
- The safety analysis problem of Boolean program is PSPACE-complete.
- The satisfiability problem of LTL is PSPACE-complete.
- The satisfiability problem of 1st-order logics is undecidable!
  - No algorithm exists!
- The safety analysis problem of algorithm is undecidable!

Things to learn in the course

- Mathematical models of computer systems
  - Only with mathematical models, you can build EDA tools.
- Their expressiveness
- Their computation powers and complexities
- Practical techniques to overcome the complexity!

課程規劃：

- 建立基本的電腦輔助驗證的知識。
  - 前面數節課，教授基礎知識。
- 互動教學，聊解最新動態。
  - 依同學人數，排定論文研讀報告。
- 三個學期計畫，深入探討技術問題。
  - 研究問題探討、實驗系統製作。
  - 結案報告以頭影片準備，要顯示適當的努力。
  - 三人一組。每次期中報告，每組準備投影片報告。

教學進度規畫：

1、2/27：課程簡介、問卷調查
2、3/6 ：propositoinal logic 與BDD技術
3、3/13：第一次學期計畫討論（Sudoko程式）
4、3/20：自動驗證/分析模型的建立與討論
5、3/27：第一次學期計畫期中報告
古典時態邏輯與自動機理論的定義
6、4/3 ：真時自動機理論與模型建立
7、4/10：繳交第一次學期計畫
第二次學期計畫討論（運算的precondition）
混和自動機與各種無限狀態空間系統的模型
8、4/17：期中考
9、4/24：表達能力與計算複雜度、第二次學期計畫期中報告

教學進度規畫（續）：

10、5/1 ：符號式自動驗證技術、Bounded Model-Checking
11、5/8 ：繳交第二次學期計畫
第三次學期計畫討論（safety analyzer的製作）
複雜度的管理、組合式驗證、on-the-fly技術
12、5/15：無限狀態系統的自動驗證技術
SOC 自動驗證技術
13、5/22：C程式的自動驗證技術、第三次學期計畫期中報告
14、5/29：論文閱讀報告
15、6/5 ：論文閱讀報告
16、6/12：論文閱讀報告、繳交第三次學期計畫
17、6/26：期末考

課程網頁

http://cc.ee.ntu.edu.tw/~farn/courses/FMV/

參考資料：

- Handbook of Logic in Computer Science: Vol. 1-2, edited by S. Abramsky (1993), Oxford.
- *Handbook of Theoretical Computer Science*, Vol. A & B, edited by J. van Leeuwen, Elsevier.
- Model Checking, E. Clarke, O. Grumberg, D. Peled, MIT Press
- Formal Methods for Real-Time Systems
  edited by C. Heitmeyer, D. Mandrioli, Wiley
- 重要論文