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[illegible]優勢只剩下五年。
- 杜書伍（聯強國[illegible]優勢
- 歐陽明[illegible]大陸的數位內[illegible]，……

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。

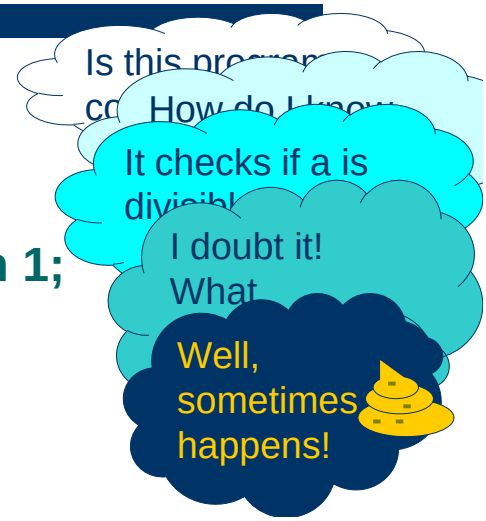
非常困難！
複雜系統的決勝關鍵！
各位同學的一條生路！
台灣產業的一條生路！

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;
}
```



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 ?

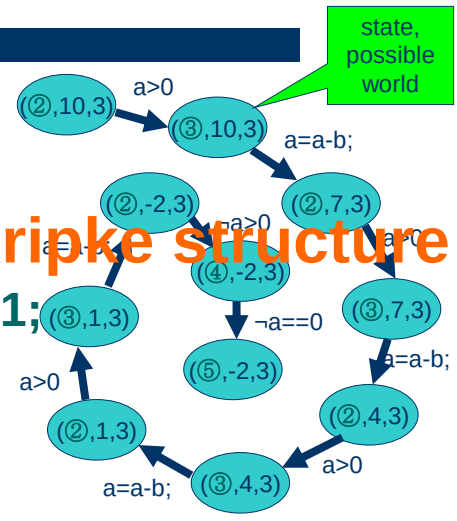
What does this `if’ statement means ?

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

①divide (a, b) {
② while (a > 0)
③ a = a-b;
④ if (a == 0) return 1;
⑤ else return 0;
⑥}



Kripke structure

瞭解電腦系統的formal semantics - an attempt

①divide (a, b) {
② while (a > 0)
③ a = a-b;
④ if (a == 0) 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))

Some incomputable problems (1/2)

- The validity of 1st-order logic formula (Hilbert's 2nd problem)

  $\exists x \forall y \exists z(\text{single}(x) \wedge (\text{parent}(y,x) \rightarrow \text{worried}(y))$
- Mortal matrix problem

  Given 15 3×3 matrices, M_1, …, M_{15}, are there $M_{i1} \times \ldots \times M_{im} = 0$?
- CFL ambiguity problem

Some incomputable problems (2/2)

- Multivariable polynomial equations (Hilbert's 10th problem)
  Incomputable for degree ≥4.
  Incomputable for 36 real variables.
  Incomputable for 11 integer variables.

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

- State-transition models of computer systems
  - Only with mathematical models, you can build EDA tools.
- Mathematical model construction
- Verification algorithms
- Practical techniques to overcome the complexity!

Things to learn in the course

State-transition models of computer systems

- Kripke structures

Things to learn in the course

Mathematical model construction

- With REDLIB packages
- for automata with dense-time clocks

Things to learn in the course

Verification algorithms

- BDD manipulation algorithm for propositional logics
- Automata (regular expression) learning
- Linear temporal logic satisfiability checking
- Automata safety and liveneness analysis
- CTL model checking
- Automata simulation checking

Things to learn in the course

Practical techniques to overcome the complexity!

- BDD-based techniques

Course plan：

- Basic understanding of the knowledge of computer verification
- Three projects
  - use REDLIB to solve board games
  - use REDLIB to construct system model and making verification for untimed systems
  - use REDLIB to do model-based testing for timed systems

Course schedule

1. 9/15 Introduction
2. 9/22 中秋節
3. 9/29 Propositoinal Logic & BDD technology
4. 10/6 Propositoinal Logic & BDD technology
   1st project announcement
5. 10/13 Propositoinal Logic & BDD technology
6. 10/20 State Machines & Learning
7. 10/27 State Machines
8. 11/3 Temporal Logics & Symbolic Model-Checking
   1st project report, 2nd project announcement

Course schedule (continued)

9. 11/10 Midterm Exam
10. 11/17 Temporal Logics & Symbolic Model-Checking
11. 11/24 Temporal Logics & Symbolic Model-Checking
12. 12/1 Embedded Systems
13. 12/8 Embedded Systems & Symbolic Model-Checking
    2nd project report, 3rd project announcement.
14. 12/15 Simulation & Bisimulation
15. 12/22 Games
16. 12/29 Model-based Testing
17. 1/5 3rd project report
18. 1/12 Final Exam

課程網頁

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

助教：吳哲榮
b93901098@ntu.edu.tw

Evaluation

Two scenarios

- With paper presentation
  midterm: 25%, final: 30%, projects: 30%, paper presentation: 15%
- Without paper presentation
  midterm: 30%, final: 30%, projects:30%, homework: 10%

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