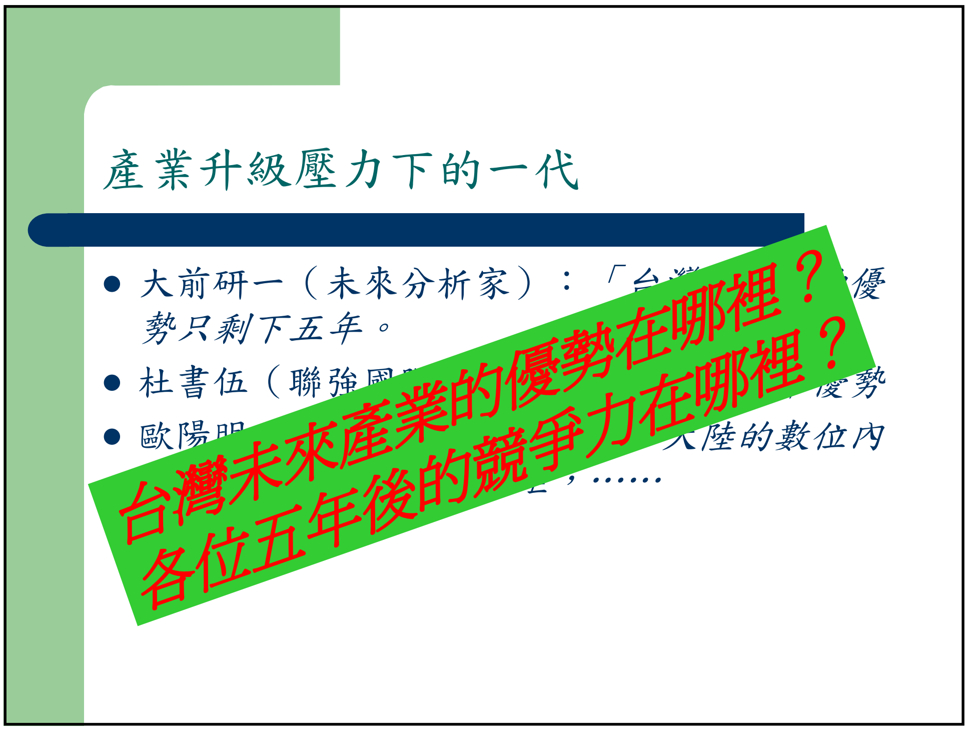
正規描述與自動驗證

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……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！

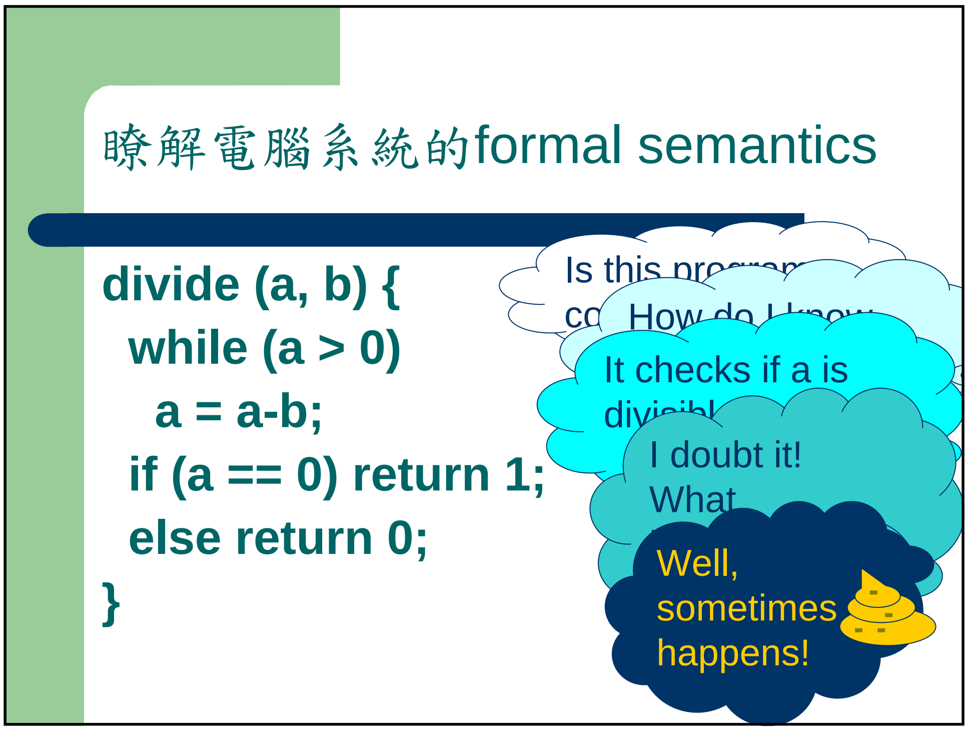
各位同學的一條生路！

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

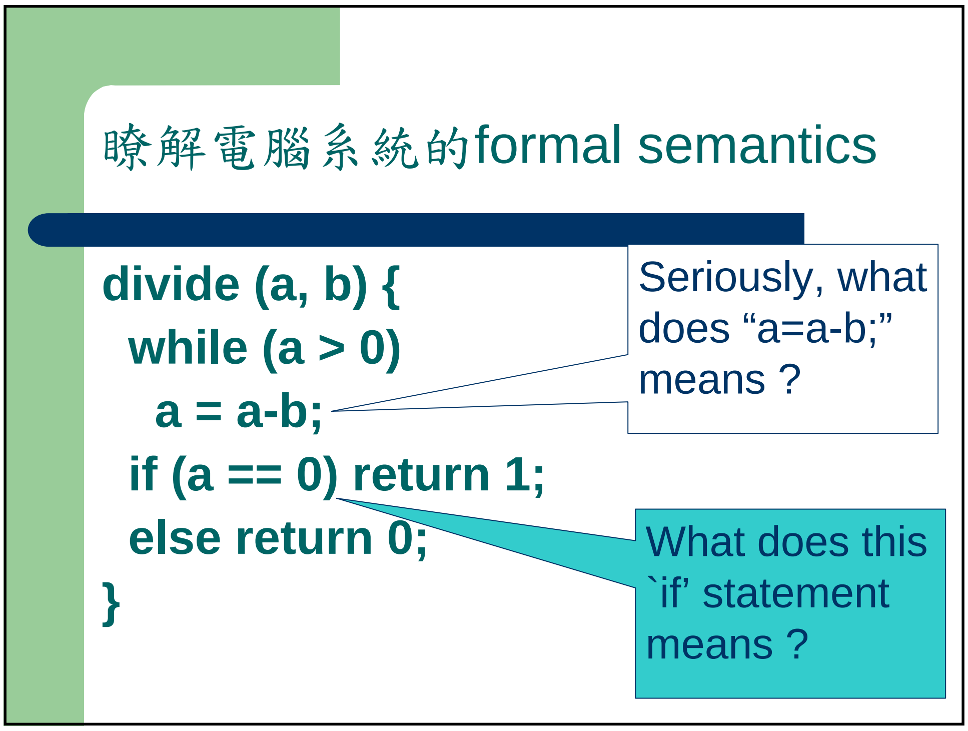
當然愛你啦！北鼻！
人家嗎？
你曾經 說，永 愛著我！
…??
…??
…!!
…!!
如果你媽和我一起掉下水，你要先救 誰？
厚！以後存款、房子是誰的名下？



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!

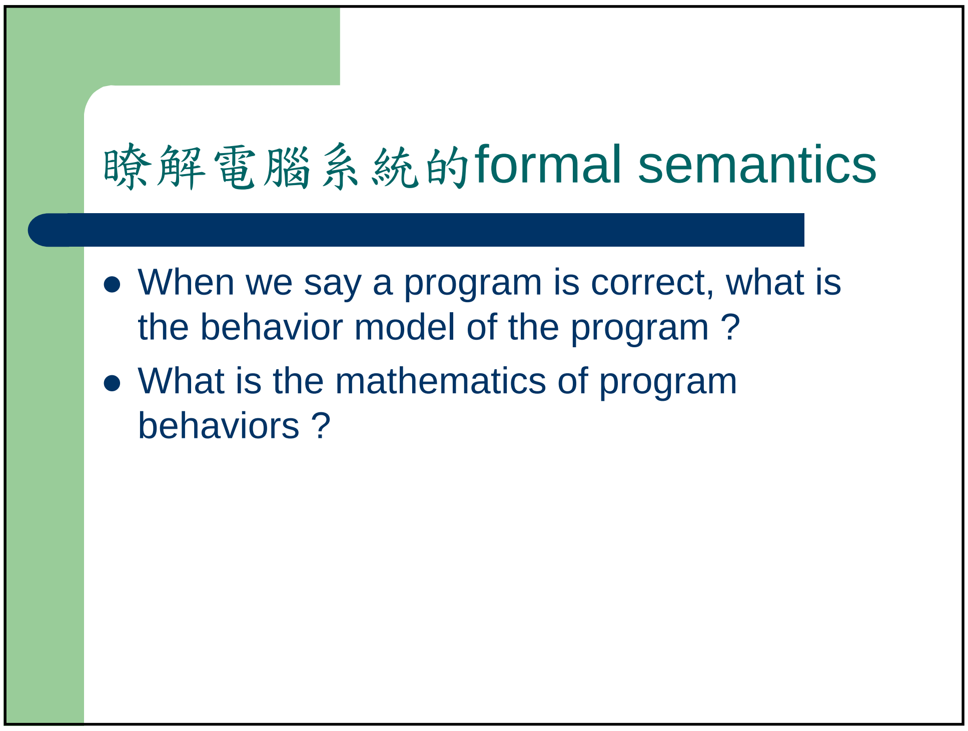


瞭解電腦系統的formal semantics

```
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 ?

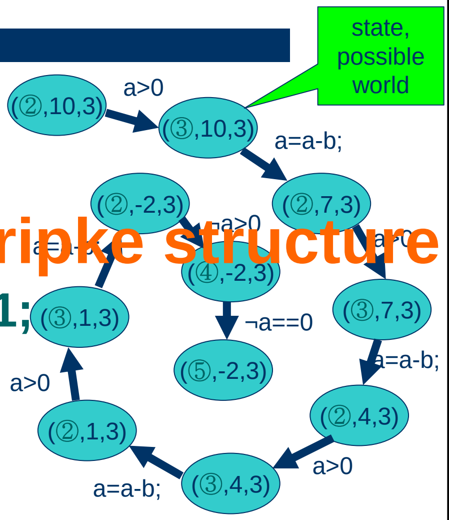


瞭解電腦系統的formal semantics

- 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

```
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

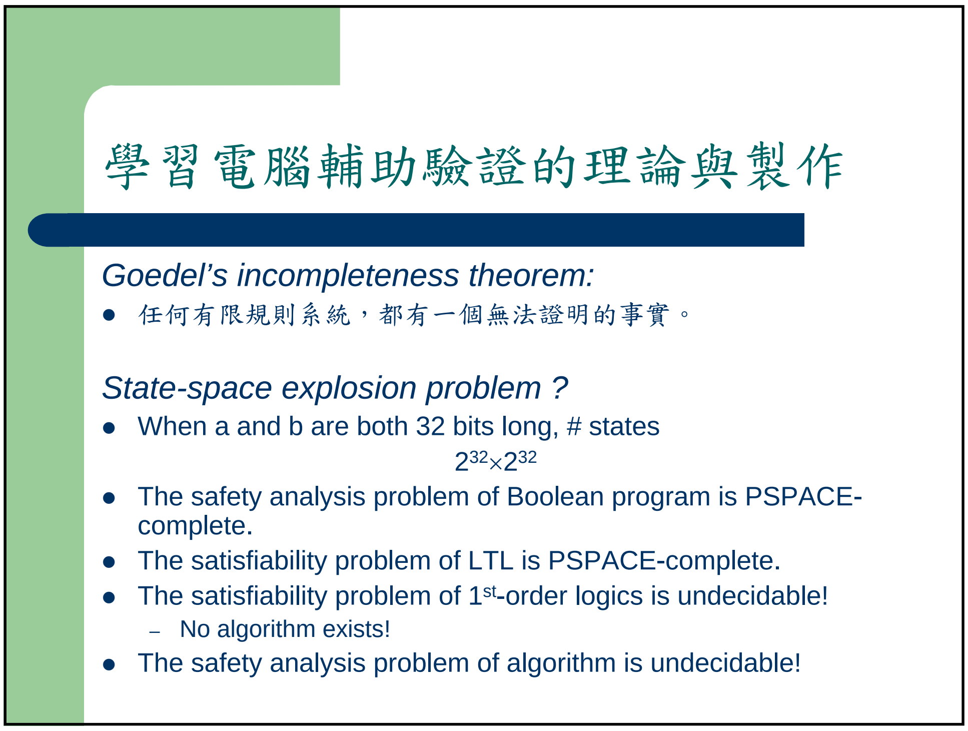
```
①divide (a, b) {     p(k
②  while (a > 0)
③    a = a-b
④  if (a ==   re
⑤  else ret
⑥}
```

tate k;
tate k;
te k;

∀k≥0(p(k) a(k+1)==a(k)-b(k))

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

太棒了！北鼻！有了logics，我們現在可以清楚瞭解彼此的行為與結構。
兩位好瞎！一定沒有修過離散、邏輯。Goedel已經證明有些定理是邏輯無法判別的。
即使可以判別，但也可能有state-space爆炸問題呀！



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 ?

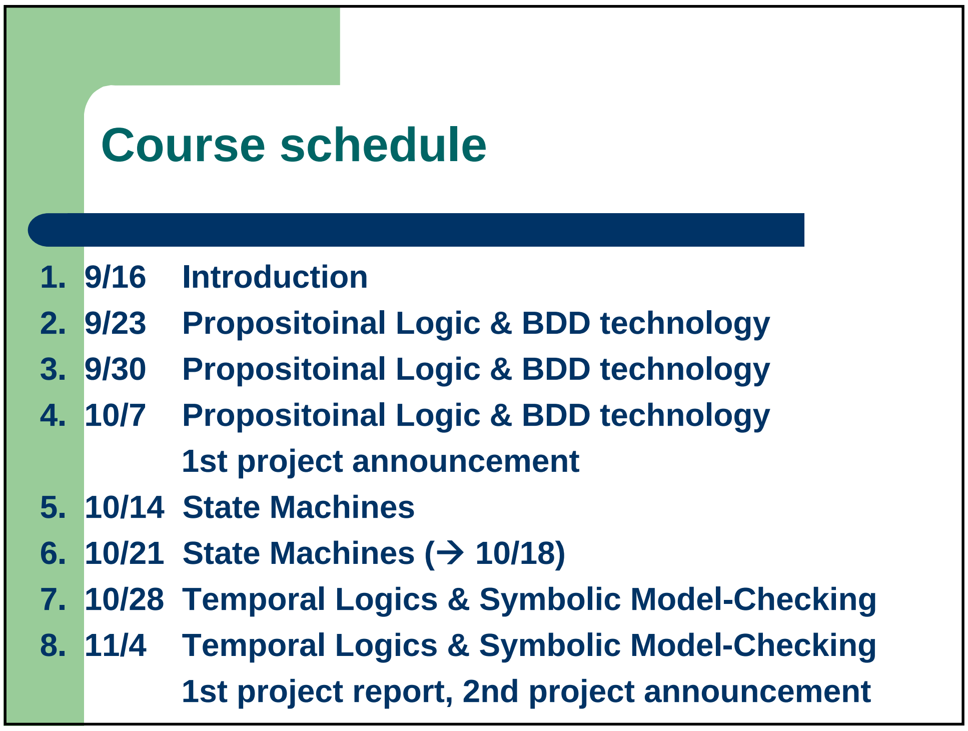
- When a and b are both 32 bits long, # states
  $2^{32}\times 2^{32}$
- The safety analysis problem of Boolean program is PSPACE-complete.
- The satisfiability problem of LTL is PSPACE-complete.
- The satisfiability problem of 1st-order logics is undecidable!
  - No algorithm exists!
- The safety analysis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

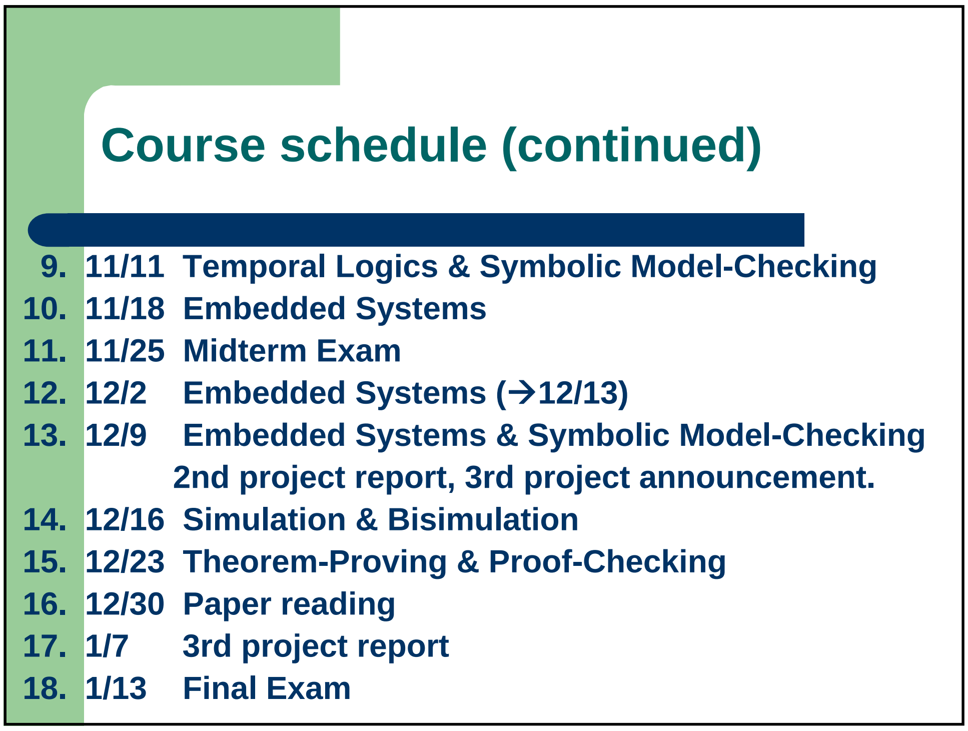
Course plan：

- 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 construct system model and making verification for timed systems



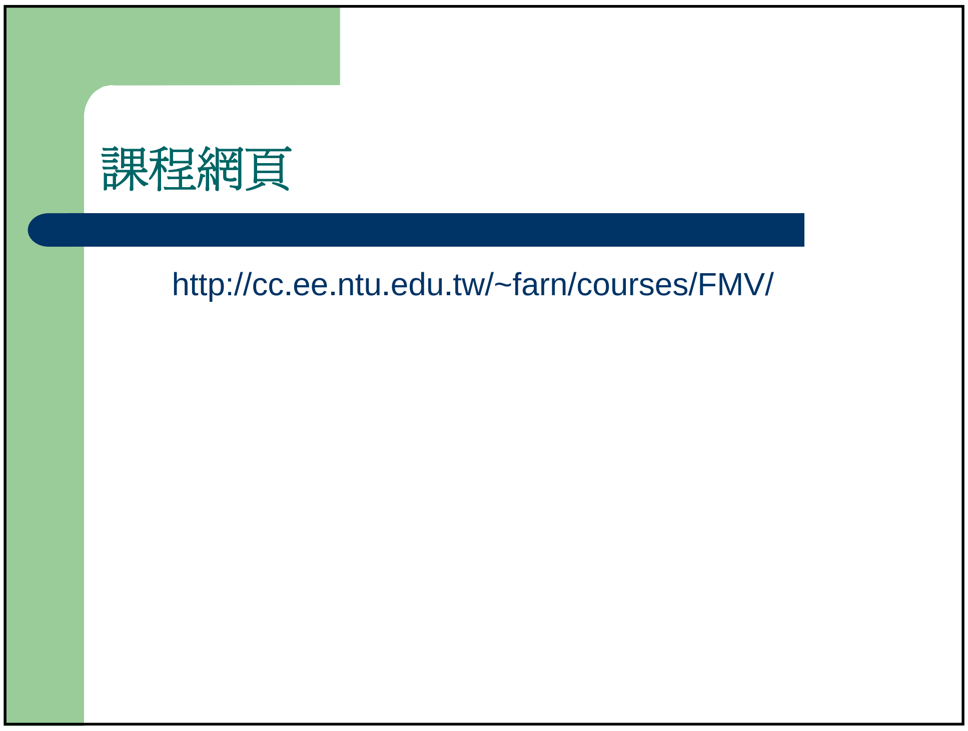
Course schedule

1. 9/16 Introduction
2. 9/23 Propositoinal Logic & BDD technology
3. 9/30 Propositoinal Logic & BDD technology
4. 10/7 Propositoinal Logic & BDD technology
   1st project announcement
5. 10/14 State Machines
6. 10/21 State Machines (→ 10/18)
7. 10/28 Temporal Logics & Symbolic Model-Checking
8. 11/4 Temporal Logics & Symbolic Model-Checking
   1st project report, 2nd project announcement



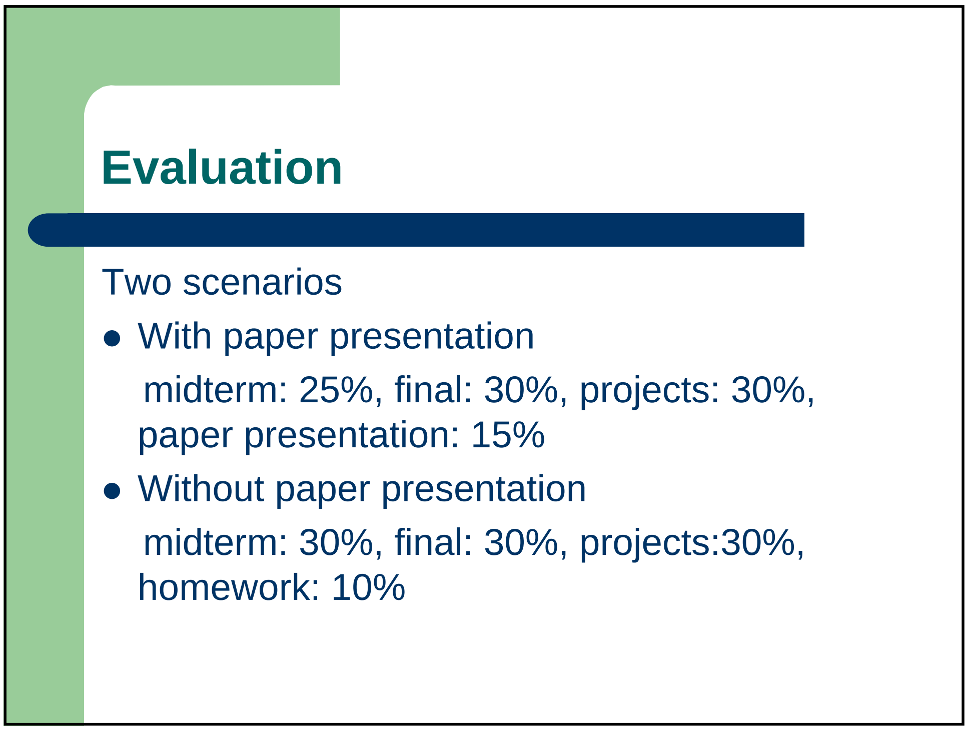
Course schedule (continued)

9. 11/11 Temporal Logics & Symbolic Model-Checking
10. 11/18 Embedded Systems
11. 11/25 Midterm Exam
12. 12/2 Embedded Systems (→12/13)
13. 12/9 Embedded Systems & Symbolic Model-Checking
    2nd project report, 3rd project announcement.
14. 12/16 Simulation & Bisimulation
15. 12/23 Theorem-Proving & Proof-Checking
16. 12/30 Paper reading
17. 1/7 3rd project report
18. 1/13 Final Exam



課程網頁

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



Evaluation

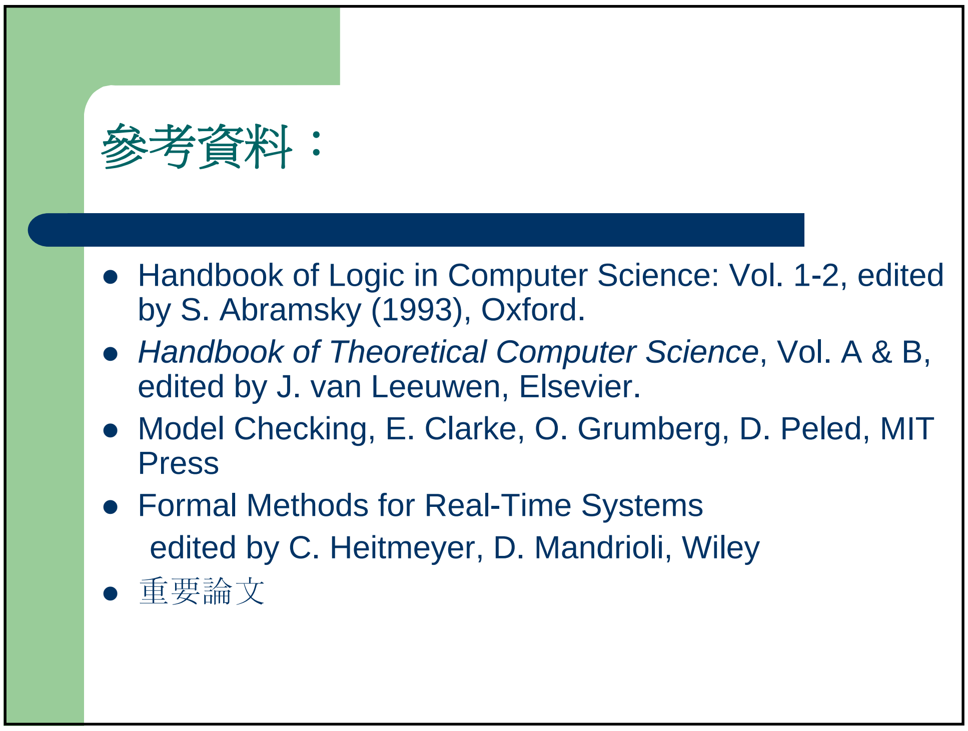
Two scenarios

- 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