Computer-Aided Verification
計算機輔助驗證
(Spring 2006)

熊博安

國立中正大學資訊工程研究所

http://www.cs.ccu.edu.tw/~pahsiung/courses/cav/

pahsiung@cs.ccu.edu.tw

(05)2720411 ext. 33119

Class: EA-205

Office: EA-512

Course Information

- **Theory + Practice**
  - **Theory: formal specification, modeling, verification**
  - **Practice: tool use, algorithm implementation, etc.**
- **Formal more than informal Verification**
  - **Semi-formal: Verification + Simulation**
- **Tools:**
  - **Model checkers: SGM, SMV, SPIN, FormalCheck, …**
  - **Simulators: Modelsim, SystemC**
- **Target Systems**:
  - Software, Hardware, Protocols, Abstract Specifications, …
  - Real-Time Systems, Embedded Systems, HW-SW SoC, …

Who should take this course?

- Interested in FORMALLY VERIFYING system correctness
- Interested in PRACTICALLY IMPLEMENTING verification theories
- MUST: C or C++ programming & tracing
- Not scared of THEORY!
- Love using TOOLS!!

Who should NOT take this course?

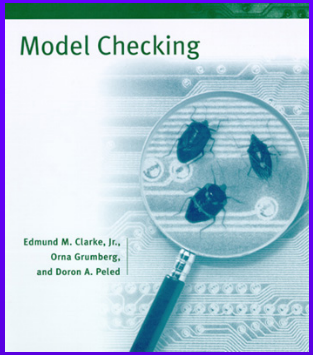
- Only wants course credits
- Does not like research
- Does not like projects and labs
- Does not like using tools
- Does not like formal theory
- Not creative (lack of new ideas)
- Yawns and goes to sleep when someone is talking about Finite State Machines!

Course Syllabus & Schedule

Contents	Week
♦ Introduction to CAV	1, 2
♦ Introduction to Model Checking	3
♦ System Model & Logic Specification	4, 5
♦ Explicit/Symbolic Model Checking	6, 7, 8
♦ Mid-Term Exam	9
♦ BMC, SAT	10
♦ Assume-Guarantee Reasoning	11, 12
♦ Priority and Urgency Verification	13
♦ Coverage Analysis	14
♦ Paper Presentations	15
♦ Project Results Presentations	16
♦ Final Exam	17

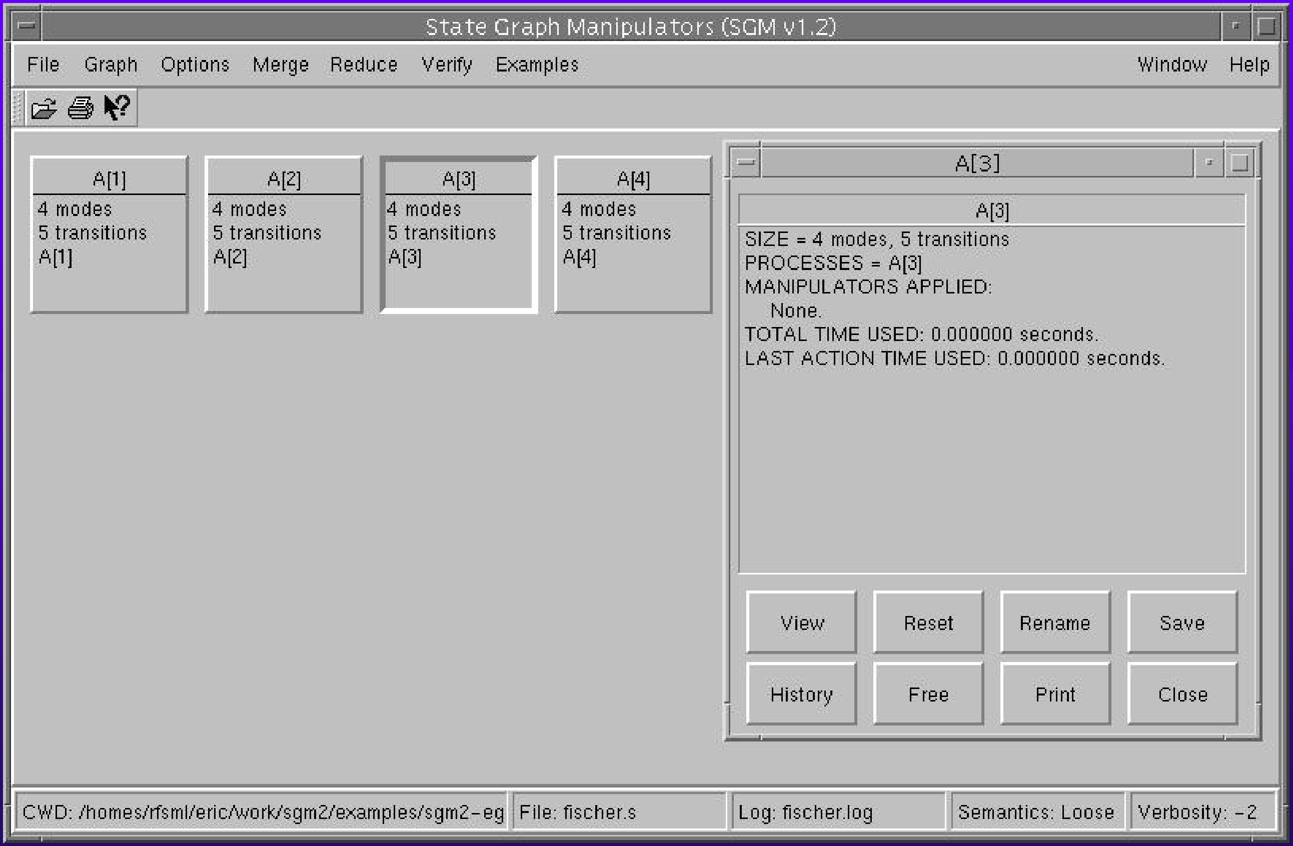
Reference Book

- Edmund M. Clarke, Orna Grumberg, and Doron A. Peled, "Model Checking," MIT Press, 1999.
  (圖書館有)



Verification Tool

- **SGM Model Checker,** Pao-Ann Hsiung and Farn Wang, http://www.cs.ccu.edu.tw/~pahsiung/sgm/



A[1]
4 modes
5 transitions
A[1]

A[2]
4 modes
5 transitions
A[2]

A[3]
4 modes
5 transitions
A[3]

A[4]
4 modes
5 transitions
A[4]

```
A[3]
SIZE = 4 modes, 5 transitions
PROCESSES = A[3]
MANIPULATORS APPLIED:
    None.
TOTAL TIME USED: 0.000000 seconds.
LAST ACTION TIME USED: 0.000000 seconds.
```

CWD: /homes/rfsml/eric/work/sgm2/examples/sgm2-eg | File: fischer.s | Log: fischer.log | Semantics: Loose | Verbosity: -2

Deadlines and Grading

Work	Deadline	Grade
Labs & Assignments	2 weeks due	25%
Project Proposal	March 29, 2006	N/A
Project Work	April, May 2006	N/A
Paper Selection	April 19, 2006	N/A
Mid-Term Exam	April 19, 2006	20%
Paper Presentation	June 7, 2006	15%
Project Report & Present	June 14, 2006	20%
Final Exam	June 21, 2006	20%

Term Project

- Form a team of 2 persons at most
- Proposal (Deadline: March 29)
  - Topic, summary, members, goals
- Presentation (Date: June 14)
  - 10 ~ 15 slides
  - 15 ~ 20 minutes talk
  - 5 minutes Q/A
- Report (Deadline: June 14)
  - See course web page for details

Project Topic (1)

- Verify a communication protocol (Bluetooth, 802.11, ATM, WAP, etc.)
  - Protocol modeled by extended timed automata (ETA)
  - Protocol requirements specified in CTL
  - Verification results

Project Topic (2)

- Verify a piece of real-time embedded software (RTES)
  - Software and hardware modeled by ETA
  - RTES requirements specified in CTL
  - Verification results

Project Topic (3)

- Verify a System-on-a-Chip (SoC)
  - SoC modeled by ETA
  - SoC design requirements specified in CTL
  - Verification results

Project Topic (4)

- Develop & Implement your own analysis technique in SGM (BONUS: 40%)
  - Algorithm for your reduction technique
  - Proof of correctness for your algorithm
  - Implementation of your algorithm in SGM
  - Application examples
  - Reduction results

Project Topic (5)

- Develop a counterexample graphical viewer (signal timing diagram, MSC, UML sequence diagram, ...) (BONUS: 40%)
  - Representation modeling (how is a counterexample represented by your selected view from above)
  - Proof of equivalence (counterexample == represented view)
  - Implementation in SGM
  - Application examples

Project Topic (6)

- Develop an interface in SGM for Safecharts (BONUS: 50%)
  - Input syntax for Safecharts
  - Semantics storing for Safecharts
  - Translation from Safecharts to extended timed automata

Project Topic (7)

- Develop an interface in SGM for UML State Machines (formerly called Statecharts) (BONUS: 40%)
  - Input syntax for State Machines
  - Semantics storing for State Machines
  - Translation from State Machines to extended timed automata

Labs

- Must be done individually
- Topics
  - Model check some simple protocols, systems, software programs, or hardware circuits on
    - SGM
    - SMV
    - SPIN
    - RED

Assignments

- ♦ Individual work
- ♦ Due 2 weeks after announcement
- ♦ Written homeworks
  - – CTL specification
  - – System model
  - – Model transformation

Paper Reading

- Individual work
- Paper Selection (Deadline: April 19)
  - CAV related papers
    - Model checking
    - Formal verification
    - Theory, Practice, Tools, Case Studies
    - Conferences: CAV, CONCUR, FORTE, FME, LICS, STOCS, …
    - Journal: IEEE trans, ACM trans, FMSD, …
- Presentation (Date: June 7)
  - 15 ~ 20 slides
  - 15 ~ 20 minutes talk
  - 5 minutes Q/A
- Written Report (Deadline: June 7)
  - 1 page summary (don’t copy abstract!)

ENJOY THE COURSE (as much as you can!)