Model Checking
A Tutorial Introduction

Seminar: Sicherheitskritische Systeme

Modelchecking--Eine Einführung

(Model Checking : A tutorial introduction)

Bearbeiter: Yuguo Sun

Introduction

Motivation, Purpose of System Verification

Examples:

- Pentium bug Intel Pentium chip
- ARIANE Failure
- Therac-25 Accident

Four principal techniques for ensuring the correctness of hardware and software systems:

- Simulation
- Testing.
- Deductive Verification
- Model Checking

System Verification via Model Checking

- What is Model Checking
- Compared with other techniques

The Process of Model Checking

3 Steps of the Model Checking

- modelling
- specification
- verification.

Step.1 --- Modelling

- what is Modelling : convert the system into a formalism. For the modelling of systems we use finite automats.

- owing to limitations on time and memory, the modelling of a design may require the use of abstraction

- We use a type of state transition graph called a Kripke structure to model a system

what is Kripke

A Kripke structure over a set of atomic propositions

AP is a four-tuple; M = (S, S0, R, L) where

- S is a finite set of states.
- S0 $\subseteq$ S is the set of initial states.
- R $\subseteq$ S $\times$ S is a transition relation
- L: S$\rightarrow$2(AP) is a function that labels each state with the set of atomic propositions true in this state.

Example with micro-oven cooking

Modelling with the Kripke-Structure

M = (S, S0, R, L)

- S = (S1, S2, S3, S4)
- S1 is the initial state
- R = ({S1, S2} {S2, S1}, {S1, S4}, {S4, S2}, {S2, S3}, {S3, S2}, {S3, S3}
- L (S1) = {¬close, ¬ start, ¬ cooking} L (S2) = {close, ¬ start, ¬ cooking} L (S3) = {close, start, cooking} L (S4) = {¬close, start, ¬ cooking}

Graph of the Kripke-Structure M of microwave-oven

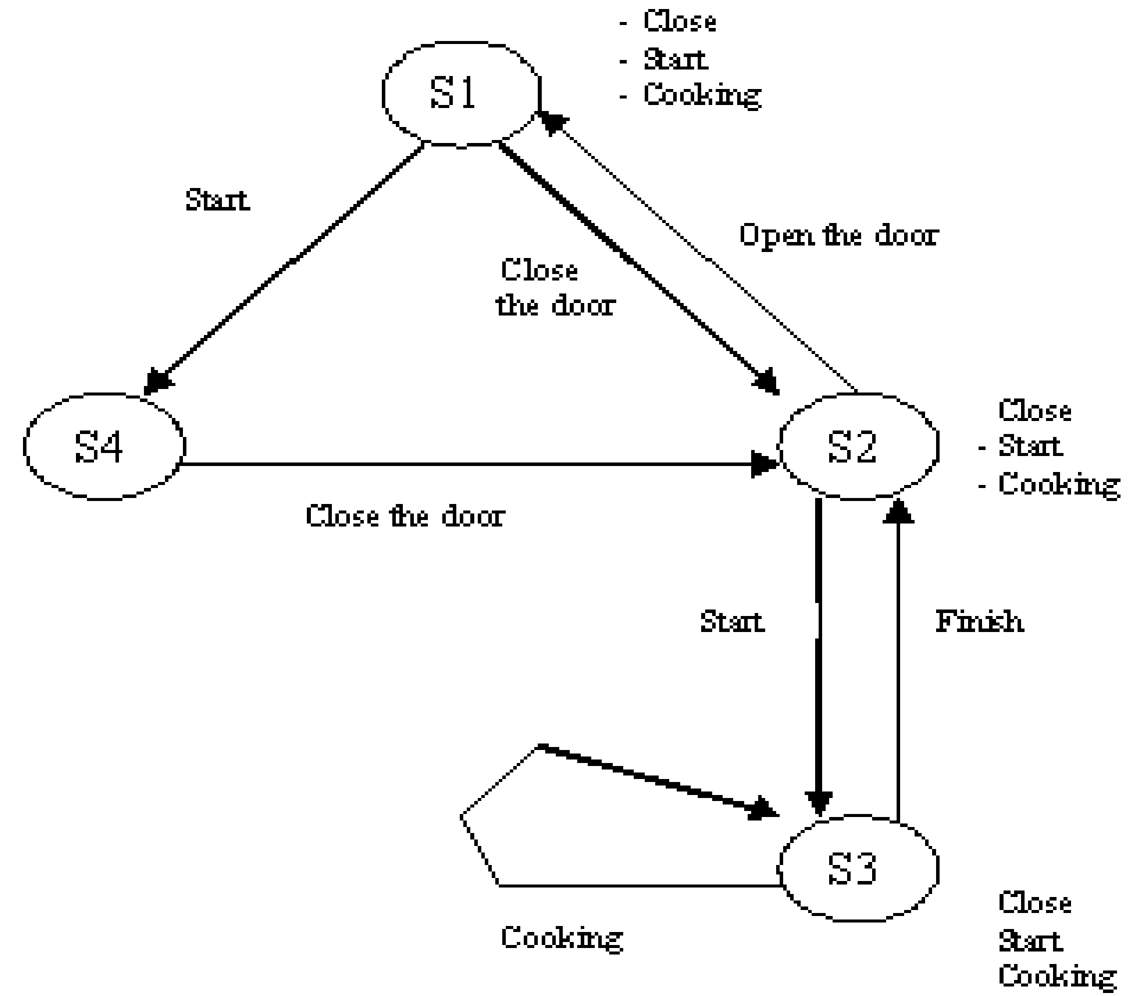


Figure: Automat

Step.2 --- Specification

- What is Specification
- Classical Logic
- Temporal Logic

Operators for the Temporal Logic

- five basic temporal

1. X (‘‘next time’’)
2. F (‘‘in the future’’)
3. G (‘‘globally’’)
4. U (‘‘until’’)
5. R (‘‘Release’’)

- Two quantifiers for the Temporal Logic

1. A (‘‘always’’)
2. E (‘‘exists’’)

Three main ways to represent Temporal Logic:

- CTL* (Computation Tree Logic*)
- CTL (Computation Tree Logic) with 10 basis operators: AX and EX; AF and EF; AG and EG; AU and EU; AR and ER.
- LTL (Linear Temporal Logic)

* Completeness?

Example with microwave-oven cooking

Specification with CTL-Formal

1. $AG\ (start \Rightarrow AF\ cooking)$

2. $AG\ ((close \wedge start) \Rightarrow AF\ cooking)$

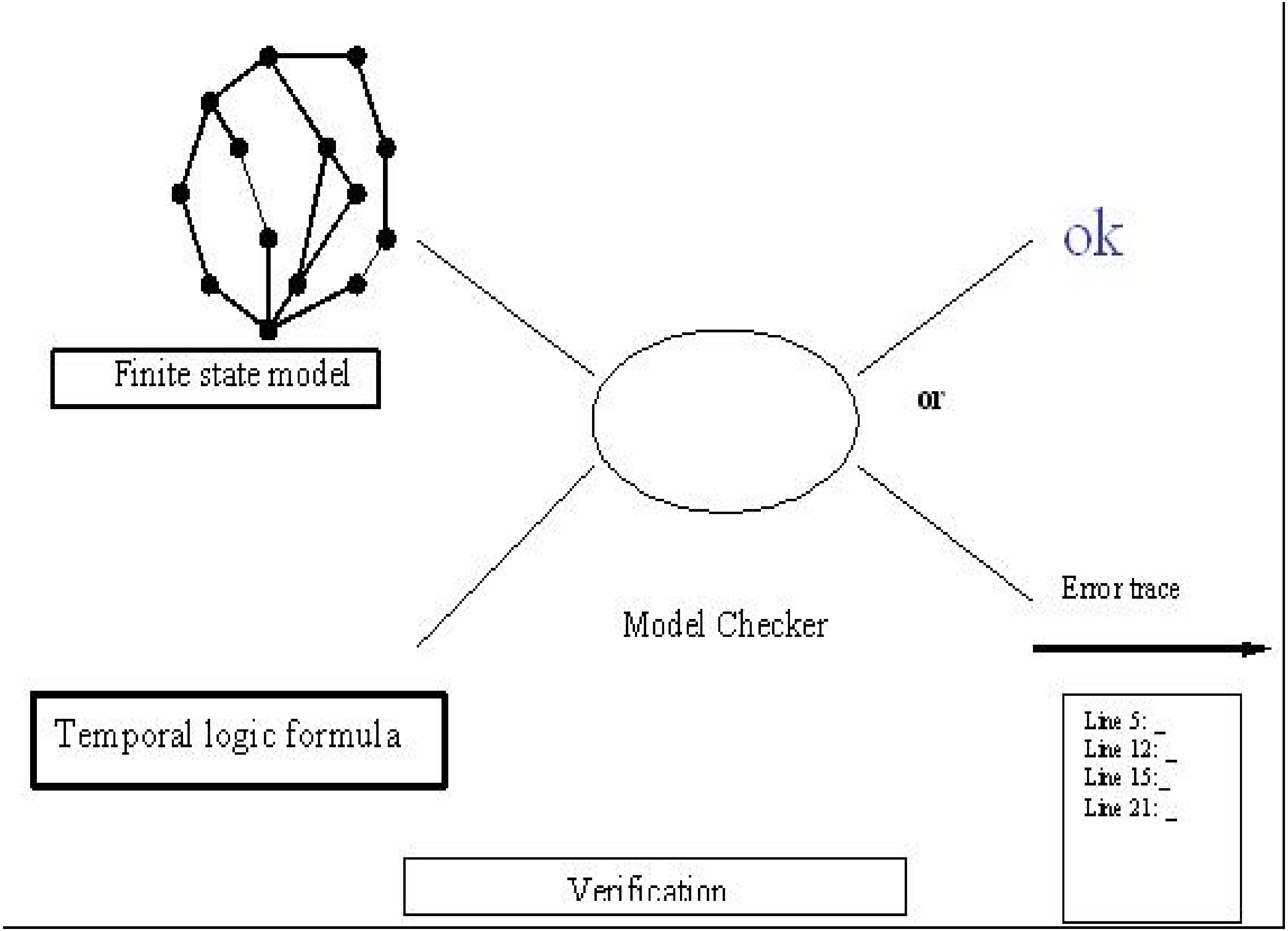
Step.3 --- Verification

CTL* -Model-Checking

CTL -Model-Checking

LTL -Model-Checking

- Human assistance ? + Error trace


Finite state model
ok
or
Model Checker
Error trace
Temporal logic formula
Line 5: _
Line 12: _
Line 15: _
Line 21: _
Verification

Example with microwave-oven cooking (1)

To the first CTL-Formal : AG (start Þ AF cooking)

1) Change formal to ¬EF (start ∧ EG ¬ cooking))

2) From simple partial formulas to the more complicated formulas, until all of the formulas are true.

- S (start) = {S3, S4}
- S (¬cooking) = {S1, S2, S4}
- S (EG ¬ cooking) = {S1, S2, S4} (all conditions lie on a path)
- S (start ∧ EG ¬ cooking) = {S4}
- S (EF (start ∧ EG ¬ cooking)) = {S1, S2, S3, S4} (can be followed with S4)
- S (¬ (EF (start ∧ EG ¬ cooking))) = {}

3) Result analyze

Example with microwave-oven cooking(2)

To the second CTL-Formal: AG ((close Ù start) Þ AF cooking)

1) change formal to ¬ EF(close ∧ start ∧ EG ¬ cooking)

2) Now the algorithm can be applied to the formula

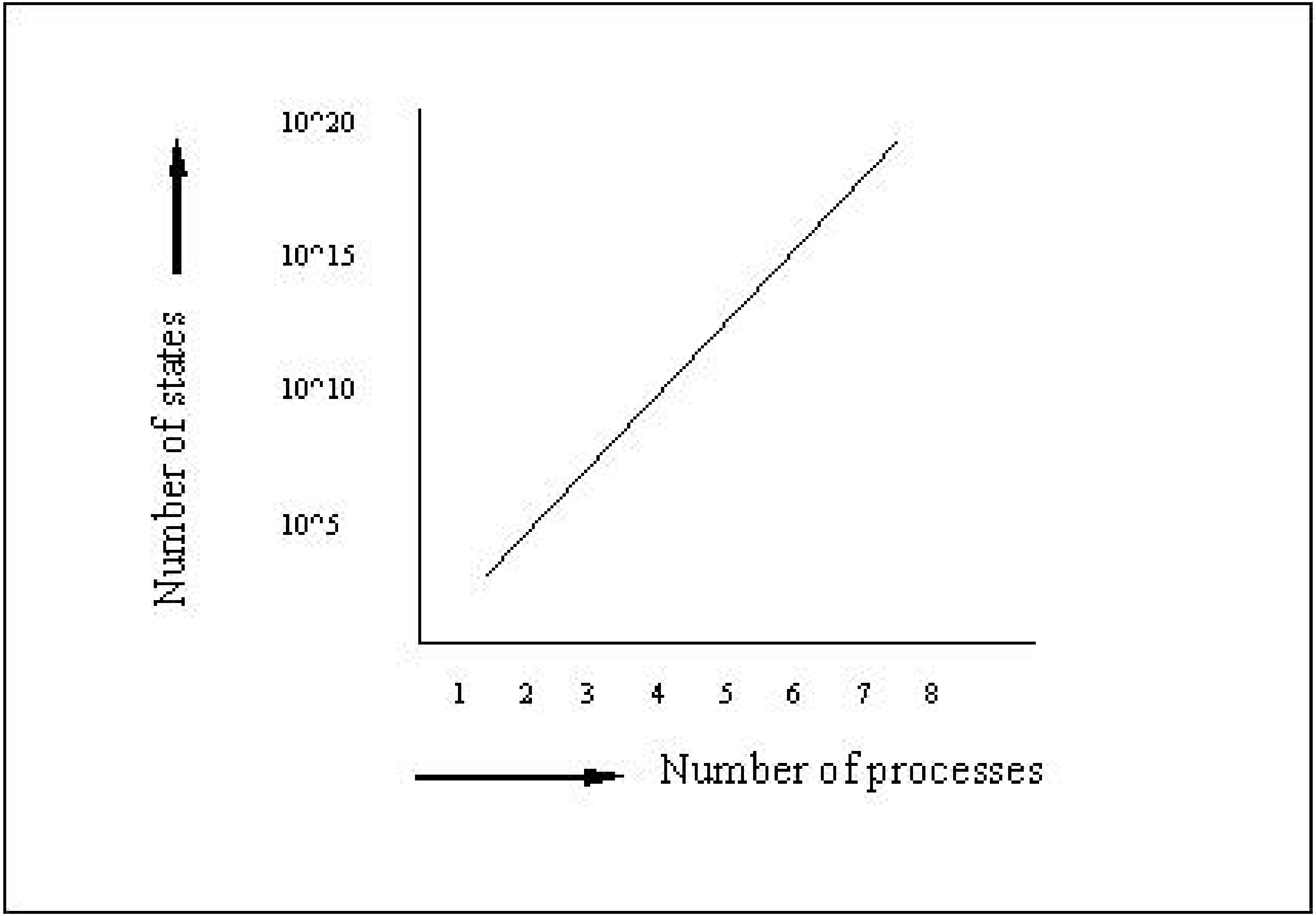
- S (close)= {S2, S3}
- S (start)= {S3, S4}
- S (¬ cooking) = {S1, S2, S4}
- S (EG ¬ cooking) = {S1, S2, S4}
- S (close ∧ start ∧ EG ¬ cooking) = {}
- S (EF (close ∧ start ∧ EG ¬ cooking) = {}
- S (¬ (EF (close ∧ start v EG ¬ cooking)) = {S1, S2, S3, S4}

3) Result analyze

Algorithms for Model Checking

- State space explosion problem

- Number of states typically grows exponentially in the number of process


10^20
10^15
10^10
10^5
Number of states
1
2
3
4
5
6
7
8
Number of processes

The major techniques for tackling this problem

- Based on Automata Theory
- Based on Symbolic Structure
- Other Methods -- Alternative Methods

Based on Automata Theory (1)

On the Fly Technology

- Definition
- Intersection in the “on-the-fly” mode checking
- Advantage of on-the-fly model checking

Based on Automata Theory (2)

Partial-Oder Reduction Technology

a) what is interleaving

b) what is partial-order representation

c) Three kinds of the partial-order reduction Technology

- dynamic partial- order reduction Technology
- Static partial-order reduction Technology
- the purely partial-order reduction Technology

Based on Symbolic Structure

- Symbolic Structure with a Boolean formula
- binary decision diagram (BDD)
- 10^5 states -- 10^20 states -- 10^120 states
- SMV language and OBDD (Bryant's ordered binary decision diagrams)
- Successful examples with SMV

Alternative Methods

- Equivalence
- Compositional Reasoning
- Abstraction
- Symmetry
- Induction

Alternative Methods

Equivalence

- What is equivalence technique
- Simulation equivalence and bisimulation equivalence

Compositional Reasoning

- What is compositional Reasoning technique
- An example
- Assume-guarantee reasoning

Alternative Methods

Abstraction

- Abstraction with mapping
- The cone of influence reduction and the data abstraction

Induction

- What is induction technique
- An example

Symmetry

- What is symmetry technique
- Examples

Model Checking for real-time Systems

- What is real-time systems
- Why it is particularly difficult to make the validation of real-time systems

Discrete real-time System

- Synchronous system
- An example
- Rate-monotonic scheduling theory (RMS)

Continuous real-time System

- Asynchronous system
- Fixed time quamtum

Model Checking developing Trend

- Relatively straightforward extensions of current systems
- Require more theoretical work
- Use combination of the abstraction and compositional reasoning techniques
- Probabilistic verification
- The ability to reason automatically about entire families of finite-state systems
- Investigation Model Checking techniques combined with theorem proving

Conclusion

- In conclusion
- Apply in industry

Vielen Dank für Ihre Aufmerksamkeit !

Universität Siegen, Fachbereich 12, Elektronik und Informatik, SS 2004