FORMAL METHODS AND ALL THAT

Ting Zhang

Theory Group
Microsoft Research Asia

August 24, 2007
OUTLINE

1. Overview

2. Specification Formalisms
   - Temporal Logics

3. Computational Models
   - Fair Transition System

4. Static Analysis Methods
   - Invariant Generation

5. Synthesis Methods
   - Game-theoretic Approach

6. Computational Engines
   - Decision Procedures
OUTLINE

1. OVERVIEW
2. SPECIFICATION FORMALISMS
   - Temporal Logics
3. COMPUTATIONAL MODELS
   - Fair Transition System
4. STATIC ANALYSIS METHODS
   - Invariant Generation
5. SYNTHESIS METHODS
   - Game-theoretic Approach
6. COMPUTATIONAL ENGINES
   - Decision Procedures
OUTLINE

1. OVERVIEW
2. SPECIFICATION FORMALISMS
   - Temporal Logics
3. COMPUTATIONAL MODELS
   - Fair Transition System
4. STATIC ANALYSIS METHODS
   - Invariant Generation
5. SYNTHESIS METHODS
   - Game-theoretic Approach
6. COMPUTATIONAL ENGINES
   - Decision Procedures
Outline

1. Overview
2. Specification Formalisms
   - Temporal Logics
3. Computational Models
   - Fair Transition System
4. Static Analysis Methods
   - Invariant Generation
5. Synthesis Methods
   - Game-theoretic Approach
6. Computational Engines
   - Decision Procedures
WHO ARE WE?

- **Wei Chen**: Lead Researcher
  Ph.D. in Computer Science, Cornell University, 2001

- **Prof. Xiang-Yang Li**: Visiting Professor
  Associate Professor in Computer Science, IIT
  Ph.D. in Computer Science, UIUC, 2001

- **Prof. Shang-Hua Teng**: Visiting Professor
  Professor in Computer Science, Boston University
  Ph.D. in Computer Science, CMU, 1991

- **Ting Zhang**: Researcher
  Ph.D. in Computer Science, Stanford University, 2006
WHO ARE WE?

- **Wei Chen**: Lead Researcher
  Ph.D. in Computer Science, Cornell University, 2001

- **Prof. Xiang-Yang Li**: Visiting Professor
  Associate Professor in Computer Science, IIT
  Ph.D. in Computer Science, UIUC, 2001

- **Prof. Shang-Hua Teng**: Visiting Professor
  Professor in Computer Science, Boston University
  Ph.D. in Computer Science, CMU, 1991

- **Ting Zhang**: Researcher
  Ph.D. in Computer Science, Stanford University, 2006
WHO ARE WE?

- **Wei Chen**: Lead Researcher
  Ph.D. in Computer Science, Cornell University, 2001

- **Prof. Xiang-yang Li**: Visiting Professor
  Associate Professor in Computer Science, IIT
  Ph.D. in Computer Science, UIUC, 2001

- **Prof. Shang-hua Teng**: Visiting Professor
  Professor in Computer Science, Boston University
  Ph.D. in Computer Science, CMU, 1991

- **Ting Zhang**: Researcher
  Ph.D. in Computer Science, Stanford University, 2006
WHO ARE WE?

- **Wei Chen**: Lead Researcher  
  Ph.D. in Computer Science, Cornell University, 2001

- **Prof. Xiang-Yang Li**: Visiting Professor  
  Associate Professor in Computer Science, IIT  
  Ph.D. in Computer Science, UIUC, 2001

- **Prof. Shang-Hua Teng**: Visiting Professor  
  Professor in Computer Science, Boston University  
  Ph.D. in Computer Science, CMU, 1991

- **Ting Zhang**: Researcher  
  Ph.D. in Computer Science, Stanford University, 2006
What Are We Doing?

- **Wei Chen**: Theory of Distributed Computing, Fault Tolerance
- **Prof. Xiang-yang Li**: Networking Algorithms, Cryptography and Network Security, Computational Geometry
- **Prof. Shang-hua Teng**: Computational Geometry, Combinatorial Optimization, Computational Game Theory
- **Ting Zhang**: Formal Methods, Automated Deduction, Decision Procedures
What Are We Doing?

**Wei Chen**: Theory of Distributed Computing, Fault Tolerance

**Prof. Xiang-yang Li**: Networking Algorithms, Cryptography and Network Security, Computational Geometry

**Prof. Shang-hua Teng**: Computational Geometry, Combinatorial Optimization, Computational Game Theory

**Ting Zhang**: Formal Methods, Automated Deduction, Decision Procedures
What Are We Doing?

- **Wei Chen**: Theory of Distributed Computing, Fault Tolerance
- **Prof. Xiang-yang Li**: Networking Algorithms, Cryptography and Network Security, Computational Geometry
- **Prof. Shang-hua Teng**: Computational Geometry, Combinatorial Optimization, Computational Game Theory
- Ting Zhang: Formal Methods, Automated Deduction, Decision Procedures
What Are We Doing?

- **Wei Chen**: Theory of Distributed Computing, Fault Tolerance
- **Prof. Xiang-yang Li**: Networking Algorithms, Cryptography and Network Security, Computational Geometry
- **Prof. Shang-hua Teng**: Computational Geometry, Combinatorial Optimization, Computational Game Theory
- **Ting Zhang**: Formal Methods, Automated Deduction, Decision Procedures
FORMAL METHODS AND ALL THAT

Overview

FORMAL METHODS AND ALL THAT

- **SPECIFICATION FORMALISMS**: first-order logic, temporal logic (LTL, CTL, μ-Calculus, ATL), automata, Petri Nets

- **COMPUTATIONAL MODELS**: transition systems, real-time systems, hybrid systems, event-based systems

- **STATIC ANALYSIS METHODS**: invariant generation, termination analysis

- **VERIFICATION METHODS**: deductive, algorithmic, abstraction, diagrams

- **SYNTHESIS METHODS**: deductive, algorithmic, game-theoretic, diagrams

- **COMPUTATIONAL ENGINES**: decision procedures, quantifier elimination methods, constraint solvers
FORMAL METHODS AND ALL THAT

Overview

FORMAL METHODS AND ALL THAT

- **Specification formalisms**: first-order logic, temporal logic (LTL, CTL, $\mu$-Calculus, ATL), automata, Petri Nets
- **Computational models**: transition systems, real-time systems, hybrid systems, event-based systems
- **Static analysis methods**: invariant generation, termination analysis
- **Verification methods**: deductive, algorithmic, abstraction, diagrams
- **Synthesis methods**: deductive, algorithmic, game-theoretic, diagrams
- **Computational engines**: decision procedures, quantifier elimination methods, constraint solvers
Formal Methods and All That

Overview

Formal Methods and All That

- **Specification formalisms**: first-order logic, temporal logic (LTL, CTL, μ-Calculus, ATL), automata, Petri Nets
- **Computational models**: transition systems, real-time systems, hybrid systems, event-based systems
- **Static analysis methods**: invariant generation, termination analysis
- **Verification methods**: deductive, algorithmic, abstraction, diagrams
- **Synthesis methods**: deductive, algorithmic, game-theoretic, diagrams
- **Computational engines**: decision procedures, quantifier elimination methods, constraint solvers
FORMAL METHODS AND ALL THAT

Overview

**FORMAL METHODS AND ALL THAT**

- **SPECIFICATION FORMALISMS**: first-order logic, temporal logic (LTL, CTL, μ-Calculus, ATL), automata, Petri Nets
- **COMPUTATIONAL MODELS**: transition systems, real-time systems, hybrid systems, event-based systems
- **STATIC ANALYSIS METHODS**: invariant generation, termination analysis
- **VERIFICATION METHODS**: deductive, algorithmic, abstraction, diagrams
- **SYNTHESIS METHODS**: deductive, algorithmic, game-theoretic, diagrams
- **COMPUTATIONAL ENGINES**: decision procedures, quantifier elimination methods, constraint solvers
Formal Methods and All That

Overview

Formal Methods and All That

- **Specification formalisms**: first-order logic, temporal logic (LTL, CTL, μ-Calculus, ATL), automata, Petri Nets
- **Computational models**: transition systems, real-time systems, hybrid systems, event-based systems
- **Static analysis methods**: invariant generation, termination analysis
- **Verification methods**: deductive, algorithmic, abstraction, diagrams
- **Synthesis methods**: deductive, algorithmic, game-theoretic, diagrams
- **Computational engines**: decision procedures, quantifier elimination methods, constraint solvers
Formal Methods and All That

Overview

**Formal Methods and All That**

- **Specification formalisms**: first-order logic, temporal logic (LTL, CTL, μ-Calculus, ATL), automata, Petri Nets
- **Computational models**: transition systems, real-time systems, hybrid systems, event-based systems
- **Static analysis methods**: invariant generation, termination analysis
- **Verification methods**: deductive, algorithmic, abstraction, diagrams
- **Synthesis methods**: deductive, algorithmic, game-theoretic, diagrams
- **Computational engines**: decision procedures, quantifier elimination methods, constraint solvers
FORMAL METHODS AND ALL THAT

OUTLINE

1. OVERVIEW

2. SPECIFICATION FORMALISMS
   - Temporal Logics

3. COMPUTATIONAL MODELS
   - Fair Transition System

4. STATIC ANALYSIS METHODS
   - Invariant Generation

5. SYNTHESIS METHODS
   - Game-theoretic Approach

6. COMPUTATIONAL ENGINES
   - Decision Procedures
**Temporal Logics**

- Linear Temporal Logic (LTL)
- Computational Tree Logic (CTL, CTL*)
- Alternating Temporal Logic (ATL, ATL*)
- Interval Temporal Logic (ITL)
- $\mu$-Calculus
OUTLINE

1. **Overview**

2. **Specification Formalisms**
   - Temporal Logics

3. **Computational Models**
   - Fair Transition System

4. **Static Analysis Methods**
   - Invariant Generation

5. **Synthesis Methods**
   - Game-theoretic Approach

6. **Computational Engines**
   - Decision Procedures
**Fair Transition System**

\[ \Phi = \langle V, \Theta, \mathcal{T}, \mathcal{J}, \mathcal{C} \rangle \]

- **V**: a finite set of system variables
- **\Theta**: initial condition expressed by a first-order constraints over **V**
- **\mathcal{T}**: a finite set of transitions each of which is expressed by a first order assertion \( \rho_\tau(V, V') \) over **V** and **V**', the **V** in the next state
- **\mathcal{J} \subseteq \mathcal{T}**: just (weakly fair) transitions
- **\mathcal{C} \subseteq \mathcal{T}**: compassionate (strongly fair) transitions
Fair Transition System

Example

\[ \Phi = \langle V, \Theta, \mathcal{I}, \mathcal{J}, \mathcal{C} \rangle \]

- \( V : \{ x : \text{integer} \} \)
- \( \Theta : x > 0 \)
- \( \mathcal{I} = \{ \tau \} \)
- \( \mathcal{J} = \{ \tau \} \)
- \( \mathcal{C} = \{ \tau \} \)

\[ \tau : (2 \mid x \rightarrow x' = x/2) \land (2 \nmid x \rightarrow x' = 3x + 1) \]

Is it true that \( \Diamond(x = 1) \)?
OUTLINE

1. OVERVIEW
2. SPECIFICATION FORMALISMS
   • Temporal Logics
3. COMPUTATIONAL MODELS
   • Fair Transition System
4. STATIC ANALYSIS METHODS
   • Invariant Generation
5. SYNTHESIS METHODS
   • Game-theoretic Approach
6. COMPUTATIONAL ENGINES
   • Decision Procedures
Assertion $\varphi$ is an invariant of $P$ if and only if it is true at all the reachable states of $P$. 

Symbolically execute the program using forward propagation until fix-point is reached.

Forward propagation:
V: the variables in the current state
V': the corresponding variables in the next state.
Transition $\tau$: $\rho_\tau(V, V')$.
Post-condition $post(\tau, \varphi): (\exists V^0) \ (\rho_\tau(V^0, V) \land \varphi(V^0))$

Are we there yet?
- Convergence will never be reached
- Convergence cannot be detected
Invariant Generation
Deductive Analysis

$$\Theta(V) \rightarrow \varphi(V) \quad \text{(Initiation)}$$
$$\rho_T(V, V') \land \varphi(V) \rightarrow \varphi(V') \quad \text{(Consecution)}$$

$$\Rightarrow \square \varphi(V)$$

Who makes the guess?
Formal Methods and All That

Synthesis Methods

Game-theoretic Approach

Outline

1. Overview
2. Specification Formalisms
   - Temporal Logics
3. Computational Models
   - Fair Transition System
4. Static Analysis Methods
   - Invariant Generation
5. Synthesis Methods
   - Game-theoretic Approach
6. Computational Engines
   - Decision Procedures
**Formal Methods and All That**

---

**Synthesis Methods**

---

**Game-theoretic Approach**

---

## Reactive Systems

**Single player**

- Does this property hold of all computations?
- Models: Transition systems
- Logics: LTL, CTL, CTL*

**Multiplayer**

- Can agents $a$ and $b$ guarantee that this property will hold no matter what players $c$ and $d$ do?
- Models: Alternating transition systems
- Logics: ATL, ATL*

\[
\langle a \rangle (pU\langle a, b \rangle \Diamond q) \quad \text{ATL}
\]

\[
\langle a \rangle (\Box \Diamond p \rightarrow \Box \Diamond \langle b \rangle q) \quad \text{ATL}^*
\]
Reactive Systems

- Single player
  - Does this property hold of all computations?
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- Multiplayer
  - Can agents $a$ and $b$ guarantee that this property will hold no matter what players $c$ and $d$ do?
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\begin{align*}
\langle a \rangle (p \mathcal{U} \langle a, b \rangle \diamond q) & \quad \text{ATL} \\
\langle a \rangle (\Box \diamond p \rightarrow \Box \diamond \langle b \rangle q) & \quad \text{ATL}^* 
\end{align*}
\]
**Reactive Systems**

- **Single player**
  - *Does this property hold of all computations?*
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- **Multiplayer**
  - *Can agents a and b guarantee that this property will hold no matter what players c and d do?*
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\langle a \rangle (p U \langle a, b \rangle \diamond q) \quad \text{ATL}
\]
\[
\langle a \rangle (\Box \diamond p \rightarrow \Box \diamond \langle b \rangle q) \quad \text{ATL}^*
\]
ReactiVe Systems

- Single player
  - *Does this property hold of all computations?*
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- Multiplayer
  - *Can agents $a$ and $b$ guarantee that this property will hold no matter what players $c$ and $d$ do?*
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\langle \langle a \rangle \rangle (p \mathcal{U} \langle \langle a, b \rangle \rangle \lozenge q) \quad \text{ATL}
\]

\[
\langle \langle a \rangle \rangle (\blacksquare \lozenge p \rightarrow \blacksquare \lozenge \langle \langle b \rangle \rangle q) \quad \text{ATL}^*
\]
Reactive Systems

- Single player
  - *Does this property hold of all computations?*
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- Multiplayer
  - Can agents \( a \) and \( b \) guarantee that this property will hold no matter what players \( c \) and \( d \) do?
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\langle a \rangle (p \lor \langle a, b \rangle \diamond q) \quad \text{ATL}
\]
\[
\langle a \rangle (\Box \diamond p \rightarrow \Box \diamond \langle b \rangle q) \quad \text{ATL}^*
\]
**Reactive Systems**

- **Single player**
  - *Does this property hold of all computations?*
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- **Multiplayer**
  - *Can agents a and b guarantee that this property will hold no matter what players c and d do?*
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\langle a \rangle (p \mathbin{\mathbf{U}} \langle a, b \rangle \mathbin{\Diamond} q) \quad \text{ATL} \\
\langle a \rangle (\square \mathbin{\mathbf{p}} \rightarrow \square \langle b \rangle q) \quad \text{ATL}^*
\]
**Reactive Systems**

- **Single player**
  - *Does this property hold of all computations?*
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- **Multiplayer**
  - *Can agents $a$ and $b$ guarantee that this property will hold no matter what players $c$ and $d$ do?*
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\langle a \rangle (p \mathbin{U} \langle a, b \rangle \lozenge q) \quad \text{ATL}
\]

\[
\langle a \rangle (\square \lozenge p \rightarrow \square \lozenge \langle b \rangle q) \quad \text{ATL*}
\]
**Reactive Systems**

- **Single player**
  - *Does this property hold of all computations?*
  - Models: Transition systems
  - Logics: LTL, CTL, CTL*

- **Multiplayer**
  - *Can agents a and b guarantee that this property will hold no matter what players c and d do?*
  - Models: Alternating transition systems
  - Logics: ATL, ATL*

\[
\langle a \rangle(pU\langle a, b \rangle\diamond q) \quad \text{ATL}
\]

\[
\langle a \rangle(\Box \diamond p \rightarrow \Box \diamond \langle b \rangle q) \quad \text{ATL}^*
\]
Alternating Temporal Logic (atl*)

Extend CTL* with game-theoretic path quantifiers

- □ϕ ("ϕ always in the future")
- ◊ϕ ("ϕ eventually in the future")
- ... other LTL operators
- ⟨⟨a⟩⟩ϕ ≡ “Player a has a strategy to ensure ϕ”
- [a]ϕ ≡ “Player a cannot avoid ϕ”

Example:

⟨⟨a⟩⟩◊(x = -1)
□(p → ⟨⟨a, b⟩⟩◊q)
**Alternating System**

- **board**: \( \mathbb{Z}^2 \rightarrow \{ \square, \bullet, \circ \} \)
- **Local strategies**:
  - Player \( \bullet \): \((x, y) : \mathbb{Z}^2 \) \( \Rightarrow \) \( \text{board}[x, y] = \square \)
  - Player \( \circ \): \((x, y) : \mathbb{Z}^2 \) \( \Rightarrow \) \( \text{board}[x, y] = \bullet \)
- **Local game**:

\[
(x = x \land y = y \land \text{board}' = \text{board}) \lor \\
((x \neq x \lor y \neq y) \land \\
\text{board}' = \text{board}\{(x, y) \leftrightarrow \bullet, (x, y) \leftrightarrow \circ\})
\]
ATL* INFORMALLY

\[ \langle \Box \rangle \ldots = \text{"Player } \Box \text{ has a strategy to achieve } \ldots \text{"} \]

Example:

\[ \langle \Box \rangle \Diamond \exists \text{ a line of } \Box \text{'s of length 5} \]

\[ \langle \Box \rangle \left[ \neg \exists \text{ a line of } \Box \text{'s of length 5} \cup \right. \]

\[ \left. \left( \exists \text{ a line of } \Box \text{'s of length 5} \land \neg \exists \text{ a line of } \Box \text{'s of length 5} \right) \right] \]

OUTLINE

1. OVERVIEW

2. SPECIFICATION FORMALISMS
   - Temporal Logics

3. COMPUTATIONAL MODELS
   - Fair Transition System

4. STATIC ANALYSIS METHODS
   - Invariant Generation

5. SYNTHESIS METHODS
   - Game-theoretic Approach

6. COMPUTATIONAL ENGINES
   - Decision Procedures
DECISION PROCEDURES
Decision Procedures
Decision Procedures

Decision Procedure

Input $\varphi$

Output satisfiable

Output unsatisfiable
Decision procedures exist for specific theories

- Arithmetic: integers, reals, . . . ,
- Data types: lists, queues, arrays, sets, multisets, . . . ,
- Algebraic structures: linear dense orders . . . ,

But

- programming languages involve multiple theories
- verification conditions do not belong to a single theory

Need to combine decision procedures for different theories
**Milestone Decision Procedures**

- **Presburger arithmetic** \( \mathbb{PA} = \langle \mathbb{N}, 0, <, + \rangle \)


- **Tarski arithmetic** \( \mathbb{RA} = \langle \mathbb{R}, 0, <, +, \cdot \rangle \)


- **Theory of Finite trees and its extensions**

  Mal’cev 1971, and many others
MILESTONE DECISION PROCEDURES

- Presburger arithmetic \( \mathbb{PA} = \langle \mathbb{N}, 0, <, + \rangle \)


- Tarski arithmetic \( \mathbb{RA} = \langle \mathbb{R}, 0, <, +, \cdot \rangle \)


- Theory of Finite trees and its extensions
  Mal'cev 1971, and many others
FORMAL METHODS AND ALL THAT

COMPUTATIONAL ENGINES

Decison Procedures

MILESTONE DECISION PROCEDURES

- Presburger arithmetic $\text{PA} = \langle \mathbb{N}, 0, <, + \rangle$
  

- Tarski arithmetic $\text{RA} = \langle \mathbb{R}, 0, <, +, \cdot \rangle$
  

- Theory of Finite trees and its extensions
  
  Mal’cev 1971, and many others