Formal Methods and All That

Ting Zhang

Theory Group
Microsoft Research Asia

August 24, 2007

Outline

Outline

Outline

Outline

Outline

Outline

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

- 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

- 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

- 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

- 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

- 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

- 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

Outline

Temporal Logics

- Linear Temporal Logic (LTL)
- Computational Tree Logic (CTL, CTL*)
- Alternating Temporal Logic (ATL, ATL*)
- Interval Temporal Logic (ITL)
- μ-Calculus

Outline

Fair Transition System

$\Phi = \langle V, \Theta, \mathscr{T}, \mathscr{J}, \mathscr{C} \rangle$

- V: a finite set of system variables
- Θ: initial condition expressed by a first-order constraints over V
- $\mathscr{T}$: 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
- $\mathscr{J} \subseteq \mathscr{T}$: just (weakly fair) transitions
- $\mathscr{C} \subseteq \mathscr{T}$: compassionate (strongly fair) transitions

Fair Transition System

Example

$\Phi = \langle V, \Theta, \mathscr{T}, \mathscr{J}, \mathscr{C} \rangle$

- $V : \{x : \text{integer}\}$
- Θ: $x > 0$
- $\mathscr{T} = \{\tau\}$

$$\tau : (2 \mid x \rightarrow x' = x/2) \ \wedge \ (2 \nmid x \rightarrow x' = 3x + 1)$$

- $\mathscr{J} = \{\tau\}$
- $\mathscr{C} = \{\tau\}$

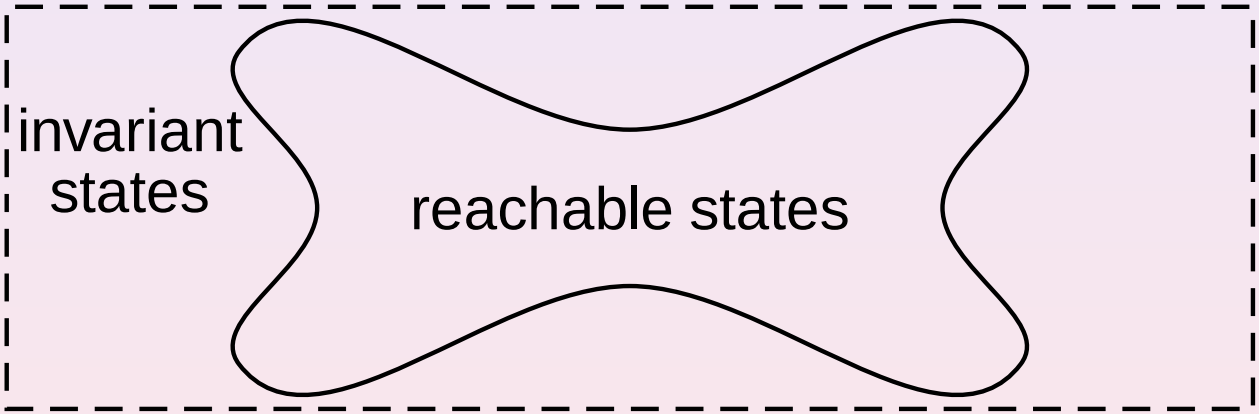
Is it true that $\Diamond(x = 1)$?

Outline

Invariant

[Floyd, Hoare : 1967,1969]

Assertion φ is an *invariant* of P if and only if it is true at all the *reachable states* of P.



Invariant Generation

Fix-point computation

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 τ: $\rho_\tau(V, V')$.
Post-condition $post(\tau, \varphi)$: $(\exists V^0)\ (\ \rho_\tau(V^0, V)\ \wedge\ \varphi(V^0)\ )$

Are we there yet?

- Convergence will never be reached
- Convergence cannot be detected

Invariant Generation

Deductive Analysis

$$\Theta(V) \rightarrow \varphi(V) \quad \text{(Initiation)}$$
$$\rho_\tau(V, V') \wedge \varphi(V) \rightarrow \varphi(V') \quad \text{(Consecution)}$$

$$\Rightarrow \Box\varphi(V)$$

Who makes the guess?

Outline

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box\Diamond p \rightarrow \Box\Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL*}$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box\Diamond p \rightarrow \Box\Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL*}$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box\Diamond p \rightarrow \Box\Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL*}$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box\Diamond p \rightarrow \Box\Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL*}$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box\Diamond p \rightarrow \Box\Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL}^*$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box \Diamond p \rightarrow \Box \Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL}^*$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box \Diamond p \rightarrow \Box \Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL}^*$$

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*

$$\langle\langle a \rangle\rangle (p \mathsf{U} \langle\langle a, b \rangle\rangle \Diamond q) \qquad \text{ATL}$$

$$\langle\langle a \rangle\rangle (\Box\Diamond p \rightarrow \Box\Diamond \langle\langle b \rangle\rangle q) \qquad \text{ATL*}$$

Alternating Temporal Logic (ATL*)

[Alur-Henzinger-Kupferman : 1997]

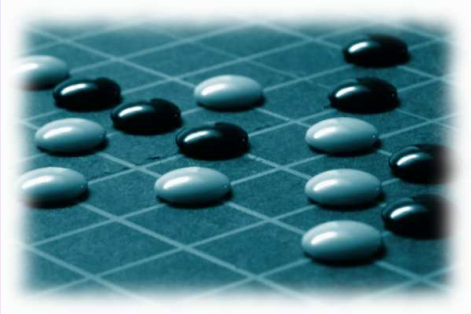
Extend CTL* with game-theoretic path quantifiers

- $\Box\varphi$ ("φ always in the future")
- $\Diamond\varphi$ ("φ eventually in the future")
- ... other LTL operators
- $\langle\langle a\rangle\rangle\varphi \equiv$ "Player a has a strategy to ensure φ"
- $[\![a]\!]\varphi \equiv$ "Player a cannot avoid φ"

Example:

$$\langle\langle a\rangle\rangle\Diamond(x = -1)$$
$$\Box(p \rightarrow \langle\langle a, b\rangle\rangle\Diamond q)$$

Alternating System



- $board : \mathbb{Z}^2 \to \{$ ⬛, ⚫, ⚪ $\}$
- Local strategies:
  - Player ⚫: $(x_{⚫}, y_{⚫}) : \mathbb{Z}^2 \qquad board[x_{⚫}, y_{⚫}] =$ ⬛
  - Player ⚪: $(x_{⚪}, y_{⚪}) : \mathbb{Z}^2 \qquad board[x_{⚪}, y_{⚪}] =$ ⬛
- Local game:

$$
\begin{aligned}
&(x_{⚫} = x_{⚪} \wedge y_{⚫} = y_{⚪} \wedge board' = board) \vee \\
&\quad \big((x_{⚫} \neq x_{⚪} \vee y_{⚫} \neq y_{⚪}) \wedge \\
&\qquad board' = board\{(x_{⚫}, y_{⚫}) \mapsto ⚫, (x_{⚪}, y_{⚪}) \mapsto ⚪\}\big)
\end{aligned}
$$

ATL* Informally

$\langle\langle$ ● $\rangle\rangle \ldots =$ "*Player* ● *has a strategy to achieve* ..."

Example:

$\langle\langle$ ● $\rangle\rangle \Diamond$ "$\exists$ *a line of* ●*'s of length* 5"

$\langle\langle$ ● $\rangle\rangle \Big[$"$\nexists$ *a line of* ○*'s of length* 5" U

$\big($"$\exists$ *a line of* ●*'s of length* 5"$\wedge$

"$\nexists$ *a line of* ○*'s of length* 5"$\big)\Big]$

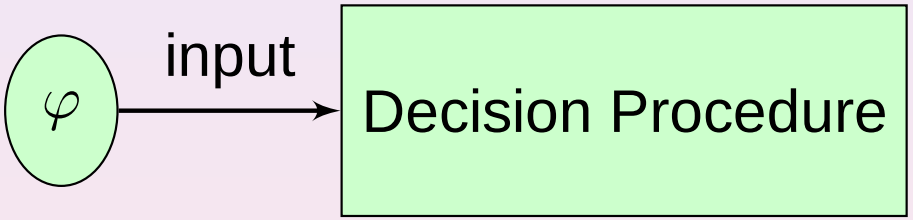
M. Slanina and H.B. Sipma and Z. Manna "Proving ATL* Properties of Infinite-State Systems", Proc. *Theoretical Aspects of Computing (06)*.

Outline

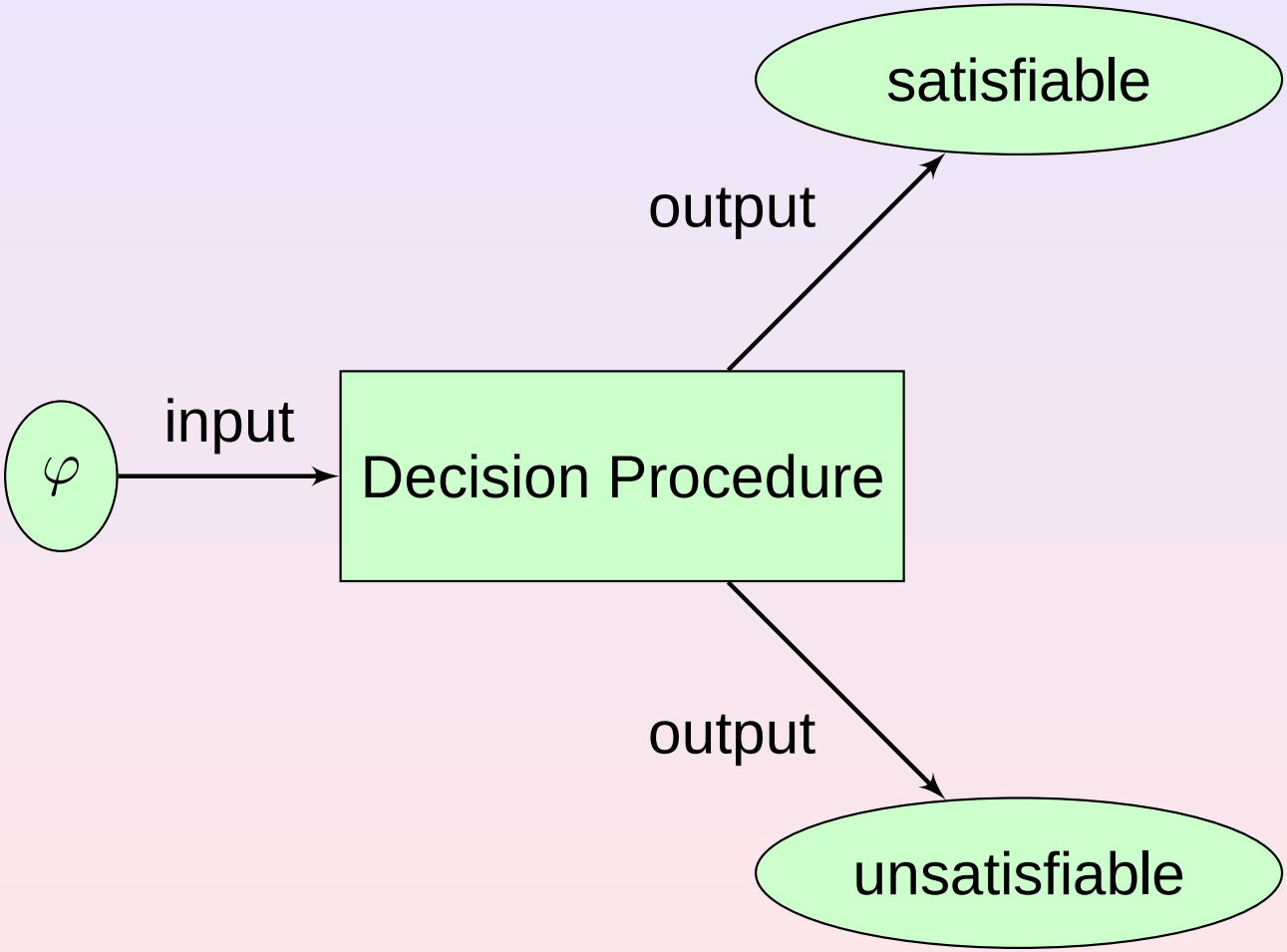
Decision Procedures

φ

Decision Procedures



Decision Procedures



Why Do We Need New Decision Procedures?

Decision procedures exist for specific theories

- Arithmetic: integers, reals, . . . ,
- Data types: lists, queues, arrays, sets, multisets, . . . ,
- Algebriac 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 PA $= \langle \mathbb{N}, 0, <, + \rangle$

  Presburger 1929, Cooper 1972, Reddy-Loveland 1978, Fischer-Rabin 1974

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

  Tarski 1950, Collins 1975, Weispfenning 1992 QEPCAD [Hong 1993], REDLOG [Dolzmann-Sturm 1997]

- Theory of Finite trees and its extensions

  Mal'cev 1971, and many others

Milestone Decision Procedures

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

  Presburger 1929, Cooper 1972, Reddy-Loveland 1978, Fischer-Rabin 1974

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

  Tarski 1950, Collins 1975, Weispfenning 1992 QEPCAD [Hong 1993], REDLOG [Dolzmann-Sturm 1997]

- Theory of Finite trees and its extensions

  Mal'cev 1971, and many others

Milestone Decision Procedures

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

  Presburger 1929, Cooper 1972, Reddy-Loveland 1978, Fischer-Rabin 1974

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

  Tarski 1950, Collins 1975, Weispfenning 1992 QEPCAD [Hong 1993], REDLOG [Dolzmann-Sturm 1997]

- Theory of Finite trees and its extensions

  Mal'cev 1971, and many others