Rabin Theory and Game Automata
An Introduction

Ting Zhang

Stanford University

November 2002

Outline

1. Monadic second-order theory of two successors (S2S)
2. Rabin Automata
3. Game Automata
4. Equivalence of Rabin and Game Automata
5. Forgetful Determinacy
6. Complementation of Game Automata
7. Decidability of S2S

Monadic Second-Order Logic

Monadic second-order language is formulated by the following rules.

1. All rules of forming first-order languages.
2. For any monadic predicate P, Px is an atomic formula.
3. If φ is a formula and P is a monadic predicate, then $\forall P \varphi$ is a formula.

A **universal monadic second-order** formula is a formula of the form

$$\forall P_1 \ldots \forall P_n \varphi(P_1, \ldots P_n)$$

where $\varphi(P_1, \ldots P_n)$ contains no second-order quantifiers.

Decidability of Monadic Second-Order Logics I

A decidable class?	Universal monadic second-order logic
All flows of time	no
Reals	yes (Burgess, Gurevich)
Dedekind complete flows	yes (Gurevich)
Continuous flows	yes (Gurevich)
Circles	yes (Reynolds)
Dense Flows	yes (Gurevich)
Discrete Flows	yes (Gurevich)
All linear flows	yes (Gurevich)
Rationals	yes
Finite linear flows	yes
Natural numbers, integers	yes
Well-ordered flows	yes (Rabin)

Decidability of Monadic Second-Order Logics II

A decidable class?	Full monadic second-order logic
All flows of time	no
Reals	no (Shelah, Gurevich)
Dedekind complete flows	no
Continuous flows	no
Circles	no
Dense Flows	no
Discrete Flows	no
All linear flows	no (Shelah, Gurevich)
Rationals	yes (Rabin)
Finite linear flows	yes (Büchi)
Natural numbers, integers	yes (Büchi)
Well-ordered flows	no

Binary Tree and S2S

- **Binary tree**: $\mathcal{T} = \{0,1\}^*$.
- **$\boldsymbol{\Sigma}$-(valued) tree**: $F : \mathcal{T} \mapsto \Sigma$ assigning every node of the infinite binary tree a letter from Σ.
- **Monadic second-order theory of $\mathbf{2}$ successors ($\mathbf{S2S}$)**: the monadic theory of the structure

$$\mathfrak{T} \stackrel{def}{=} (\mathcal{T}, S_0, S_1)$$

  with successor functions $S_0(u) = u0$ and $S_1(u) = u1$.
- **Weak Monadic second-order theory of $\mathbf{2}$ successors ($\mathbf{WS2S}$)**: S2S with the restriction that second-order quantifiers range over finite subsets of $\{0,1\}^*$.

Examples of S2S

- Prefix relation:

$$x < y \stackrel{def}{=} \forall P((Pw \to P(S_0 w) \vee P(S_1 w)) \wedge Px \to Py)$$

- The countably infinite dense linear order without endpoints:

$$x \prec y \stackrel{def}{=} S_0 y < x \vee S_1 x < y \vee \exists z(S_0 z < x \wedge S_1 z < y)$$

☞ The monadic second-order theory of $(\mathbb{Q}, <)$ is decidable.

Examples of S2S

- Chains.

$$CHAIN(P) \stackrel{def}{=} \forall x \forall y (Px \wedge Py \rightarrow x < y \vee x = y \vee y < x)$$

- Infinite chains.

$$INFCHAIN(P) \stackrel{def}{=} CHAIN(P) \wedge \forall x (Px \rightarrow \exists y (x < y \wedge Py))$$

☞ $S1S$ can be interpreted in $S2S$.

Examples of S2S

- Prefix closure: P is the prefix closure of Q.

$$PRECL(P, Q) \stackrel{def}{=} \forall x(Px \leftrightarrow \exists y(x \leq y \wedge Qy))$$

- Finiteness.

$$FINITE(P) \stackrel{def}{=} \neg\exists Q \exists R(PRECL(Q, P) \\ \wedge R \subseteq Q \wedge INFCHAIN(R))$$

☞ $WS2S$ can be interpreted in $S2S$.

Rabin Automata

A **Rabin Automata** over alphabet Σ is a quadruple $\mathcal{A} = \langle S, S_0, \delta, \mathcal{F} \rangle$:

- S: a finite set of states,
- $S_0 \subseteq S$: a set of initial states,
- $\delta : S \times \Sigma \mapsto \mathcal{P}(S \times S)$: transition table,
- $\mathcal{F} \subseteq \mathcal{P}(S)$: accepting condition.

Run and Acceptance

A **run** $\mathbf{G}$ of a Rabin automaton $\mathcal{A}$ on a Σ-tree F is a S-valued tree

$$\mathbf{G} : T \mapsto S \quad \text{such that}$$

- $\mathbf{G}(\lambda) \in S_0$ and
- for all $u \in \mathcal{T}$, $(\mathbf{G}(u0), \mathbf{G}(u1)) \in \delta(\mathbf{G}(u), F(u))$.

$\mathcal{A}$ **accepts** F if there exists a run G of $\mathcal{A}$ such that for every path π in $\mathcal{T}$,

$$\{s \in S : G(w) = s \text{ for infinitely many } w \in \pi\} \in \mathcal{F}$$

An Example of Rabin Automata

Let $\mathcal{A} = \langle S, S_0, \delta, \mathcal{F} \rangle$ be such that

- $S = \{s, t\}, \quad S_0 = \{s\},$
- $\delta(s, b) = \{(s, t), (t, s)\}, \quad \delta(s, a) = \mathit{false},$
  $\delta(t, a) = \{(t, t)\}, \quad \delta(t, b) = \mathit{false},$
- $\mathcal{F} = \{\{t\}, \{s\}\}.$

☞ The automaton $\mathcal{A}$ accepts all $\{a, b\}$-valued trees with a path whose nodes all are labeled by b while all other nodes not in the path are labeled by a.

Game Automata

A **Game Automata** over alphabet Σ is a quadruple $\mathcal{A} = \langle S, \delta_0, \delta, \mathcal{F} \rangle$:

- S: a finite set of states,
- $\delta_0 : \Sigma \mapsto \mathcal{P}(S)$: initial transition table,
- $\delta : S \times \{0, 1\} \times \Sigma \mapsto \mathcal{P}(S)$: transition table,
- $\mathcal{F} \subseteq \mathcal{P}(S)$: accepting condition.

Game-theoretic View of Run

Game $\Gamma(\mathcal{A}, F)$ between $\mathcal{A}$ and Pathfinder $\mathcal{P}$ on Σ-tree F:

- At the root $\mathcal{A}$ chooses a state $s_0 \in \delta_0(F(\lambda))$.
- At odd step, $\mathcal{P}$ chooses a direction $d \in \{0, 1\}$.
- At even step, $\mathcal{A}$ chooses state s_{n+1} such that

$$s_{n+1} \in \delta(s_n, d_{n+1}, F(d_1 \ldots d_{n+1})).$$

- **Play**: an infinite sequence $\pi = s_0 d_1 s_1 d_2 s_2 \ldots$.
- **Position**: a prefix of a play.

Game-theoretic View of Acceptance

- **Win**: $\mathcal{A}$ wins a play π if the set of states occurring infinitely often in π is contained in $\mathcal{F}$. Otherwise $\mathcal{P}$ wins.
- **Strategy**: a function f assigning each position a set of legal moves to the corresponding player.
- **Winning strategy**: a strategy by which a player wins all plays no matter what the opponent does.
- **Acceptance**: $\mathcal{A}$ accepts F iff $\mathcal{A}$ has a winning strategy for game $\Gamma(\mathcal{A}, F)$.

An Example of Game Automata

Let $\mathcal{A} = \langle S, \delta_0, \delta, \mathcal{F} \rangle$ be such that

- $S = \{s, t\}$,
- $\delta_0(a) = \emptyset$, $\quad \delta_0(b) = \{s\}$,
- $\delta(s, d, a) = \{t\}$, $\quad \delta(t, d, b) = \mathit{false}$, $\qquad (d \in \{0, 1\})$
  $\delta(s, d, b) = \{s\}$, $\quad \delta(t, d, a) = \{t\}$,
- $\mathcal{F} = \{\{t\}, \{s\}\}$.

☞ The automaton $\mathcal{A}$ accepts all $\{a, b\}$-valued trees whose paths are in the form $b^n a^\omega$ for $n > 0$.

Rabin Automata "$\prec$" Game Automata

$\mathcal{A} = \langle S, S_0, \delta, \mathcal{F} \rangle$: a Rabin automaton.

$\mathcal{A}' = \langle S', \delta'_0, \delta', \mathcal{F}' \rangle$: the corresponding game automaton such that

- $S' = S \times S \times S$,
- $\delta'_0(\sigma) = \{(t_0, t_1, t_2) : (t_0, t_1) \in \delta(t_2, \sigma), t_2 \in S_0\}$,
- $\delta((t_0, t_1, t_2), d, \sigma) = \{(t'_0, t'_1, t_d) : (t'_0, t'_1) \in \delta(t_d, \sigma)\}$, $(d \in \{0, 1\})$
- $\mathcal{F}' = \{X \subseteq S \times S \times S : \mathbf{Proj}_3 X \in \mathcal{F}\}$.

☞ Every successful run of $\mathcal{A}$ on a Σ-tree F corresponds to a winning strategy of $\mathcal{A}'$ on the game $\Gamma(\mathcal{A}', F)$ and vice versa.

Game Automata "$\prec$" Rabin Automata

$\mathcal{A} = \langle S, \delta_0, \delta, \mathcal{F} \rangle$: a game automaton.

$\mathcal{A}' = \langle S', S_0' \delta', \mathcal{F}' \rangle$: the corresponding Rabin automaton such that

- $S' = S \times \{0,1\} \cup \{s_0\}$, $S_0' = \{s_0\}$, $\quad$ (s_0 is a new state)

- $\delta'(s_0, \sigma) = \{((s,0),(s,1) : s \in \delta_0(\sigma)\}, \qquad (s \in S, d \in \{0,1\})$
  $\delta'((s,d),\sigma) = \{((s',0),(s',1)) : s' \in \delta(s,d,\sigma)\},$
- $\mathcal{F}' = \{X \subseteq S \times \{0,1\} : \mathbf{Proj}_1 X \in \mathcal{F}\}$.

☞ Every winning strategy of $\mathcal{A}$ on game $\Gamma(\mathcal{A}, F)$ corresponds to a successful run of $\mathcal{A}'$ on Σ-tree F and vice versa.

Game Automata "=" Rabin Automata

- Given a Rabin automaton $\mathcal{A}$, one can effectively construct a game automaton $\mathcal{A}'$ such that $\mathcal{A}$ and $\mathcal{A}'$ accept the same set of Σ-trees.
- Given a game automaton $\mathcal{A}$, one can effectively construct a Rabin automaton $\mathcal{A}'$ such that $\mathcal{A}$ and $\mathcal{A}'$ accept the same set of Σ-trees.

☞ Rabin automata and game automata are equally expressive.

Latest Appearance Record

- LAR(p): the list of states of position p (without repitition) in the order of their latest appearance.
  - If $q = pd$ is an odd position ($d \in \{0, 1\}$),
  $$LAR(q) \stackrel{def}{=} LAR(p).$$
  - If $q = ps$ is an even position ($s \in S$),
  $$LAR(q) \stackrel{def}{=} rs$$
  where r is obtained by removing s from $LAR(p)$.

Forgetful Determinacy

- Node(p): the subsequence of all letters at odd indices of position p, i.e., the node currently being visited at p.
- Residue(p): the subtree rooted at $Node(p)$.

☞ One of the players has a winning strategy f for game $\Gamma(\mathcal{A}, F)$ that satisfies the following condition:

If p and q are positions from which the winner makes moves, and

$$LAR(p) = LAR(q), \quad Residue(p) = Residue(q)$$

then $f(p) = f(q)$.

Encode Pathfinder's Strategy

Automaton $\mathcal{A} = \langle S^a, \delta_0^a, \delta^a, \mathcal{F}^a \rangle$.

- R: the set of all possible LARs.
- $g : \{0,1\}^* \times R \mapsto \{0,1\}$: $\mathcal{P}$'s (deterministic) strategy.
- Δ: the set of all functions $h : R \mapsto \{0,1\}$.
- Δ-tree G:

$$G(w) = \lambda r.g(w, r) \qquad (w \in \{0,1\}^*,\ r \in R)$$

- $(\Sigma \times \Delta)$-tree (F, G): the product of F and G.

Recognize Pathfinder's Strategy

- L(r): the rightmost (last) member of $r \in R$.
- U(r,s): the new LAR obtained from r by removing s (if it occurs there) and appending it to the end of r.
- Automaton $\mathcal{B} = \langle S^b, \delta_0^b, \delta^b, \mathcal{F}^b \rangle$:
  - $S^b = R \cup \{win\}, \quad \delta_0^b((\sigma, h)) = \delta_0^a(\sigma)$,
  - $\delta^b(win, d, (\sigma, h)) = \{win\}, \qquad (\sigma \in \Sigma,\ \ h \in \Delta,\ \ d \in \{0, 1\})$

$$\delta^b(r, d, (\sigma, h)) = \begin{cases} \{win\} & \text{if } h(r) \neq d, \\ \{U(r, s) : s \in \delta(L(r), d, \sigma)\} & \text{if } h(r) = d. \end{cases}$$

  - $\mathcal{F}^b = \{win\} \bigcup \{R_0 \subset R : \{Last(r) : r \in R_0\} \notin \mathcal{F}^a\}$.

Recognize Pathfinder's Strategy

☞ Given an automaton $\mathcal{A}$ and a Σ-tree F, one can effectively construct an automaton $\mathcal{B}$ such that Pathfinder wins $\Gamma(\mathcal{A}, F)$ via a forgetful strategy encoded by Δ-tree G if and only $\mathcal{B}$ wins **all** plays of the game $\Gamma(\mathcal{B}, (F, G))$.

$\Rightarrow$
- $\mathcal{P}$ plays according to strategy G: $\mathcal{B}$ wins the corresponding play of $\Gamma(\mathcal{B}, (F, G))$ as $\mathcal{F}^b$ encodes the "complement" of $\mathcal{F}^a$.
- $\mathcal{P}$ doesn't plays according to strategy G: $\mathcal{B}$ wins outright.

$\Leftarrow$ If $\mathcal{A}$ wins a play with a strategy f against $\mathcal{P}$'s strategy G, $\mathcal{B}$ loses the corresponding play of $\Gamma(\mathcal{B}, (F, G))$ as the final state set of the play is in $\mathcal{F}^a$.

Encode $\mathcal{B}$'s Behavior via Büchi Automata

An automaton $\mathcal{B} = \langle S^b, \delta_0^b, \delta^b, \mathcal{F}^b \rangle$ wins all plays of $\Gamma(\mathcal{B}, H)$ iff

each path $d_0 d_1 \ldots$ and the corresponding Σ-path $H(d_0)H(d_0 d_1)\ldots$ satisfy the following condition: ($d_0 \equiv \lambda$)

(*) For all infinite sequences $s_0 s_1 \ldots$ if $s_0 \in \delta(H(\lambda))$ and $s_{n+1} \in \delta(s_n, d_{n+1}, H(d_0 \ldots d_{n+1}))$, $n \geq 0$, then the collection of states that occur infinitely often in $s_0 s_1 \ldots$ belongs to $\mathcal{F}$.

☞ There exists a Büchi automaton $\mathcal{C} = \langle S^c, S_0^c, \delta^c, \mathcal{F}^c \rangle$ over alphabet $\{0, 1\} \times \Sigma$ such that $\mathcal{C}$ accepts $(d_0, H(d_0))(d_1, H(d_0 d_1))\ldots$ if and only if $d_0 d_1 \ldots$ and $H(d_0)H(d_0 d_1)\ldots$ satisfy (*).

Encode $\mathcal{B}$'s Behavior via Game Automata

Automaton $\mathcal{D} = \langle S^d, \delta_0^d, \delta^d, \mathcal{F}^d \rangle$ on Σ-tree:

- $S^d = S^c$,
- $\delta_0^d(\sigma) = \bigcup_{s \in S_0^c} \bigcup_{d \in \{0,1\}} \delta^c(s, (d, \sigma))$
- $\delta^d(s, d, \sigma) = \delta^c(s, (d, \sigma))$
- $\mathcal{F}^d = \{X \subseteq S^c : X \cap \mathcal{F}^c \neq \emptyset\}$.

☞ $\mathcal{D}$ accepts a Σ-tree H if and only if $\mathcal{B}$ wins all plays of $\Gamma(\mathcal{B}, H)$.

Simulate $\mathcal{D}$

Automaton $\mathcal{E} = \langle S^e, \delta_0^e, \delta^e, \mathcal{F}^e \rangle$ simulate $\mathcal{D}$ on Σ-tree:

- $S^e = S^d$,
- $\delta_0^e(\sigma) = \bigcup_{h \in \Delta} \delta_0^d((\sigma, h))$,
- $\delta^e(s, d, \sigma) = \bigcup_{h \in \Delta} \delta^d(s, d, (\sigma, h))$,
- $\mathcal{F}^e = \{X \subseteq \mathbf{Proj}_1(S^d) : \exists Y X = \mathbf{Proj}_1 Y \wedge Y \in \mathcal{F}^d\}$.

☞ For every Σ-tree F $\mathcal{E}$ guesses a Δ-tree G, an encoding of Pathfinder's winning strategy, and then simulates $\mathcal{D}$ on the (Σ, Δ)-tree (F, G).

Complementation of Automaton $\mathcal{A}$

Given automaton $\mathcal{A} = \langle S, \delta_0, \delta, \mathcal{F} \rangle$, construct

1. Automaton $\mathcal{B}$: $\mathcal{B}$ wins all plays on a $\Sigma \times \Delta$-tree (F, G).
2. Automaton $\mathcal{D}$: $\mathcal{D}$ accepts (F, G) if and only if $\mathcal{B}$ wins all plays on (F, G).
3. Automaton $\mathcal{E}$: $\mathcal{E}$ accepts a Σ-tree F if and only if there exists Δ-tree G such that $\mathcal{D}$ accepts (F, G).

☞ $\mathcal{E}$ accepts F if and only if Pathfinder has a winning strategy G if and only if $\mathcal{A}$ rejects F. In other words, $\mathscr{L}(\mathcal{E}) = \overline{\mathscr{L}(\mathcal{A})}$.

Reformulate $S2S$

- Introduce purely second-order signature $Succ_0$, $Succ_1$ and $Sing$.

$$
\begin{aligned}
Y = Succ_i X &\stackrel{int}{=} Y = S_i(X) && (i \in \{0,1\}) \\
Sing(X) &\stackrel{int}{=} \exists! x \in X
\end{aligned}
$$

- Transform formulas.

$$
\begin{aligned}
v &\stackrel{\rho}{\mapsto} X_v && (X_v \text{ is a second-order fresh variable}) \\
Pt &\stackrel{\rho}{\mapsto} t^\rho \subseteq P \\
S_i t &\stackrel{\rho}{\mapsto} Succ_i t^\rho && (i \in \{0,1\}) \\
\forall v \varphi(v) &\stackrel{\rho}{\mapsto} \forall X_v (Sing(X_v) \rightarrow \varphi^\rho(X_v)) \\
\psi(v_1 \ldots v_n) &\stackrel{\mu}{\mapsto} \psi^\rho(X_{v_1}, \ldots X_{v_n}) \wedge \textstyle\bigwedge_i Sing(X_{v_i})
\end{aligned}
$$

☞ This reformulation turns S2S into a formally first-order theory.

Link S2S with Game Automata

Map valuations of $V_1, \ldots, V_n$ to $\{0,1\}^n$-tree $\mathfrak{T}(V_1, \ldots, V_n)$

$$\mathfrak{T}(u) = C_{V_1}(u), \ldots, C_{V_n}(u)$$

where $u \in \{0,1\}$ and C_{V_i} is characteristic function of V_i.

- ☞ There is a 1-1 correspondence between automaton and formula constructions.
- ☞ For every $S2S$-formula $\varphi(X_1, \ldots, X_n)$ there exists $\{0,1\}^n$-tree automaton $\mathcal{A}_\varphi$ such that for every $V_1, \ldots, V_n \in \{0,1\}^*$

$$\mathfrak{T} \models \varphi[V_1, \ldots, V_n] \Leftrightarrow \mathfrak{T}(V_1, \ldots, V_n) \in \mathscr{L}(\mathcal{A}_\varphi)$$

Emptiness Problem for Game Automata

- Let $\mathcal{A} = \langle S, \delta_0, \delta, \mathcal{F} \rangle$ be a game automaton.
- Let $\mathcal{B} = \langle S, \delta_0', \delta', \mathcal{F} \rangle$ be such that

$$\delta_0'(0) = \bigcup_{\sigma \in \Sigma} \delta_0(\sigma),$$

$$\delta'(s, d, 0) = \bigcup_{\sigma \in \Sigma} \delta(s, d, \sigma).$$

☞ The automaton $\mathcal{A}$ accepts some Σ-tree if and only if the automaton $\mathcal{B}$ accepts the unique $\{0\}$-tree T.

Check Emptiness

- There are finite number of strategies for $\mathcal{B}$ and Pathfinder $\mathcal{P}$.

$$f_1, \ldots, f_m \text{ for } \mathcal{B}, \quad g_1, \ldots, g_n \text{ for } \mathcal{P}$$

- Check each f_i against each g_j on game $\Gamma(\mathcal{B}, T)$.
  - Each play consistent with (f_i, g_j) stabilize, i.e., the play becomes periodic eventually.
  - There exists a depth limit after which every play stabilizes.
- $\mathcal{B}$ has a winning strategy f if $\mathcal{B}$ wins via some f_i against all $g_j's$.

Decidability of S2S

- It is effective to decide whether $\mathcal{B}$ accepts the unique $\{0\}$-tree.
- It is effective to decide whether an automaton $\mathcal{A}$ accepts a non-empty language.
- A formula φ is a $S2S$ theorem if and only if $\neg\varphi$ is unsatisfiable .
- A formula ψ is unsatisfiable if and only if the language of $\mathcal{A}_\psi$ is empty.
- $S2S$ is decidable.