Delay Games with WMSO+U Winning Conditions

Martin Zimmermann

Saarland University

March 6th, 2015

AVACS Meeting, Freiburg, Germany

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two extensions:
  - Type of interaction: one player may delay her moves.
  - Type of winning conditions: quantitative instead of qualitative.

Introduction

Büchi-Landweber: The winner of a zero-sum two-player game of infinite duration with ω-regular winning condition can be determined effectively.

- Many possible extensions: non-zero-sum, $n > 2$ players, type of winning condition, concurrency, imperfect information, etc.
- We consider two extensions:
  - Type of interaction: one player may delay her moves.
  - Type of winning conditions: quantitative instead of qualitative.

Weak MSO with the unbounding quantifier:

- quantitative extension of (weak) MSO
- able to express many high-level quantitative specification languages, e.g., parameterized LTL, finitary parity conditions, etc.

Outline

1. **WMSO with the Unbounding Quantifier**
2. Delay Games
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. Constant Lookahead is not Sufficient
5. Conclusion

Monadic Second-order Logic

- Monadic Second-order Logic (MSO)
  - Existential/universal quantification of elements: $\exists x$, $\forall x$.
  - Existential/universal quantification of sets: $\exists X$, $\forall X$.
  - Unary predicates P_a for every $a \in \Sigma$.
  - Order relation $<$ and successor relation S.

Monadic Second-order Logic

- Monadic Second-order Logic (MSO)
  - Existential/universal quantification of elements: $\exists x$, $\forall x$.
  - Existential/universal quantification of sets: $\exists X$, $\forall X$.
  - Unary predicates P_a for every $a \in \Sigma$.
  - Order relation $<$ and successor relation S.
- weak MSO (WMSO)
  - Restrict second-order quantifiers to finite sets.

Monadic Second-order Logic

- Monadic Second-order Logic (MSO)
  - Existential/universal quantification of elements: $\exists x$, $\forall x$.
  - Existential/universal quantification of sets: $\exists X$, $\forall X$.
  - Unary predicates P_a for every $a \in \Sigma$.
  - Order relation $<$ and successor relation S.
- weak MSO (WMSO)
  - Restrict second-order quantifiers to finite sets.

Theorem (Büchi '62)

The following are (effectively) equivalent:

1. *L MSO-definable.*
2. *L WMSO-definable.*
3. *L recognized by Büchi automaton.*

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

$L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

$$L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$$

L defined by

$$\forall x \exists y(y > x \land P_b(y)) \land$$
$$UX\ \ [\forall x \forall y \forall z(x < y < z \land x \in X \land z \in X \rightarrow y \in X)$$
$$\land \forall x(x \in X \rightarrow P_a(x))\ ]$$

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

  $L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$

Decidability is a delicate issue:

Theorem (Bojańczyk et al. '14)

There is no algorithm that decides MSO+U on infinite trees and has a correctness proof using the axioms of ZFC.

The Unbounding Quantifier

Bojańczyk: Let's add a new quantifier

- $UX\varphi(X)$ holds, if there are arbitrarily large finite sets X such that $\varphi(X)$ holds.

$$L = \{a^{n_0}ba^{n_1}ba^{n_2}b\cdots \mid \limsup_i n_i = \infty\}$$

Decidability is a delicate issue:

Theorem (Bojańczyk et al. '14)

There is no algorithm that decides MSO+U on infinite trees and has a correctness proof using the axioms of ZFC.

Theorem (Bojańczyk et al. '15)

MSO+U on infinite words is undecidable.

WMSO+U

Restricting the second-order quantifiers saves the day:

Theorem (Bojańczyk '09)

WMSO+U over infinite words is decidable.

WMSO+U

Restricting the second-order quantifiers saves the day:

Theorem (Bojańczyk '09)

WMSO+U over infinite words is decidable.

Theorem (Bojańczyk, Torunczyk '12)

WMSO+U over infinite trees is decidable.

WMSO+U

Restricting the second-order quantifiers saves the day:

Theorem (Bojańczyk '09)

WMSO+U over infinite words is decidable.

Theorem (Bojańczyk, Torunczyk '12)

WMSO+U over infinite trees is decidable.

Theorem (Bojańczyk '14)

WMSO+U with path quantifiers over infinite trees is decidable.

WMSO+U

Restricting the second-order quantifiers saves the day:

Theorem (Bojańczyk '09)

WMSO+U over infinite words is decidable.

Theorem (Bojańczyk, Torunczyk '12)

WMSO+U over infinite trees is decidable.

Theorem (Bojańczyk '14)

WMSO+U with path quantifiers over infinite trees is decidable.

Corollary

Games with WMSO+U winning conditions are decidable.

Max-Automata

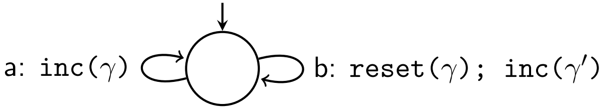
Equivalent automaton model for WMSO+U on infinite words:

- Deterministic finite automata with counters
- counter actions: `incr`, `reset`, `max`
- acceptance: boolean combination of "counter γ is bounded".

Max-Automata

Equivalent automaton model for WMSO+U on infinite words:

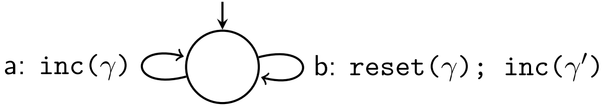
- Deterministic finite automata with counters
- counter actions: `incr`, `reset`, `max`
- acceptance: boolean combination of "counter γ is bounded".



Max-Automata

Equivalent automaton model for WMSO+U on infinite words:

- Deterministic finite automata with counters
- counter actions: `incr`, `reset`, `max`
- acceptance: boolean combination of "counter γ is bounded".



Theorem (Bojańczyk '09)

The following are (effectively) equivalent:

1. *L WMSO+U-definable.*
2. *L recognized by max-automaton.*

Outline

1. WMSO with the Unbounding Quantifier
2. **Delay Games**
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. Constant Lookahead is not Sufficient
5. Conclusion

Delay Games

The delay game $\Gamma_f(L)$:

- Delay function: $f : \mathbb{N} \rightarrow \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).

Delay Games

The delay game $\Gamma_f(L)$:

- Delay function: $f\colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).

Delay Games

The delay game $\Gamma_f(L)$:

- Delay function: $f : \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Delay Games

The delay game $\Gamma_f(L)$:

- Delay function: $f\colon \mathbb{N} \to \mathbb{N}_+$.
- ω-language $L \subseteq (\Sigma_I \times \Sigma_O)^\omega$.
- Two players: Input (I) vs. Output (O).
- In round i:
  - I picks word $u_i \in \Sigma_I^{f(i)}$ (building $\alpha = u_0 u_1 \cdots$).
  - O picks letter $v_i \in \Sigma_O$ (building $\beta = v_0 v_1 \cdots$).
- O wins iff $\binom{\alpha(0)}{\beta(0)}\binom{\alpha(1)}{\beta(1)} \cdots \in L$.

Definition: f is constant, if $f(i) = 1$ for every $i > 0$.

Outline

1. WMSO with the Unbounding Quantifier
2. Delay Games
3. **WMSO+U Delay Games w.r.t. Constant Lookahead**
4. Constant Lookahead is not Sufficient
5. Conclusion

Determinacy

Theorem (Z. '14)

Delay Games with WMSO+U winning conditions w.r.t fixed delay functions are determined.

Determinacy

Theorem (Z. '14)

Delay Games with WMSO+U winning conditions w.r.t fixed delay functions are determined.

Proof idea:

- Winning condition recognized by some automaton $\mathcal{A}$.
- Encode game as parity game in countable arena. States store:
  - Current lookahead (queue over Σ_I)
  - state $\mathcal{A}$ reaches on current play prefix.
  - Current counter values after this run prefix.
  - Maximal counter values seen thus far.
  - Flag marking whether maximum was increased during last transition.
- Thus: counter γ unbounded if corresponding flag is raised infinitely often $\Rightarrow$ parity condition.

Capturing Finite Runs of Max-Automata

Theorem (Z. '14)

The following problem is decidable: given a max-automaton $\mathcal{A}$, does Player O win $\Gamma_f(L(\mathcal{A}))$ for some constant delay function f.

Capturing Finite Runs of Max-Automata

Theorem (Z. '14)

The following problem is decidable: given a max-automaton $\mathcal{A}$, does Player O win $\Gamma_f(L(\mathcal{A}))$ for some constant delay function f.

Proof Idea:

- Adapt technique for parity automata to max-automata.
- Capture behavior of $\mathcal{A}$, i.e., evolution of counter values:
  - Transfers from counter γ to γ'.
  - Existence of increments, but not how many.
  - $\Rightarrow$ equivalence relation $\equiv$ of exponential index.

Capturing Finite Runs of Max-Automata

Theorem (Z. '14)

The following problem is decidable: given a max-automaton $\mathcal{A}$, does Player O win $\Gamma_f(L(\mathcal{A}))$ for some constant delay function f.

Proof Idea:

- Adapt technique for parity automata to max-automata.
- Capture behavior of $\mathcal{A}$, i.e., evolution of counter values:
  - Transfers from counter γ to γ'.
  - Existence of increments, but not how many.
  - $\Rightarrow$ equivalence relation $\equiv$ of exponential index.

Lemma

Let $(x_i)_{i\in\mathbb{N}}$ and $(x_i')_{i\in\mathbb{N}}$ be two sequences of words over Σ^* with $\sup_i |x_i| < \infty$, $\sup_i |x_i'| < \infty$, and $x_i \equiv x_i'$ for all i. Then, $x = x_0x_1x_2\cdots \in L(\mathcal{A})$ if and only if $x' = x_0'x_1'x_2'\cdots \in L(\mathcal{A})$.

Removing Delay

- Player I picks equivalence classes,
- Player O constructs run on representatives (always one step behind to account for delay).

Removing Delay

- Player I picks equivalence classes,
- Player O constructs run on representatives (always one step behind to account for delay).

Resulting game is delay-free with WMSO+U winning condition.

- Can be solved effectively by a reduction to a satisfiability problem for WMSO+U with path quantifiers over infinite trees.
- Doubly-exponential upper bound on necessary constant lookahead.

Outline

1. WMSO with the Unbounding Quantifier
2. Delay Games
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. **Constant Lookahead is not Sufficient**
5. Conclusion

Constant Lookahead is not Sufficient

- $\Sigma_I = \{0, 1, \#\}$ and $\Sigma_O = \{0, 1, *\}$.
- Input block: $\#w$ with $w \in \{0, 1\}^+$. Length: $|w|$.
- Output block:

$$\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)} \in (\Sigma_I \times \Sigma_O)^+$$

for $\alpha(j) \in \{0, 1\}$. Length: n.

Constant Lookahead is not Sufficient

- $\Sigma_I = \{0, 1, \#\}$ and $\Sigma_O = \{0, 1, *\}$.
- Input block: $\#w$ with $w \in \{0, 1\}^+$. Length: $|w|$.
- Output block:

$$\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)} \in (\Sigma_I \times \Sigma_O)^+$$

for $\alpha(j) \in \{0, 1\}$. Length: n.

Define language L: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Constant Lookahead is not Sufficient

- $\Sigma_I = \{0, 1, \#\}$ and $\Sigma_O = \{0, 1, *\}$.
- Input block: $\#w$ with $w \in \{0, 1\}^+$. Length: $|w|$.
- Output block:

$$\binom{\#}{\alpha(n)}\binom{\alpha(1)}{*}\binom{\alpha(2)}{*}\cdots\binom{\alpha(n-1)}{*}\binom{\alpha(n)}{\alpha(n)} \in (\Sigma_I \times \Sigma_O)^+$$

for $\alpha(j) \in \{0, 1\}$. Length: n.

Define language L: if infinitely many $\#$ and arbitrarily long input blocks, then arbitrarily long output blocks.

Theorem (Z. '14)

I wins $\Gamma_f(L)$, if f is a constant delay function, O if f is unbounded.

Proof Idea

1. Let f be constant:

Proof Idea

1. Let f be constant:

$$I: \quad \# \quad 0 \quad 0 \quad \cdots \quad 0$$

Proof Idea

1. Let f be constant:

$$\begin{array}{rccccc} I\colon & \# & 0 & 0 & \cdots & 0 \\ O\colon & 0 & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{rcccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 \\ O: & 0 & & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{rcccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 \\ O: & 0 & * & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{rcccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 \\ O: & 0 & * & & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{rcccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 \\ O: & 0 & * & * & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{rcccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 & 1\cdots \\ O: & 0 & * & * & & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{lcccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 & 1\cdots \\ O: & 0 & * & * & \cdots & & & & \end{array}$$

Proof Idea

1. Let f be constant:

$$\begin{array}{lcccccccc} I\colon & \# & 0 & 0 & \cdots & 0 & 1 & 1 & 1\cdots \\ O\colon & 0 & * & * & \cdots & & & & \end{array}$$

   - Lookahead contains only input blocks of length $f(0)$.
   - Player I can react to Player O's declaration at beginning of an output block to bound size of output blocks while producing arbitrarily large input blocks.

Proof Idea

1. Let f be constant:

$$\begin{array}{lccccccccc} I: & \# & 0 & 0 & \cdots & 0 & 1 & 1 & 1\cdots \\ O: & 0 & * & * & \cdots & & & & \end{array}$$

- Lookahead contains only input blocks of length $f(0)$.
- Player I can react to Player O's declaration at beginning of an output block to bound size of output blocks while producing arbitrarily large input blocks.

2. Let f be unbounded:

- If Player I produces arbitrarily long input blocks, then the lookahead will contain arbitrarily long input blocks.
- Thus, Player O can produce arbitrarily long output blocks.

Outline

1. WMSO with the Unbounding Quantifier
2. Delay Games
3. WMSO+U Delay Games w.r.t. Constant Lookahead
4. Constant Lookahead is not Sufficient
5. **Conclusion**

Conclusion

- Delay games with WMSO+U winning conditions w.r.t. constant delay functions are decidable.
- But constant delay is not always sufficient for Player O.

Conclusion

- Delay games with WMSO+U winning conditions w.r.t. constant delay functions are decidable.
- But constant delay is not always sufficient for Player O.

Current work:

- Solve games w.r.t. arbitrary delay functions.

Conjecture

The following are equivalent for L definable in WMSO+U:

1. *Player O wins $\Gamma_f(L)$ for some f.*
2. *Player O wins $\Gamma_f(L)$ for every unbounded f.*

Conclusion

- Delay games with WMSO+U winning conditions w.r.t. constant delay functions are decidable.
- But constant delay is not always sufficient for Player O.

Current work:

- Solve games w.r.t. arbitrary delay functions.

Conjecture

The following are equivalent for L definable in WMSO+U:

1. *Player O wins $\Gamma_f(L)$ for some f.*
2. *Player O wins $\Gamma_f(L)$ for every unbounded f.*

- Matching bounds on necessary lookahead for the case of constant delay functions.
- A general determinacy theorem.