Optimal Bounds in Parametric LTL Games

Martin Zimmermann

RWTH Aachen University

June 16th, 2011

GandALF 2011
Minori, Italy

Motivation

LTL as specification language in formal verification. Advantages:

- compact, variable-free syntax,
- intuitive semantics,
- successfully employed in model checking tools.

Motivation

LTL as specification language in formal verification. Advantages:

- compact, variable-free syntax,
- intuitive semantics,
- successfully employed in model checking tools.

However, LTL lacks capabilities to express timing constraints.

There are many extensions of LTL that deal with this. We consider

- Parametric LTL (Alur, Etessami, La Torre, Peled '99)
- Prompt LTL (Kupferman, Piterman, Vardi '07)

Here: infinite games with winning conditions in parametric LTL.

Outline

1. **Introduction**
2. Decision Problems
3. Optimization Problems
4. Conclusion

Parametric LTL

LTL:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \varphi\mathbf{R}\varphi$$

Parametric LTL

PLTL:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{R} \varphi \mid \mathbf{F}_{\leq x}\varphi \mid \mathbf{G}_{\leq y}\varphi$$

where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are variables ranging over $\mathbb{N}$.

Parametric LTL

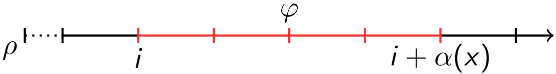
PLTL:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \varphi\mathbf{R}\varphi \mid \mathbf{F}_{\leq x}\varphi \mid \mathbf{G}_{\leq y}\varphi$$

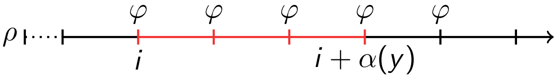
where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are variables ranging over $\mathbb{N}$.

Semantics defined w.r.t. variable valuation $\alpha\colon \mathcal{X} \cup \mathcal{Y} \to \mathbb{N}$:

$(\rho, i, \alpha) \models \mathbf{F}_{\leq x}\varphi$:



$(\rho, i, \alpha) \models \mathbf{G}_{\leq y}\varphi$:



Parametric LTL

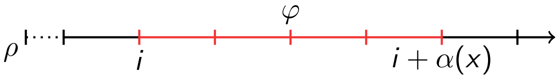
PLTL:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi\mathbf{U}\varphi \mid \varphi\mathbf{R}\varphi \mid \mathbf{F}_{\leq x}\varphi \mid \mathbf{G}_{\leq y}\varphi$$

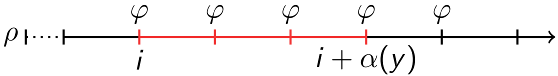
where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are variables ranging over $\mathbb{N}$.

Semantics defined w.r.t. variable valuation $\alpha\colon \mathcal{X} \cup \mathcal{Y} \to \mathbb{N}$:

$(\rho, i, \alpha) \models \mathbf{F}_{\leq x}\varphi$:



$(\rho, i, \alpha) \models \mathbf{G}_{\leq y}\varphi$:

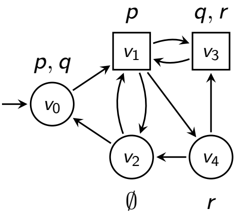


- $\mathrm{PROMPT-LTL}$: $\mathrm{var}(\varphi) = \{x\} \subseteq \mathcal{X}$.
- The operators $\mathbf{U}_{\leq x}$, $\mathbf{R}_{\leq y}$, $\mathbf{F}_{>y}$, $\mathbf{G}_{>x}$, $\mathbf{U}_{>y}$, and $\mathbf{R}_{>x}$ (with the expected semantics) are syntactic sugar, and will be ignored.

Infinite Games

An arena $\mathcal{A} = (V, V_0, V_1, E, v_0, l)$ consists of

- a finite, directed graph (V, E),
- a partition $\{V_0, V_1\}$ of V,
- an initial vertex v_0,
- a labeling $l\colon V \to 2^P$ for some set P of atomic propositions.

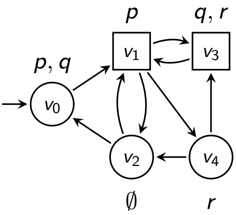


Winning conditions are expressed by a PLTL formula φ over P.

Infinite Games

An arena $\mathcal{A} = (V, V_0, V_1, E, v_0, l)$ consists of

- a finite, directed graph (V, E),
- a partition $\{V_0, V_1\}$ of V,
- an initial vertex v_0,
- a labeling $l \colon V \to 2^P$ for some set P of atomic propositions.



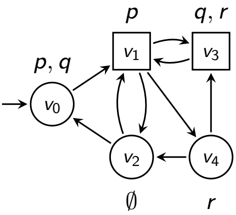
Winning conditions are expressed by a PLTL formula φ over P.

- Play: path $\rho_0\rho_1\rho_2\ldots$ through (V, E) starting in v_0.
- $\rho_0\rho_1\rho_2\ldots$ winning for Player 0 w.r.t. variable valuation α: $(\rho_0\rho_1\rho_2\ldots, 0, \alpha) \models \varphi$. Otherwise winning for Player 1.

Infinite Games

An arena $\mathcal{A} = (V, V_0, V_1, E, v_0, l)$ consists of

- a finite, directed graph (V, E),
- a partition $\{V_0, V_1\}$ of V,
- an initial vertex v_0,
- a labeling $l \colon V \to 2^P$ for some set P of atomic propositions.

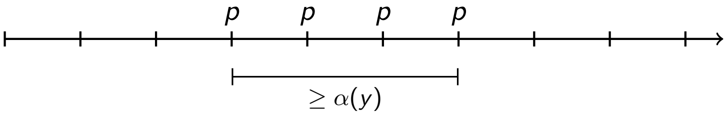


Winning conditions are expressed by a PLTL formula φ over P.

- Play: path $\rho_0\rho_1\rho_2\ldots$ through (V, E) starting in v_0.
- $\rho_0\rho_1\rho_2\ldots$ winning for Player 0 w.r.t. variable valuation α: $(\rho_0\rho_1\rho_2\ldots, 0, \alpha) \models \varphi$. Otherwise winning for Player 1.
- Strategy for Player i: $\sigma \colon V^*V_i \to V$ s.t. $(v, \sigma(wv)) \in E$.
- Winning strategy for Player i w.r.t. α: every play that is consistent with σ is won by Player i w.r.t. α.

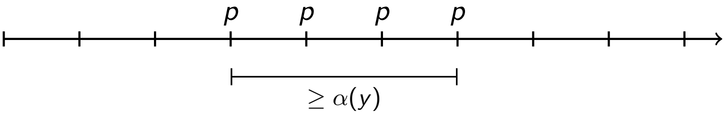
PLTL Games: Examples

- Winning condition $\mathbf{FG}_{\leq y} p$. Player 0's goal: eventually satisfy p for at least $\alpha(y)$ steps.

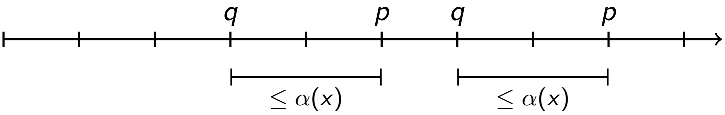


PLTL Games: Examples

- Winning condition $\mathbf{FG}_{\leq y} p$. Player 0's goal: eventually satisfy p for at least $\alpha(y)$ steps.

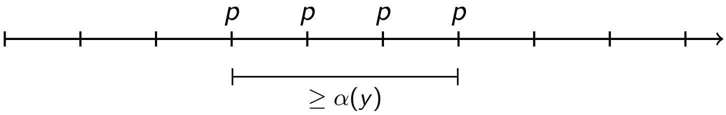


- Winning condition $\mathbf{G}(q \rightarrow \mathbf{F}_{\leq x} p)$. Player 0's goal: uniformly bound the waiting times between requests q and responses p by $\alpha(x)$.

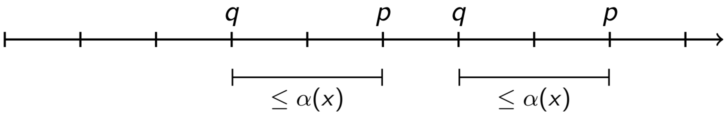


PLTL Games: Examples

- Winning condition $\mathbf{FG}_{\leq y} p$. Player 0's goal: eventually satisfy p for at least $\alpha(y)$ steps.



- Winning condition $\mathbf{G}(q \to \mathbf{F}_{\leq x} p)$. Player 0's goal: uniformly bound the waiting times between requests q and responses p by $\alpha(x)$.



Note: both winning conditions induce an optimization problem: maximize $\alpha(y)$ respectively minimize $\alpha(x)$.

Outline

1. Introduction
2. **Decision Problems**
3. Optimization Problems
4. Conclusion

Previous Work

Theorem (Pnueli, Rosner '89)

Determining the winner of an LTL *game is* **2EXPTIME***-complete.*

Previous Work

Theorem (Pnueli, Rosner '89)

Determining the winner of an LTL *game is* **2EXPTIME***-complete.*

The set of winning valuations for Player i in a PLTL game $\mathcal{G}$ is

$$\mathcal{W}^i_{\mathcal{G}} = \{\alpha \mid \text{Player } i \text{ has winning strategy for } \mathcal{G} \text{ w.r.t. } \alpha\} \enspace .$$

Previous Work

Theorem (Pnueli, Rosner '89)

Determining the winner of an LTL *game is* **2EXPTIME***-complete.*

The set of winning valuations for Player i in a PLTL game $\mathcal{G}$ is

$$\mathcal{W}^i_{\mathcal{G}} = \{\alpha \mid \text{Player } i \text{ has winning strategy for } \mathcal{G} \text{ w.r.t. } \alpha\} \ .$$

Theorem (Kupferman, Piterman, Vardi '07)

The following problem is **2EXPTIME***-complete: Given a* $\mathrm{PROMPT}-\mathrm{LTL}$ *game* $\mathcal{G}$*, is* $\mathcal{W}^0_{\mathcal{G}}$ *non-empty?*

Solving PLTL Games

Useful properties of PLTL:

- Duality: $\mathbf{F}_{\leq x}\varphi \equiv \neg\mathbf{G}_{\leq x}\neg\varphi$.
- Monotonicity: $\alpha(x) \leq \beta(x)$ and $\alpha(y) \geq \beta(y)$.
  - $(\rho, i, \alpha) \models \mathbf{F}_{\leq x}\varphi \Rightarrow (\rho, i, \beta) \models \mathbf{F}_{\leq x}\varphi$.
  - $(\rho, i, \alpha) \models \mathbf{G}_{\leq y}\varphi \Rightarrow (\rho, i, \beta) \models \mathbf{G}_{\leq y}\varphi$.

Solving PLTL Games

Useful properties of PLTL:

- Duality: $\mathbf{F}_{\leq x}\varphi \equiv \neg\mathbf{G}_{\leq x}\neg\varphi$.
- Monotonicity: $\alpha(x) \leq \beta(x)$ and $\alpha(y) \geq \beta(y)$.
  - $(\rho, i, \alpha) \models \mathbf{F}_{\leq x}\varphi \Rightarrow (\rho, i, \beta) \models \mathbf{F}_{\leq x}\varphi$.
  - $(\rho, i, \alpha) \models \mathbf{G}_{\leq y}\varphi \Rightarrow (\rho, i, \beta) \models \mathbf{G}_{\leq y}\varphi$.

Application:

Theorem

The following problems are **2EXPTIME***-complete: Given* PLTL *game* $\mathcal{G}$ *and* $i \in \{0, 1\}$.

i) *Is* $\mathcal{W}^i_{\mathcal{G}}$ *non-empty?*

ii) *Is* $\mathcal{W}^i_{\mathcal{G}}$ *infinite?*

iii) *Is* $\mathcal{W}^i_{\mathcal{G}}$ *universal?*

Outline

1. Introduction
2. Decision Problems
3. **Optimization Problems**
4. Conclusion

Finding Optimal Bounds

If φ contains only $\mathbf{F}_{\leq x}$ respectively only $\mathbf{G}_{\leq y}$, then solving games is an optimization problem: which is the *best* valuation in $\mathcal{W}_{\mathcal{G}}^{0}$?

Finding Optimal Bounds

If φ contains only $\mathbf{F}_{\leq x}$ respectively only $\mathbf{G}_{\leq y}$, then solving games is an optimization problem: which is the *best* valuation in $\mathcal{W}^0_{\mathcal{G}}$?

Theorem

Let $\varphi_{\mathbf{F}}$ be $\mathbf{G}_{\leq y}$-free and $\varphi_{\mathbf{G}}$ be $\mathbf{F}_{\leq x}$-free, let $\mathcal{G}_{\mathbf{F}} = (\mathcal{A}, \varphi_{\mathbf{F}})$ and $\mathcal{G}_{\mathbf{G}} = (\mathcal{A}, \varphi_{\mathbf{G}})$. The following values can be computed in doubly-exponential time:

- $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{F}}}} \max_{x \in \mathrm{var}(\varphi_{\mathbf{F}})} \alpha(x)$.

Finding Optimal Bounds

If φ contains only $\mathbf{F}_{\leq x}$ respectively only $\mathbf{G}_{\leq y}$, then solving games is an optimization problem: which is the *best* valuation in $\mathcal{W}_{\mathcal{G}}^0$?

Theorem

Let $\varphi_{\mathbf{F}}$ be $\mathbf{G}_{\leq y}$-free and $\varphi_{\mathbf{G}}$ be $\mathbf{F}_{\leq x}$-free, let $\mathcal{G}_{\mathbf{F}} = (\mathcal{A}, \varphi_{\mathbf{F}})$ and $\mathcal{G}_{\mathbf{G}} = (\mathcal{A}, \varphi_{\mathbf{G}})$. The following values can be computed in doubly-exponential time:

- $\min_{\alpha \in \mathcal{W}_{\mathcal{G}_{\mathbf{F}}}^0} \max_{x \in \mathrm{var}(\varphi_{\mathbf{F}})} \alpha(x)$.
- $\min_{\alpha \in \mathcal{W}_{\mathcal{G}_{\mathbf{F}}}^0} \min_{x \in \mathrm{var}(\varphi_{\mathbf{F}})} \alpha(x)$.
- $\max_{\alpha \in \mathcal{W}_{\mathcal{G}_{\mathbf{G}}}^0} \max_{y \in \mathrm{var}(\varphi_{\mathbf{G}})} \alpha(y)$.
- $\max_{\alpha \in \mathcal{W}_{\mathcal{G}_{\mathbf{G}}}^0} \min_{y \in \mathrm{var}(\varphi_{\mathbf{G}})} \alpha(y)$.

A First Idea

Duality, monotonicity, alternating-color technique **[KPV07]** $\Rightarrow$ it suffices to consider $\mathrm{PROMPT-LTL}$ games $\mathcal{G}_P$: determine $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x)$.

A First Idea

Duality, monotonicity, alternating-color technique **[KPV07]** $\Rightarrow$ it suffices to consider $\mathrm{PROMPT-LTL}$ games $\mathcal{G}_P$: determine $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x)$.

Lemma

There exists a $k \in \mathcal{O}(|\mathcal{A}| \cdot 2^{2^{|\varphi|}})$ *such that*

$$\mathcal{W}^0_{\mathcal{G}_P} \neq \emptyset \iff x \mapsto k \in \mathcal{W}^0_{\mathcal{G}_P} \iff \min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x) \leq k \ .$$

A First Idea

Duality, monotonicity, alternating-color technique **[KPV07]** $\Rightarrow$ it suffices to consider $\mathrm{PROMPT-LTL}$ games $\mathcal{G}_P$: determine $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x)$.

Lemma

There exists a $k \in \mathcal{O}(|\mathcal{A}| \cdot 2^{2^{|\varphi|}})$ *such that*

$$\mathcal{W}^0_{\mathcal{G}_P} \neq \emptyset \iff x \mapsto k \in \mathcal{W}^0_{\mathcal{G}_P} \iff \min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x) \leq k \ .$$

As we can test $\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$ effectively, it suffices to check all $k' < k$.

Example: $\varphi = \mathbf{G}(q \rightarrow \mathbf{F}_{\leq x} p)$ and $\alpha(x) = 2$:

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } (\mathcal{A}, \mathbf{G}(q \rightarrow p \vee \mathbf{X}(p \vee \mathbf{X}p))) \ .$$

A First Idea

Duality, monotonicity, alternating-color technique **[KPV07]** $\Rightarrow$ it suffices to consider $\mathrm{PROMPT-LTL}$ games $\mathcal{G}_P$: determine $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x)$.

Lemma

There exists a $k \in \mathcal{O}(|\mathcal{A}| \cdot 2^{2^{|\varphi|}})$ *such that*

$$\mathcal{W}^0_{\mathcal{G}_P} \neq \emptyset \iff x \mapsto k \in \mathcal{W}^0_{\mathcal{G}_P} \iff \min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_P}} \alpha(x) \leq k \ .$$

As we can test $\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$ effectively, it suffices to check all $k' < k$.

Example: $\varphi = \mathbf{G}(q \rightarrow \mathbf{F}_{\leq x} p)$ and $\alpha(x) = 2$:

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } (\mathcal{A}, \mathbf{G}(q \rightarrow p \vee \mathbf{X}(p \vee \mathbf{X}p))) \ .$$

Problem: this approach takes quadruply-exponential time.

A Better Idea

Faster algorithm for "$\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$?" provided $\alpha(x) \leq k$:

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'.
2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$.
3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.
4. Solve parity game $\mathcal{A} \times \mathfrak{P}$

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } \mathcal{A} \times \mathfrak{P}$$

A Better Idea

Faster algorithm for "$\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$?" provided $\alpha(x) \leq k$:

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'. $|\varphi'| \leq |\varphi|$
2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$.
3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.
4. Solve parity game $\mathcal{A} \times \mathfrak{P}$

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } \mathcal{A} \times \mathfrak{P}$$

A Better Idea

Faster algorithm for "$\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$?" provided $\alpha(x) \leq k$:

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'. $|\varphi'| \leq |\varphi|$
2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$. $|\mathfrak{A}_{\varphi'}| \leq |\varphi'| \cdot 2^{|\varphi'|}$
3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.
4. Solve parity game $\mathcal{A} \times \mathfrak{P}$

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } \mathcal{A} \times \mathfrak{P}$$

A Better Idea

Faster algorithm for "$\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$?" provided $\alpha(x) \leq k$:

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'. $|\varphi'| \leq |\varphi|$
2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$. $|\mathfrak{A}_{\varphi'}| \leq |\varphi'| \cdot 2^{|\varphi'|}$
3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.
   $|\mathfrak{P}| \leq 2^{|\mathfrak{A}_{\varphi'}|^2} \cdot \alpha(x)^{|\mathfrak{A}_{\varphi'}|}$ with $|\mathfrak{A}_{\varphi'}|$ colors
4. Solve parity game $\mathcal{A} \times \mathfrak{P}$

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } \mathcal{A} \times \mathfrak{P}$$

A Better Idea

Faster algorithm for "$\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$?" provided $\alpha(x) \leq k$:

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'. $|\varphi'| \leq |\varphi|$
2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$. $|\mathfrak{A}_{\varphi'}| \leq |\varphi'| \cdot 2^{|\varphi'|}$
3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.
   $|\mathfrak{P}| \leq 2^{|\mathfrak{A}_{\varphi'}|^2} \cdot \alpha(x)^{|\mathfrak{A}_{\varphi'}|}$ with $|\mathfrak{A}_{\varphi'}|$ colors
4. Solve parity game $\mathcal{A} \times \mathfrak{P}$ in doubly-exponential time

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } \mathcal{A} \times \mathfrak{P}$$

A Better Idea

Faster algorithm for "$\alpha \in \mathcal{W}^0_{\mathcal{G}_P}$?" provided $\alpha(x) \leq k$:

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'. $|\varphi'| \leq |\varphi|$
2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$. $|\mathfrak{A}_{\varphi'}| \leq |\varphi'| \cdot 2^{|\varphi'|}$
3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.
   $|\mathfrak{P}| \leq 2^{|\mathfrak{A}_{\varphi'}|^2} \cdot \alpha(x)^{|\mathfrak{A}_{\varphi'}|}$ with $|\mathfrak{A}_{\varphi'}|$ colors
4. Solve parity game $\mathcal{A} \times \mathfrak{P}$ in doubly-exponential time

$$\alpha \in \mathcal{W}^0_{\mathcal{G}_P} \iff \text{Player 0 wins } \mathcal{A} \times \mathfrak{P}$$

So, we have to solve exponentially many parity games, each in doubly-exponential time: gives doubly-exponential time.

An Example

Consider $\varphi = \mathbf{F}_{\leq x}\mathbf{G}p$ and $\alpha(x) = 2$.

An Example

Consider $\varphi = \mathbf{F}_{\leq x}\mathbf{G}p$ and $\alpha(x) = 2$.

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'.

Here: $\varphi' = \mathbf{FG}p$

An Example

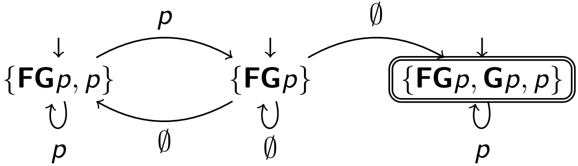
Consider $\varphi = \mathbf{F}_{\leq x}\mathbf{G}p$ and $\alpha(x) = 2$.

1. Replace all $\mathbf{F}_{\leq x}$ by $\mathbf{F}$ to obtain φ'.

Here: $\varphi' = \mathbf{FG}p$

2. Build Büchi automaton $\mathfrak{A}_{\varphi'}$ (textbook method).

Here:



Accepting run: visit accepting state every $\alpha(x)$ transitions.

An Example

3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.

$\mathfrak{A}_{\varphi'}$ is always unambiguous: no two accepting runs for any input.

An Example

3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.

$\mathfrak{A}_{\varphi'}$ is always unambiguous: no two accepting runs for any input.

Use **[Morgenstern, Schneider '10]**: Determinization of unambiguous Büchi automata

- States (essentially) a list $(S_0, \ldots, S_n)$ with $S_i \subseteq Q$, $n = |\mathfrak{A}_{\varphi'}|$.
- S_0 contains set of states reachable in $\mathfrak{A}_{\varphi'}$ via prefix of input.
- Build product with counters c_q keeping track of last visit in F by the unique run of $\mathfrak{A}_{\varphi'}$ ending in q.

An Example

3. Determinize $\mathfrak{A}_{\varphi'}$ and add counters simulating α to obtain deterministic parity automaton $\mathfrak{P}$.

$\mathfrak{A}_{\varphi'}$ is always unambiguous: no two accepting runs for any input.

Use **[Morgenstern, Schneider '10]**: Determinization of unambiguous Büchi automata

- States (essentially) a list $(S_0, \ldots, S_n)$ with $S_i \subseteq Q$, $n = |\mathfrak{A}_{\varphi'}|$.
- S_0 contains set of states reachable in $\mathfrak{A}_{\varphi'}$ via prefix of input.
- Build product with counters c_q keeping track of last visit in F by the unique run of $\mathfrak{A}_{\varphi'}$ ending in q.

$$|\mathfrak{P}| \leq \underbrace{2^{|\mathfrak{A}_{\varphi'}|^2}}_{(S_0,\ldots,S_n)} \cdot \underbrace{\alpha(x)^{|\mathfrak{A}_{\varphi'}|}}_{c_q}$$

Outline

Conclusion

We have presented an algorithm to determine optimal bounds in PLTL games in doubly-exponential time.

- For a known (doubly-exponential) upper bound k we test all smaller values $k' < k$.
- Each test can be done in doubly-exponential time.

The problem requires at least doubly-exponential time, as solving LTL games is **2EXPTIME**-complete.

Conclusion

We have presented an algorithm to determine optimal bounds in PLTL games in doubly-exponential time.

- For a known (doubly-exponential) upper bound k we test all smaller values $k' < k$.
- Each test can be done in doubly-exponential time.

The problem requires at least doubly-exponential time, as solving LTL games is **2EXPTIME**-complete.

Open questions:

- Ongoing research: Model-Checking and Games on pushdown graphs.
- Is there a *direct* algorithm that avoids checking all $k' < k$?
- Is there a tradeoff between the size of a finite-state winning strategy and its *quality*?