
Approximating Optimal Bounds in Prompt-LTL Realizability in Doubly-exponential Time

Joint work with Leander Tentrup and Martin Zimmermann

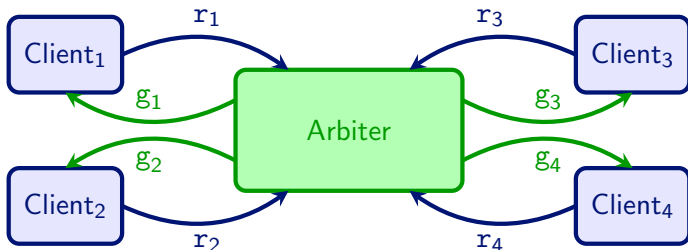
Alexander Weinert

Saarland University

September, 16th 2016
GandALF '16

Realizability: a Toy Example

- Setting: an arbiter with 4 clients
- Requests r_i from client i (controlled by the environment)
- Grants g_i for client i (controlled by the system)



Goal: Formal specification of arbiter's behavior

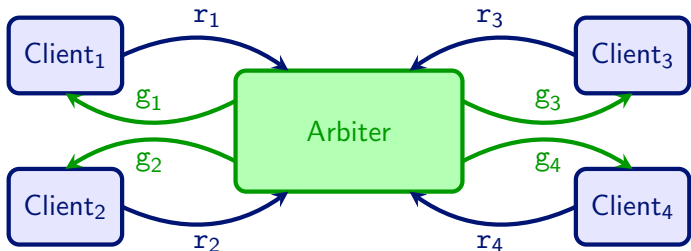
Linear Temporal Logic

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

+ typical shorthands

where p ranges over a finite set P of atomic propositions.

Continuing the Example: Specification



Specification:

$$\bigwedge_{i=1}^4 \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$$

Admissible execution:

| | | | | | | | | | | | |
|------|-------|-------|-------|-------|---|-------|-------|---|---|-------|-------|
| Env: | r_1 | r_1 | — | r_1 | — | — | r_1 | — | — | — | r_1 |
| Sys: | g_1 | — | g_1 | — | — | g_1 | — | — | — | g_1 | ... |

Prompt-LTL

Problem: $\mathbf{F} \varphi$ does not guarantee **when** φ holds true.

Solution: Add **prompt-eventually** operator $\mathbf{F_P}$:

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

Semantics: Given some word α , $k \in \mathbb{N}$

$(\alpha, k) \models \mathbf{F_P} \varphi$ if, and only if,

φ holds true within at most k steps

Prompt-LTL Example

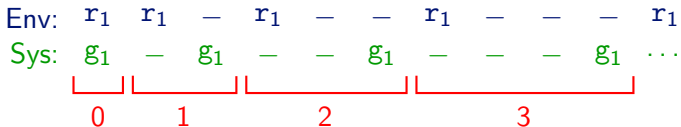
Before:

$$\bigwedge_{i=1}^4 \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$$

Now:

$$\bigwedge_{i=1}^4 \mathbf{G}(r_i \rightarrow \mathbf{F}_P g_i)$$

Execution α :



There exists no k such that

$$\alpha \models \bigwedge_{i=1}^4 \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$$

$$(\alpha, k) \models \bigwedge_{i=1}^4 \mathbf{G}(r_i \rightarrow \mathbf{F}_P g_i)$$

Prompt-LTL Realizability

Theorem (Kupferman, Piterman, Vardi '07)

The following problem is 2EXPTIME-complete:

Input: *Prompt-LTL formula φ over $I \cup O$*

Question: *Does there exist a strategy $\sigma: (2^I)^+ \rightarrow 2^O$ and a bound k , such that every word consistent with σ models φ w.r.t. k ?*

Now: Prompt-LTL realizability as **optimization problem**

Theorem (Z. '11)

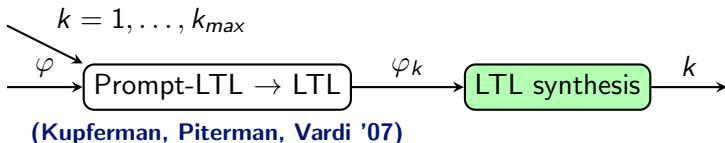
The minimal k such that there exists a strategy $\sigma: (2^I)^+ \rightarrow 2^O$ such that every word consistent with σ models φ w.r.t. k can be determined in triply-exponential time.

Prompt-LTL Realizability

Theorem

*The minimal k as defined previously can be **approximated** within a factor of 2 in **doubly**-exponential time.*

Prompt-LTL Approximation



Theorem (Kupferman, Piterman, Vardi '07)

For every Prompt-LTL formula φ and each bound $k \in \mathbb{N}$, there exists an LTL formula φ_k , such that

- if φ_k is realizable, then $(\varphi, 2k)$ is realizable, and
- if (φ, k) is realizable, then φ_k is realizable

Theorem (Kupferman, Piterman, Vardi '07)

If (φ, k) is realizable for some $k \in \mathbb{N}$, then (φ, k') is realizable for some k' doubly exponential in $|\varphi|$.

Construction of φ_k : Alternating Color

Given: Prompt-LTL formula φ , bound $k \in \mathbb{N}$.

Wanted: LTL formula φ_k .

Idea: Use fresh proposition $r \notin P$, “color” α .

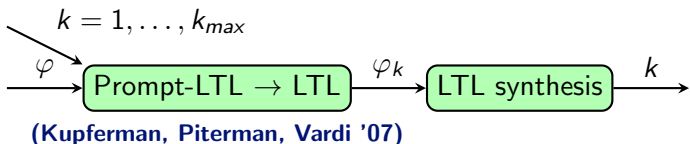
$$\alpha = \alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \quad \alpha_5 \quad \alpha_6 \quad \alpha_7 \quad \alpha_8 \quad \alpha_9$$
$$r \quad r \quad \neg r \quad \neg r \quad r \quad r \quad \neg r \quad \neg r \quad r \quad r$$

1. Replace each $\mathbf{F_P} \psi$ by LTL formula $\text{rel}(\mathbf{F_P} \psi)$ stating
 - “ ψ holds within one color change”
2. Add ψ_k stating
 - “The coloring changes after at most k steps”

$$\begin{array}{ccc} \text{(Prompt-LTL)} & & \text{(LTL)} \\ & \searrow & \swarrow \\ & \varphi \rightsquigarrow \text{rel}(\varphi) \wedge \psi_k & \end{array}$$

Correctness due to **(Kupferman, Piterman, Vardi '07)**

The Algorithm

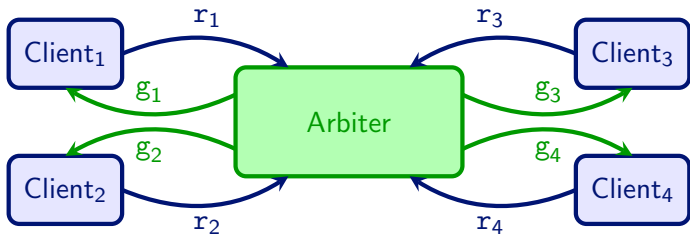


- 1: **if** φ unrealizable **then**
- 2: **return** " φ unrealizable"
- 3: **for** $k = 0, 1, 2, \dots, 2^{2^{|\varphi|}}$ **do**
- 4: **if** $\text{rel}(\varphi) \wedge \psi_k$ realizable **then**
- 5: **return** $2k$

Run-time: doubly-exponential in $|\varphi|$:

- Lines 1 and 4: doubly-exponential time.
- At most doubly-exponentially many iterations.

Back to the Example



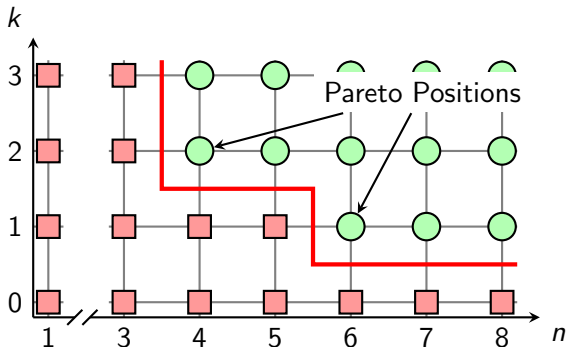
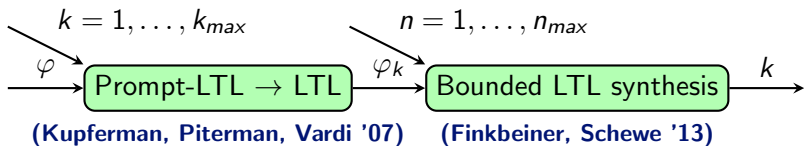
Parameters:

- Number of clients: r
 - Number of prioritized clients: r_p
1. Answer every request of clients 1 through r_p promptly:
 $\bigwedge_{1 \leq i \leq r_p} \mathbf{G}(r_i \rightarrow \mathbf{F}_P g_i)$
 2. Answer every other request eventually: $\bigwedge_{r_p < i} \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$
 3. At most one grant at a time: $\mathbf{G} \bigwedge_{i \neq j} \neg(g_i \wedge g_j)$

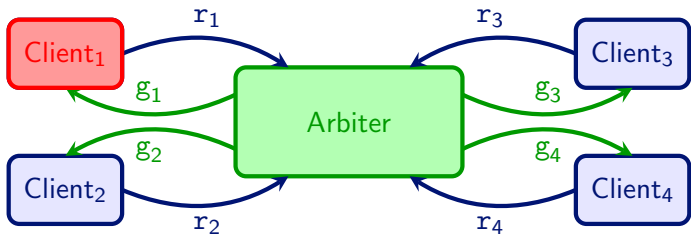
LTL synthesis vs. Prompt-LTL synthesis

| Resources | Prioritized Resources | LTL [s] | Prompt-LTL [s] |
|-----------|-----------------------|---------|----------------|
| 3 | 0 | 0.26 | 0.37 |
| | 1 | | 0.47 |
| | 2 | | 0.64 |
| | 3 | | 0.72 |
| 4 | 0 | 0.32 | 0.47 |
| | 1 | | 1.32 |
| | 2 | | 1.52 |
| | 3 | | 1.72 |
| | 4 | | 1.72 |

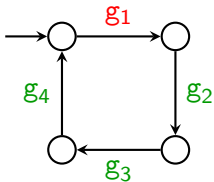
Bounded Prompt-LTL Approximation



Strategies: Slow, but Small

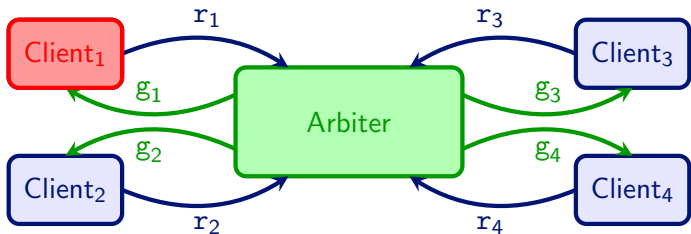


Always assume the worst: All requests in each step

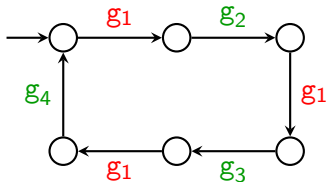


\rightsquigarrow 4 states, maximal delay 3

Strategies: Fast, but Large



Always assume the worst: All requests in each step



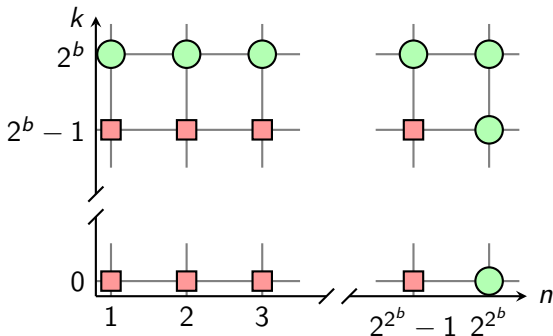
\rightsquigarrow 6 states, maximal delay 2

Pareto Positions

Theorem

There exists a family of Prompt-LTL formulas φ_b of size linear in b such that the output player has:

- a positional strategy realizing φ_b w.r.t. $k = 2^b$, and
- a strategy of size $n = 2^{2^b}$ realizing φ_b w.r.t. $k = 0$.



Conclusion

Our contribution:

- The first approximation algorithm for Prompt-LTL realizability with doubly-exponential running time
- Computes a realizing strategy
- Applicable to stronger logics as well
- Prototypical implementation
- Upper and lower bounds on tradeoff time vs. memory

Take-away:

- Relaxing the optimality requirement for Prompt-LTL yields exponentially better runtime
- In general, memory can be traded for response time and vice versa.