Pareto-Rational Verification

Clément Tamines (Université de Mons)

Joint work with

Véronique Bruyère (Université de Mons)

Jean-François Raskin (Université libre de Bruxelles)

September 2022

CONCUR 2022

Outline

1. Background

2. Pareto-Rational Verification

3. Universal Pareto-Rational Verification

4. Our Results

Outline

1. Background
2. Pareto-Rational Verification
3. Universal Pareto-Rational Verification
4. Our Results

Formal Verification

Motivation: ensure the correctness of systems responsible for **critical tasks**

Classical approach to **Formal Verification** (FV)

- **model of the system** to verify
- **model of the environment** in which it is executed
- **specification** φ to be enforced by the system

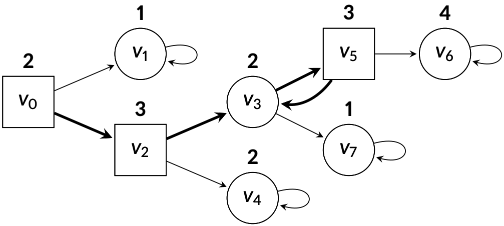
Goal: check if φ **satisfied in all executions** of the system in the environment

Limitations:

- check **single behavior** of the system
- against **potentially irrational behaviors** of environment

Games: Arenas, Plays and Objectives

Game Arena: tuple $G = (V, V_0, V_1, E, v_0)$ with (V, E) a directed graph



Play: infinite path starting with the **initial vertex** v_0, $\rho = v_0 v_2 (v_3 v_5)^\omega$

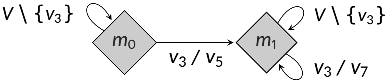
Objective Ω_i for Player $i \in \{0, 1\}$:

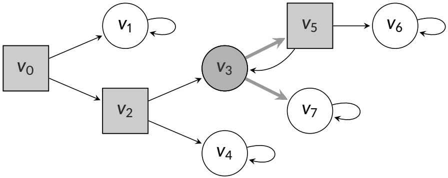
- subset of plays, ρ **satisfies** Ω_i if $\rho \in \Omega_i$
- **parity**: plays whose **minimum priority** seen infinitely often is **even**

Games: Strategies and Consistency

Finite-memory strategy $\sigma_i : V^* \times V_i \rightarrow V$ **dictates the choices** of Player i

$\rightarrow$ given $\underbrace{v_0 v_1 \ldots}_{h} \underbrace{v_k}_{\in V_i}$ yields v_{k+1} using a **deterministic Moore machine** $\mathcal{M}$





A play is **consistent** with σ_i if $v_{k+1} = \sigma_i(v_0 \ldots v_k)$ $\forall k \in \mathbb{N}, \forall v_k \in V_i$

Consider the **set of plays consistent** with a strategy σ_0

$\rightarrow$ $\text{Plays}_{\sigma_0} = \{v_0 v_1{}^\omega, v_0 v_2 v_4{}^\omega, v_0 v_2 v_3 v_5 v_6{}^\omega, v_0 v_2 v_3 v_5 v_3 v_7{}^\omega\}$

Outline

1. Background
2. Pareto-Rational Verification
3. Universal Pareto-Rational Verification
4. Our Results

The Model

Stackelberg-Pareto game (SP game) [BRT21]: $\mathcal{G} = (G, \Omega_0, \Omega_1, \ldots, \Omega_t)$

- Player 0 (system): objective Ω_0
- Player 1 (environment): **several objectives** $\Omega_1, \ldots, \Omega_t$ (components)
- **Non-zero-sum**: multi-component environment with its own objectives

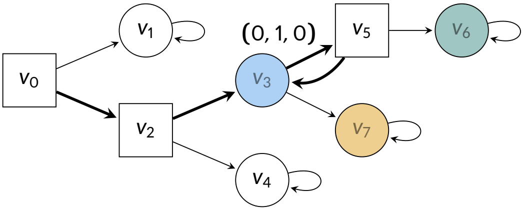
Payoff of ρ for Player 1 is the **vector of Booleans** $\text{pay}(\rho) \in \{0, 1\}^t$

- **order** $\leq$ on payoffs, e.g., $(0, 1, 0) < (0, 1, 1)$

$\Omega_1 = \text{Inf}(\{v_6\})$

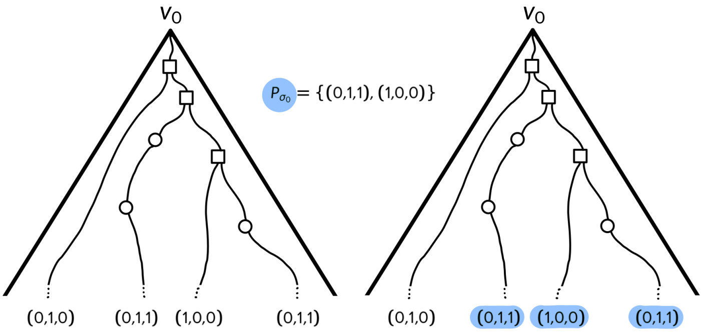
$\Omega_2 = \text{Inf}(\{v_3\})$

$\Omega_3 = \text{Inf}(\{v_7\})$



Pareto-Optimal Payoffs

1. Player 0 **provides** $\mathcal{M}$ encoding his strategy σ_0 (that we want to verify)
2. Player 1 **considers** Plays_{σ_0}
   - corresponding **set of payoffs** $\{\text{pay}(\rho) \mid \rho \in \text{Plays}_{\sigma_0}\}$
   - identify **Pareto-optimal** (PO) payoffs (maximal w.r.t. $\leq$) : set P_{σ_0}

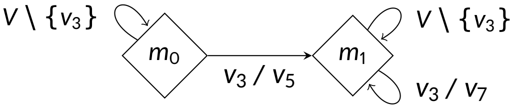


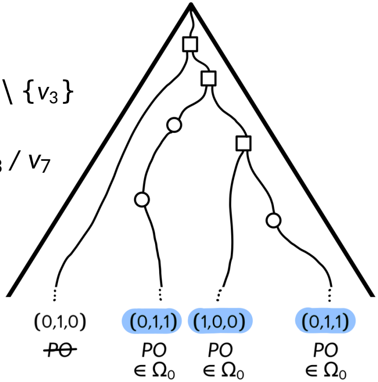
Pareto-Rational Verification problem (PRV problem)

Given a deterministic Moore machine $\mathcal{M}$ encoding a strategy σ_0, verify if every play $\rho \in \text{Plays}_{\sigma_0}$ with $\text{pay}(\rho) \in P_{\sigma_0}$ is such that $\rho \in \Omega_0$

Environment is **rational** and responds to σ_0 **to get a Pareto-optimal payoff**

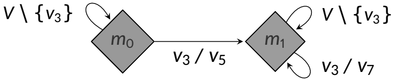
$\longrightarrow$ Verify that σ_0 **satisfies** Ω_0 in every such **rational response**



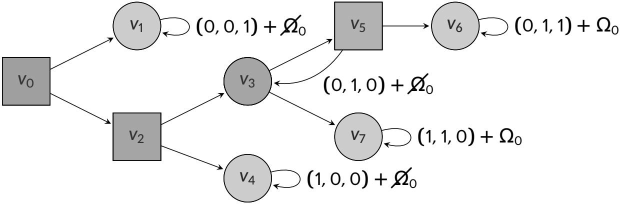


Example of the PRV Problem

Moore machine $\mathcal{M}$



SP game $\mathcal{G}$



- $\text{Plays}_{\sigma_0} = \{ \quad v_0 v_1{}^\omega, \quad v_0 v_2 v_4{}^\omega, \quad v_0 v_2 v_3 v_5 v_6{}^\omega, \quad v_0 v_2 v_3 v_5 v_3 v_7^\omega \}$
- $\text{payoffs} = \{ (0,0,1), \ (1,0,0), \quad (0,1,1), \quad (1,1,0) \quad \}$
- $P_{\sigma_0} = \{ (1,1,0), \ (0,1,1) \}$

$\longrightarrow$ together $\mathcal{M}$ and $\mathcal{G}$ form a **positive instance** to the PRV problem

Outline

1. Background

2. Pareto-Rational Verification

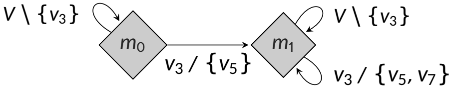
3. Universal Pareto-Rational Verification

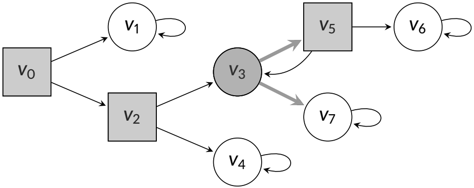
4. Our Results

Specifying Multiple Strategies

Nondeterministic Moore machine: lift determinism of next-move function

→ given $\underbrace{v_0 v_1 \dots}_{h} \ \underbrace{v_k}_{\in V_i}$ yields v_{k+1} from a **set of possible successors**





The machine $\mathcal{M}$ **embeds** a (possibly infinite) **set of strategies** $[\![\mathcal{M}]\!]$

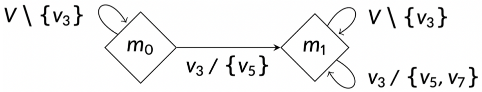
- $\sigma_0^k, k \geq 1$ such that $\sigma_0^k(hv_3) = v_5$, $\sigma_0^k(v_0v_2(v_3v_5)^k v_3) = v_7$
- σ_0 such that $\sigma_0(v_3) = v_5$

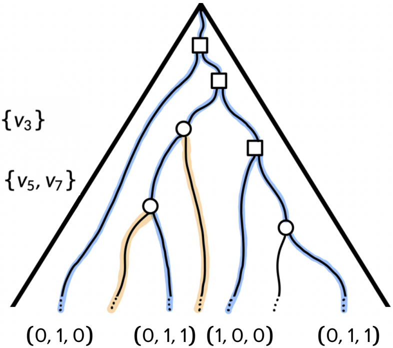
Different from determinizing by selecting a single successor

Universal Pareto-Rational Verification problem (UPRV problem)

Given a nondeterministic Moore machine $\mathcal{M}$, verify if for all strategies $\sigma_0 \in [\![\mathcal{M}]\!]$, every play $\rho \in \text{Plays}_{\sigma_0}$ with $\text{pay}(\rho) \in P_{\sigma_0}$ is such that $\rho \in \Omega_0$

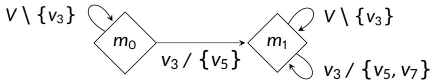
Generalization of the PRV problem to **multiple strategies**



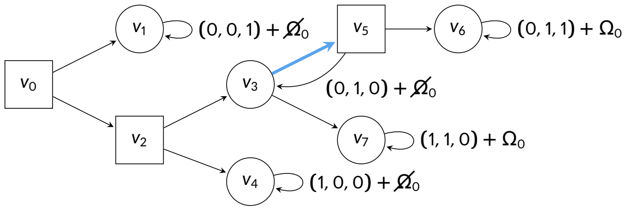


Example of the UPRV Problem

Moore machine $\mathcal{M}$



SP game $\mathcal{G}$



- $\text{Plays}_{\sigma_0} = \{ \quad v_0 v_1{}^{\omega}, \quad v_0 v_2 v_4{}^{\omega}, \quad v_0 v_2 (v_3 v_5)^* v_6{}^{\omega}, \quad v_0 v_2 (v_3 v_5)^{\omega} \ \}$
- $\text{payoffs} = \{ \ (0,0,1), \ (1,0,0), \quad (0,1,1), \quad (0,1,0) \quad \}$
- $P_{\sigma_0} = \{ \ (1,0,0), \ (0,1,1) \ \}$

$\rightarrow$ together $\mathcal{M}$ and $\mathcal{G}$ form a **negative instance** to the UPRV problem

Outline

1. Background

2. Pareto-Rational Verification

3. Universal Pareto-Rational Verification

4. Our Results

Complexity Results

Study both problems for **parity**, **Boolean Büchi**, and **LTL** objectives

PRV Problem

Objective	Complexity class
Parity	co-NP-complete
Boolean Büchi	Π_2P-complete
LTL	PSPACE-complete

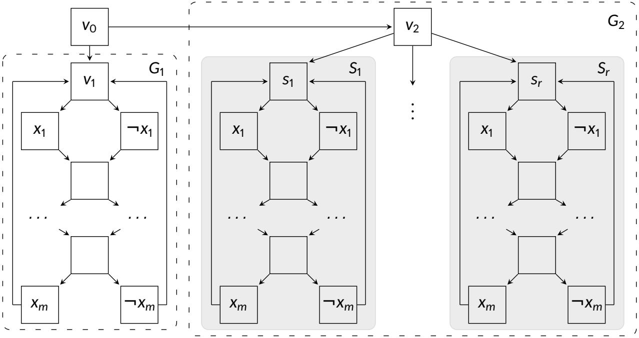
UPRV Problem

Objective	Complexity class
Parity	PSPACE, NP-hard, co-NP-hard
Boolean Büchi	PSPACE-complete
LTL	2EXPTIME-complete

co-NP-hardness of PRV for Parity Objectives

Shown using the **co-3SAT** (co-NP-complete) [Pap94]

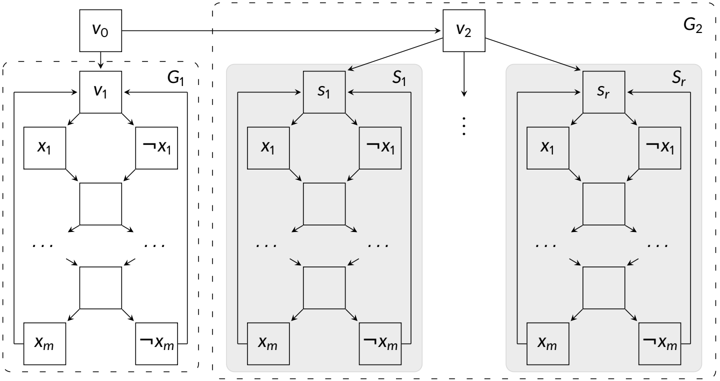
- $\psi = D_1 \wedge \cdots \wedge D_r$ in **3-Conjunctive Normal Form** over X
- decide whether **all valuations** of the variables in X **falsify** the formula



Goal: instance of **PRV positive** if and only if instance of **co-3SAT positive**

The Reduction: Objectives

	Ω_0	Ω_1	Ω_{x_1}	$\Omega_{\neg x_1}$	...	Ω_{x_m}	$\Omega_{\neg x_m}$	$(\Omega_{\ell^{1,1}}$	$\Omega_{\ell^{1,2}}$	$\Omega_{\ell^{1,3}})$	...	$(\Omega_{\ell^{r,1}}$	$\Omega_{\ell^{r,2}}$	$\Omega_{\ell^{r,3}})$
G_1	0	0	1	0	...	0	1	0	0	0	...	1	1	0
S_1	1	1	1	0	...	0	1	0	0	0	...	1	1	1
S_r	1	1	1	0	...	0	1	1	1	1	...	0	0	0



Fixed-Parameter Complexity

PRV and UPRV problem

Both problems are fixed-parameter tractable (FPT) for parity and Boolean Büchi with various parameters

Sound: in practice, we can assume those parameters to have **small values**

Additional Algorithm: based on **counterexamples**
→ **implemented and compared** using **toy example** and random instances

Thank you!

Bibliography I

[BRT21] Véronique Bruyère, Jean-François Raskin, and Clément Tamines.
Stackelberg-pareto synthesis.
In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference*, volume 203 of *LIPIcs*, pages 27:1–27:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[DF12] R.G. Downey and M.R. Fellows.
Parameterized Complexity.
Monographs in Computer Science. Springer New York, 2012.

[Pap94] Christos H. Papadimitriou.
Computational complexity.
Addison-Wesley, 1994.

Fixed-Parameter Complexity [DF12]

A problem is **fixed-parameter tractable** (FPT) for parameter k if there exists a solution running in $f(k) \times n^{\mathcal{O}(1)}$ where f is a function of k independent of n

Example: solving a problem is polynomial in input size, **exponential in k**
$\longrightarrow$ solving the problem is fixed-parameter tractable (easy if fix a small k)

PRV and UPRV problem

Both problems are fixed-parameter tractable (FPT) for parity and Boolean Büchi with various parameters

Sound: in practice, we can assume those parameters to have **small values**