Game Reductions in Formal Methods

Recent work on improved game analysis

Daniel Hausmann
Gothenburg University, Sweden

FM Retreat – January 12, 2023

Why Games?

- Model checking: $\mathcal{M} \models \varphi$?
- Validity checking: $\forall \mathcal{M}.\, \mathcal{M} \models \varphi$?
- Reactive synthesis: construct controller from specification φ

Why Games?

- Model checking: $\mathcal{M} \models \varphi$?
- Validity checking: $\forall \mathcal{M}.\, \mathcal{M} \models \varphi$?
- Reactive synthesis: construct controller from specification φ

Model $\mathcal{M}$: (Weighted) Transition system, Markov chain, game frame, ...

Formula φ: LTL / CTL, graded, probabilistic, ATL, ...

Why Games?

- Model checking: $\mathcal{M} \models \varphi$?
- Validity checking: $\forall \mathcal{M}. \mathcal{M} \models \varphi$?
- Reactive synthesis: construct controller from specification φ

Model $\mathcal{M}$: (Weighted) Transition system, Markov chain, game frame, ...

Formula φ: LTL / CTL, graded, probabilistic, ATL, ...

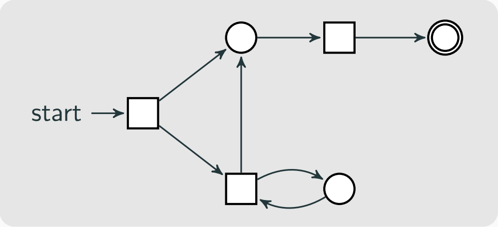
Why Games?

All these problems reduce to solving infinite duration 2-player games!

What are Games?

$$G = (V_\circ, V_\Box, E, \alpha)$$

nodes $V = V_\circ \mathbin{\dot\cup} V_\Box$ moves $E \subseteq V \times V$ objective $\alpha \subseteq V^\omega$

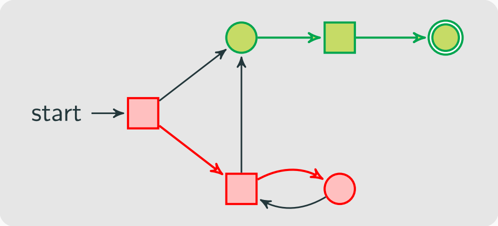


- (positional) $\circ$-strategy: function $s : V_\circ \to V$
- s is winning for player $\circ$ iff plays(s) $\subseteq \alpha$
- determinacy: every node is won by exactly one player

What are Games?

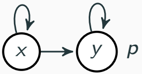
$$G = (V_\circ, V_\Box, E, \alpha)$$

nodes $V = V_\circ \mathbin{\dot\cup} V_\Box$ moves $E \subseteq V \times V$ objective $\alpha \subseteq V^\omega$



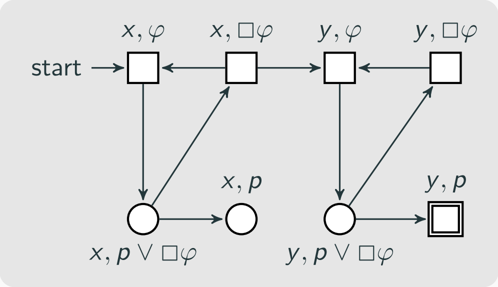
- (positional) $\circ$-strategy: function $s : V_\circ \to V$
- s is winning for player $\circ$ iff plays$(s) \subseteq \alpha$
- determinacy: every node is won by exactly one player

Model Checking Games, CTL

Transition system $\mathcal{M}$: 

CTL formula $\varphi = \mathrm{AF}\, p$

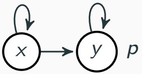
$$\mathcal{M}, x \models \varphi?$$



Theorem

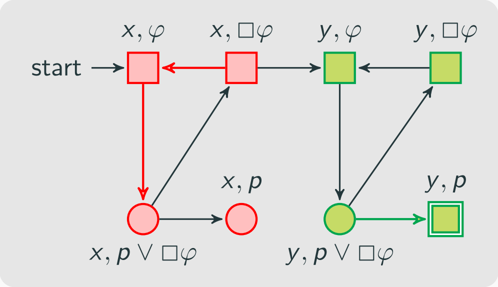
Player $\circ$ wins game if and only if $\mathcal{M} \models \varphi$.

Model Checking Games, CTL

Transition system $\mathcal{M}$: 

CTL formula $\varphi = \mathsf{AF}\, p$

$$\mathcal{M}, x \models \varphi?$$

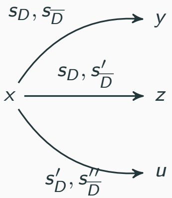


Theorem

Player $\circ$ wins game if and only if $\mathcal{M}, x \models \varphi$.

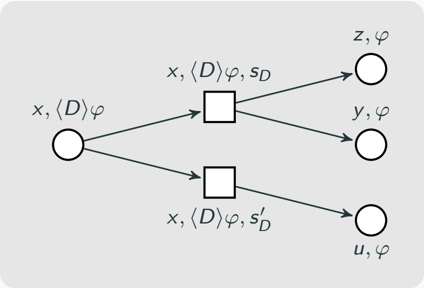
Model Checking Games, ATL

ATL: modalities $\langle D \rangle \varphi$ for coalitions D, interpreted over game frames:



$x \models \langle D \rangle \varphi$ if and only if
$\exists s_D. \forall s_{\overline{D}}. \delta(x, s_D, s_{\overline{D}}) \models \varphi$

Model checking game for
$x \models \langle D \rangle \varphi$:



Model Checking Games, ATL

Transform game frame $\mathcal{M}$ to effectivity function $\mathcal{M}'$

Theorem

$\mathcal{M} \models \varphi$ if and only if $\mathcal{M}' \models \varphi$.

Solve game for $\mathcal{M}' \models \varphi$

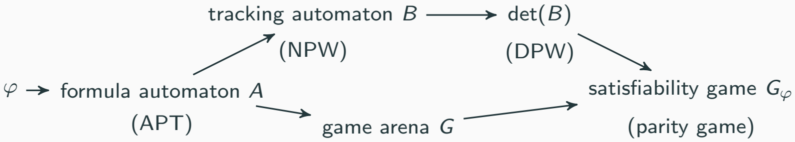
Cost: Transformation can be expensive
Benefit: Model checking game for $\mathcal{M}'$ can be much smaller than for $\mathcal{M}$

Ongoing work: Implement and benchmark this; leads to significant speed-up on (some) practical examples

Satisfiability Games

Input: CTL or μ-calculus formula φ Task: Is φ a tautology?

$$\forall \mathcal{M}.\, \mathcal{M} \models \varphi \text{ if and only if } \neg \exists \mathcal{M}.\, \mathcal{M} \models \neg\varphi$$



Theorem

Formula φ is satisfiable if and only if player $\circ$ wins G_φ.

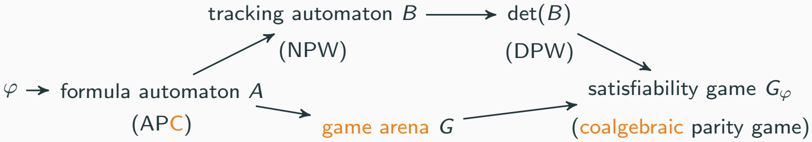
$|G_\varphi| \in \mathcal{O}(2^{|\varphi| \log |\varphi|})$, satisfiability problem is EXPTIME complete!

- *A Survey on Satisfiability Checking for the μ-Calculus through Tree Automata* [H, Piterman, 2022]

Satisfiability Games, generalized

Input: graded, probabilistic or ATL formula φ Task: $\exists \mathcal{M}. \mathcal{M} \models \varphi$?

($\mathcal{M}$: weighted TS, Markov chain or game frame)



Only modal steps in game arena G and resulting game G_φ change

Theorem [H, Schröder, 2019]

Formula φ is satisfiable if and only if player $\circ$ wins G_φ.

Ongoing work: Implementation and benchmarking of this in generic satisfiability solver "COOL 2", first reasoner for graded μ-calculus.

Reactive Synthesis

Given φ, construct controller $c : (2^I)^* \to 2^O$ s.t. $\forall i_0 i_1 \ldots \in (2^I)^\omega$,

$$(i_0 \cup c(i_0))(i_1 \cup c(i_0 i_1)) \ldots \models \varphi.$$

Workflow:

$$\varphi \xrightarrow{\text{enum}} A(\varphi) \xrightarrow{\text{enum}} \det(A(\varphi)) \xrightarrow{\text{enum}} \text{synthesis game } G_\varphi$$

(NBW) (DPW) (parity game)

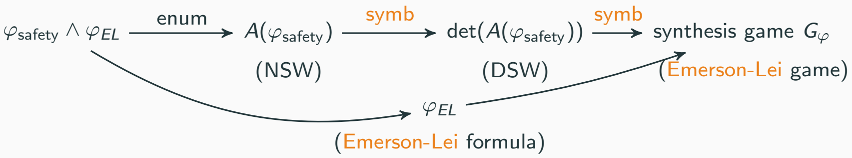
- $|G_\varphi| \in \mathcal{O}(2^{2^{|\varphi|}})$, synthesis problem is 2EXTPIME-complete
- Approach is not open to symbolic methods

Reactive Synthesis, Safety Emerson-Lei fragment

Ongoing work: Synthesis for safety Emerson-Lei fragment of LTL

$$\varphi_{\text{safety}} \wedge \varphi_{EL}$$

$\varphi_{EL} \in \mathbb{B}(\mathsf{GF}(I \cup O))$, e.g. $\varphi_{EL} = \mathsf{GF}(a) \wedge \neg\mathsf{GF}(b) = \mathsf{GF}(a) \wedge \mathsf{FG}(\neg b)$



Results so far: approach enables some amount of symbolic reasoning; novel solving algorithm for Emerson-Lei games

Summary

Take-away:

- Games capture central algorithmic content of many problems in CS
- Better game solving algorithms / smarter game reductions lead to improved problem solving

Ongoing work:

- ATL model checking in practice
- Generic satisfiability checking in practice (e.g. graded μ-calculus)
- Symbolic LTL synthesis via Emerson-Lei games