Back and Forth Between Logic and Games

Bertinoro, June 2009

Erich Grädel

Outline of this tutorial

Model Checking Games

- Model checking games for modal logic and first-order logic
- The strategy problem for finite games
- Fragments of first-order logics with efficient model checking
- Fixed point logics: LFP and modal μ-calculus
- Parity games are model checking games for fixed point logics
- Model checking complexity for LFP and μ-calculus
- Entanglement
- Definability of winning positions in parity games

Model checking via games

The model checking problem for a logic L

Given: structure $\mathfrak{A}$

formula $\psi \in L$

Question: $\mathfrak{A} \models \psi$?

Reduce model checking problem $\mathfrak{A} \models \psi$ to strategy problem for model checking game $G(\mathfrak{A}, \psi)$, played by

- Falsifier (also called Player 1, or Alter), and
- Verifier (also called Player 0, or Ego), such that

$$\mathfrak{A} \models \psi \iff \text{Verifier has winning strategy for } G(\mathfrak{A}, \psi)$$

$\Longrightarrow$ Model checking via construction of winning strategies

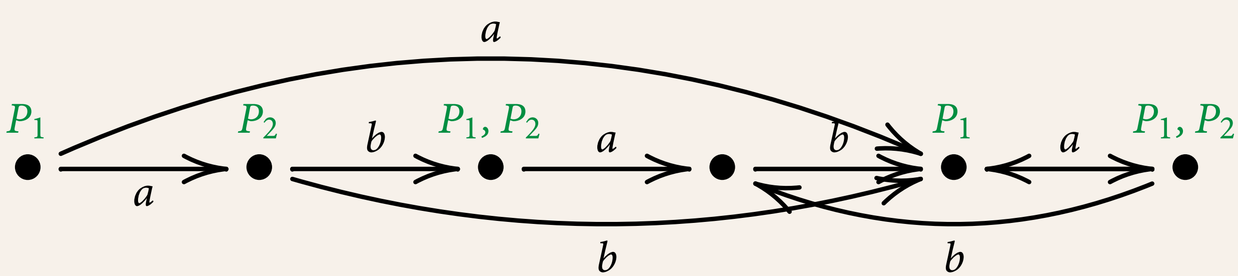
ML: propositional modal logic

Syntax: $\psi ::= P_i \mid \neg P_i \mid \psi \wedge \psi \mid \psi \vee \psi \mid \langle a \rangle \psi \mid [a]\psi$

Example: $P_1 \vee \langle a \rangle (P_2 \wedge [b]P_1)$

Semantics: transition systems = Kripke structures = labeled graphs

$$\mathcal{K} = (\underset{\substack{\text{states}\\\text{elements}}}{V}, \underset{\substack{\text{actions}\\\text{binary relations}}}{(E_a)_{a \in A}}, \underset{\substack{\text{atomic propositions}\\\text{unary relations}}}{(P_i)_{i \in I}})$$



$$[\![\psi]\!]^{\mathcal{K}} = \{v : \mathcal{K}, v \models \psi\} = \{v : \psi \text{ holds at state } v \text{ in } \mathcal{K}\}$$

$$\mathcal{K}, v \models \begin{matrix} \langle a \rangle \psi \\ [a]\psi \end{matrix} \quad :\Longleftrightarrow \quad \mathcal{K}, w \models \psi \text{ for } \begin{matrix} \text{some} \\ \text{all} \end{matrix} \; w \text{ with } (v, w) \in E_a$$

Model checking game for ML

Game $\mathcal{G}(\mathcal{K}, \psi)$ (for transition system $\mathcal{K}$ and $\psi \in \mathrm{ML}$)

Positions: (φ, v) φ subformula of ψ, $v \in V$

From position (φ, v), Verifier wants to show that $\mathcal{K}, v \models \varphi$, while Falsifier wants to prove that $\mathcal{K}, v \not\models \varphi$.

Verifier moves:

$$(\varphi \vee \vartheta, v) \longrightarrow (\varphi, v), \quad (\varphi \vee \vartheta, v) \longrightarrow (\vartheta, v) \qquad (\langle a \rangle \varphi, v) \longrightarrow (\varphi, w), \quad w \in vE_a$$

Falsifier moves:

$$(\varphi \wedge \vartheta, v) \longrightarrow (\varphi, v), \quad (\varphi \wedge \vartheta, v) \longrightarrow (\vartheta, v) \qquad ([a]\varphi, v) \longrightarrow (\varphi, w), \quad w \in vE_a$$

Terminal positions: (P_i, v), $(\neg P_i, v)$

If $\mathcal{K}, v \models P_i$ then Verifier has won at (P_i, v), otherwise Falsifier has won.

Lemma. $\mathcal{K}, v \models \varphi \iff$ Verifier has winning strategy from (φ, v).

Games and logics

Do games provide efficient solutions for model checking problems?

This depends on the logic, and on what we mean by efficient!

- How complicated are the resulting model checking games?
  - are all plays necessarily finite?
  - if not, what are the winning conditions for infinite plays?
  - structural complexity of the game graphs?
- How big are the resulting game graphs?

  how does the size of the game depend on different parameters of the input structure and the formula?

Logics and games

First-order logic (FO) or modal logic (ML): Model checking games have

- only finite plays
- positional winning condition

winning regions computable in linear time wrt. size of game graph

Fixed-point logics (LFP or L_μ): Model checking games are parity games

- admit infinite plays
- parity winning condition

Open problem: Are winning regions and winning strategies of parity games computable in polynomial time?

Reachability games

Two-player games with positional (reachability) winning condition, given by game graph (also called arena)

$$\mathcal{G} = (V, E), \qquad V = V_0 \cup V_1$$

- Player 0 (Ego) moves from positions $v \in V_0$,
  Player 1 (Alter) moves from $v \in V_1$,
- moves are along edges
  a play is a finite or infinite sequence $\pi = v_0 v_1 v_2 \cdots$ with $(v_i, v_{i+1}) \in E$
- winning condition: move or lose!
  Player σ wins at position v if $v \in V_{1-\sigma}$ and $vE = \varnothing$
  Note: this is a purely positional winning condition applying to finite plays only (infinite plays are draws)

Winning strategies and winning regions

Strategy for Player σ: $\quad f : \{v \in V_\sigma : vE \neq \varnothing\} \to V \quad$ with $(v, f(v)) \in E$.

f is **winning from position v** if Player σ wins all plays that start at v and are consistent with f.

Winning regions W_0, W_1:

$$W_\sigma = \{v \in V : \text{Player } \sigma \text{ has winning strategy from position } v\}$$

Algorithmic problems: Given a game $\mathcal{G}$

- compute winning regions W_0, W_1
- compute winning strategies

Associated decision problem:

$$\textsc{Game} := \{(\mathcal{G}, v) : \text{Player 0 has winning strategy for } \mathcal{G} \text{ from position } v\}$$

Algorithms for reachability games

Theorem

GAME is PTIME-complete and solvable in time $O(|V| + |E|)$.

remains true for strictly alternating games on graphs $\mathcal{G} = (V, E)$.

A simple polynomial-time algorithm

Compute winning regions inductively: $W_\sigma = \bigcup_{n\in\mathbb{N}} W_\sigma^n$ where

- $W_\sigma^0 = \{v \in V_{1-\sigma} : vE = \varnothing\}$
  (winning terminal positions for Player σ)
- $W_\sigma^{n+1} = \{v \in V_\sigma : vE \cap W_\sigma^n \neq \varnothing\} \cup \{ v \in V_{1-\sigma} : vE \subseteq W_\sigma^n\}$
  (positions with winning strategy in $\leq n + 1$ moves for Player i)

until $W_\sigma^{n+1} = W_\sigma^n$ (this happens for $n \leq |V|$).

Game and the satisfiability of propositional Horn formulae

Propositional Horn formulae: conjunctions of clauses of form

$$X \leftarrow X_1 \wedge \cdots \wedge X_n \qquad \text{and} \qquad 0 \leftarrow X_1 \wedge \cdots \wedge X_n$$

Theorem. Sat-Horn is Ptime-complete and solvable in linear time.

(actually, Game and Sat-Horn are essentially the same problem)

1) Game $\leq_{\text{log-lin}}$ Sat-Horn:

For $\mathcal{G} = (V_0 \cup V_1, E)$ construct Horn formula ψ with clauses

$$u \leftarrow v \qquad \text{for all } u \in V_0 \text{ and } (u, v) \in E$$

$$u \leftarrow v_1 \wedge \cdots \wedge v_m \qquad \text{for all } u \in V_1,\ uE = \{v_1, \ldots, v_m\}$$

The minimal model of ψ is precisely the winning region of Player 0.

$$(\mathcal{G}, v) \in \text{Game} \iff \psi_{\mathcal{G}} \wedge (0 \leftarrow v) \text{ is unsatisfiable}$$

2) SAT-HORN $\leq_{\text{log-lin}}$ GAME:

Define game $\mathcal{G}_\psi$ for Horn formula $\psi(X_1, \dots, X_n) = \bigwedge_{i \in I} C_i$

Positions: $\{0\} \cup \{X_1, \dots, X_n\} \cup \{C_i : i \in I\}$

Moves of Player 0: $X \longrightarrow C$ for $X = \text{head}(C)$

Moves of Player 1: $C \longrightarrow X$ for $X \in \text{body}(C)$

Note: Player 0 wins iff play reaches clause C with $\text{body}(C) = \varnothing$

Player 0 has winning strategy from position $X \iff \psi \models X$

Hence,

Player 0 wins from position 0 $\iff$ ψ unsatisfiable.

Alternating algorithms

nondeterministic algorithms, with states divided into accepting, rejecting, existential, and universal states

Acceptance condition: game with Players $\exists$ and $\forall$, played on computation graph $C(M, x)$ of M on input x

Positions: configurations of M

Moves: $C \to C'$ for C' successor configuration of C

- Player $\exists$ moves at existential configurations
  wins at accepting configurations
- Player $\forall$ moves at universal configurations
  wins at rejecting configurations

M accepts x $\quad :\Longleftrightarrow \quad$ Player $\exists$ has winning strategy for game on $C(M, x)$

Alternating versus deterministic complexity classes

Alternating time $\equiv$ deterministic space

Alternating space $\equiv$ exponential deterministic time

$$\begin{array}{ccccccccc} \textsc{Logspace} & \subseteq & \textsc{Ptime} & \subseteq & \textsc{Pspace} & \subseteq & \textsc{Exptime} & \subseteq & \textsc{Expspace} \\ & & \| & & \| & & \| & & \| \\ & & \textsc{Alogspace} & \subseteq & \textsc{Aptime} & \subseteq & \textsc{Apspace} & \subseteq & \textsc{Aexptime} \end{array}$$

Alternating logspace algorithm for Game: Play the game !

Evaluation game for FO

FO: $\psi ::= R_i\bar{x} \mid \neg R_i\bar{x} \mid x = y \mid x \neq y \mid \psi \wedge \psi \mid \psi \vee \psi \mid \exists x\psi \mid \forall x\psi$

The game $\mathcal{G}(\mathfrak{A}, \psi)$ (for $\mathfrak{A} = (A, R_1, \ldots, R_m)$, $R_i \subseteq A^{r_i}$)

Positions: $\varphi(\bar{a})$ $\varphi(\bar{x})$ subformula of ψ, $\bar{a} \in A^k$

Verifier moves:

$$\varphi \vee \vartheta \longrightarrow \varphi, \qquad \varphi \vee \vartheta \longrightarrow \vartheta \qquad\qquad \exists x\varphi(x, \bar{b}) \longrightarrow \varphi(a, \bar{b}) \quad (a \in A)$$

Falsifier moves:

$$\varphi \wedge \vartheta \longrightarrow \varphi, \qquad \varphi \wedge \vartheta \longrightarrow \vartheta \qquad\qquad \forall x\varphi(x, \bar{b}) \longrightarrow \varphi(a, \bar{b}) \quad (a \in A)$$

Winning condition: φ atomic / negated atomic

Verifier wins at $\varphi(\bar{a}) \iff \mathfrak{A} \models \varphi(\bar{a})$

Falsifier wins at $\varphi(\bar{a}) \iff \mathfrak{A} \not\models \varphi(\bar{a})$

Complexity of FO model checking

To decide whether $\mathfrak{A} \models \psi$, construct the game $\mathcal{G}(\mathfrak{A}, \psi)$ and check whether Verifier has winning strategy from initial position ψ.

Efficient implementation: on-the-fly construction of game while solving it

Size of game graph can be exponential: $|\mathcal{G}(\mathfrak{A}, \psi)| \leq |\psi| \cdot |A|^{\mathrm{width}(\psi)}$

width(ψ): maximal number of free variables in subformulae

Complexity of FO model checking:

alternating time: $O(|\psi| + \mathrm{qd}(\psi) \log |A|)$ qd(ψ): quantifier-depth of ψ

alternating space: $O(\mathrm{width}(\psi) \cdot \log |A| + \log |\psi|)$

deterministic time: $O(|\psi| \cdot |A|^{\mathrm{width}(\psi)})$

deterministic space: $O(|\psi| + \mathrm{qd}(\psi) \log |A|)$

Complexity of FO model checking

- **Structure complexity** (ψ fixed) : ALOGTIME $\subseteq$ LOGSPACE
- **Expression complexity** and **combined complexity:** PSPACE

Crucial parameter for complexity: width of formula

$$\mathrm{FO}^k := \{\psi \in \mathrm{FO} : \mathrm{width}(\psi) \leq k\} = k\text{-variable fragment of FO}$$

ModCheck(FO^k) is PTIME-complete and solvable in time $O(|\psi| \cdot |A|^k)$

Fragments of FO with model checking complexity $O(|\psi| \cdot \|\mathfrak{A}\|)$:

— **ML** : propositional modal logic

— **FO^2** : formulae of width two

— **GF** : the guarded fragment of first-order logic

The guarded fragment of first-order logic (GF)

Fragment of first-order logic with only guarded quantification

$$\exists \bar{y}(\alpha(\bar{x}, \bar{y}) \wedge \varphi(\bar{x}, \bar{y})) \qquad \forall \bar{y}(\alpha(\bar{x}, \bar{y}) \rightarrow \varphi(\bar{x}, \bar{y}))$$

with guards α : atomic formulae containing all free variables of φ

Generalizes modal quantification: $\mathrm{ML} \subseteq \mathrm{GF} \subseteq \mathrm{FO}$

$$\langle a \rangle \varphi \equiv \exists y(E_a xy \wedge \varphi(y)) \qquad [a]\varphi \equiv \forall y(E_a xy \rightarrow \varphi(y))$$

Guarded logics generalize and, to some extent, explain the good algorithmic and model-theoretic properties of modal logics.

Model-theoretic and algorithmic properties of GF

- Satisfiability for GF is decidable (Andréka, van Benthem, Németi)
- GF has finite model property (Grädel)
- GF has (generalized) tree model property:
  every satisfiable formula has model of small tree width (Grädel)
- Extension by fixed points remains decidable (Grädel, Walukiewicz)
- . . .
- Guarded logics have small model checking games:
  $\|\mathcal{G}(\mathfrak{A}, \psi)\| = O(|\psi| \cdot \|\mathfrak{A}\|)$
  $\Longrightarrow$ efficient game-based model checking algorithms

Advantages of game based approach to model checking

- intuitive top-down definition of semantics
  (very effective for teaching logic)
- versatile and general methodology,
  can be adapted to many logical formalisms
- isolates the real combinatorial difficulties of an evaluation problem,
  abstracts from syntactic details.
- if you understand games, you understand alternating algorithms
- closely related to automata based methods
- algorithms and complexity results for many logic problems follow
  from results on games

Logics and games

First-order logic (FO) or modal logic (ML): Model checking games have

- only finite plays
- positional winning condition

Winning regions computable in linear time wrt. size of game graph

In many computer science applications, more expressive logics are needed.
Indeed, FO can only define local properties (Gaifman's Theorem).
In particular, first-order logic cannot express:

- reachability conditions
- winning positions in games.

What about temporal logics, dynamic logics, fixed-point logics,. . . ?

Model checking games for these logics admit infinite plays and need more complicated winning conditions.

Least fixed point logics

Extend a basic logical formalism by least and greatest fixed points

FO	(first-order logic)	$\longrightarrow$	**LFP**	(least fixed point logic)
ML	(modal logic)	$\longrightarrow$	L_μ	(modal μ-calculus)
GF	(guarded fragment)	$\longrightarrow$	μ**GF**	(guarded fixed point logic)
conjunctive queries		$\longrightarrow$	Datalog / Stratified Datalog	

Idea: Capture recursion.

For any definable **monotone relational operator**

$$F_\varphi : T \mapsto \{\bar{x} : \varphi(T, \bar{x})\}$$

make also the least and the greatest fixed point of F_φ definable:

$$[\mathbf{lfp}\, T\bar{x}\,.\,\varphi(T,\bar{x})](\bar{z}) \qquad [\mathbf{gfp}\, T\bar{x}\,.\,\varphi(T,\bar{x})](\bar{z})$$

$$\mu X\,.\,\varphi \qquad \nu X\,.\,\varphi$$

Greatest fixed points (in LFP)

$[\mathbf{gfp}\, T\bar{x}\,.\,\varphi(T,\bar{x})](\bar{a})$: $\bar{a}$ contained in greatest T with $T = \{\bar{x} : \varphi(T,\bar{x})\}$

this T exists if $F_\varphi : T \mapsto \{\bar{x} : \varphi(T,\bar{x})\}$ is **monotone** (preserves $\subseteq$)
to guarantee monotonicity: require that T **positive** in φ

Inductive construction of the greatest fixed point on a structure $\mathfrak{A}$:

$$T^0 := A^k \quad \text{(all tuples of appropriate arity)}$$

$$T^{\alpha+1} := F_\varphi(T^\alpha)$$

$$T^\lambda := \bigcap_{\alpha<\lambda} T^\alpha \quad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ decreasing sequence of stages $(T^\alpha \supseteq T^{\alpha+1})$,
converges to a fixed point T^∞ of F_φ

Fact: $T^\infty = \mathbf{gfp}(F_\varphi)$ (Knaster, Tarski)

Example: Bisimulation

$\mathcal{K} = (V, E, P_1, \dots, P_m)$ transition system

Bisimilarity on $\mathcal{K}$ is the greatest equivalence relation $Z \subseteq V \times V$ such that:

if $(u, v) \in Z$ then

– u and v have the **same atomic properties**

– from u and v there are **edges into the same equivalence classes**

Thus, bisimilarity is the greatest fixed point of the refinement operator

$$Z \quad \mapsto \quad \{(u, v) : \mathcal{K} \models \varphi(Z, u, v)\} \quad \text{where}$$

$$\varphi := \bigwedge_{i \le m} P_i u \leftrightarrow P_i v \wedge$$
$$\forall x\, (Eux \rightarrow \exists y (Evy \wedge Zxy)) \;\wedge\; \forall y (Evy \rightarrow \exists x (Eux \wedge Zxy))$$

u and v are bisimilar in $\mathcal{K} \iff \mathcal{K} \models [\mathbf{gfp}\, Zuv \,.\, \varphi](u, v)$

Least fixed point logic LFP

Syntax. LFP extends FO by fixed point rule:

- For every formula $\psi(T, x_1 \dots x_k) \in \mathrm{LFP}[\tau \cup \{T\}]$,
  T k-ary relation variable, occuring only positive in ψ,
  build formulae $[\mathbf{lfp}\, T\overline{x}\,.\,\psi](\overline{x})$ and $[\mathbf{gfp}\, T\overline{x}\,.\,\psi](\overline{x})$

Semantics. On τ-structure $\mathfrak{A}$, $\psi(T, \overline{x})$ defines monotone operator

$$\begin{aligned} \psi^{\mathfrak{A}} : \mathcal{P}(A^k) &\longrightarrow \mathcal{P}(A^k) \\ T \quad &\longmapsto \{\overline{a} : (\mathfrak{A}, T) \models \psi(T, \overline{a})\} \end{aligned}$$

- $\mathfrak{A} \models [\mathbf{lfp}\, T\overline{x}\,.\,\psi(T, \overline{x})](\overline{a}) \quad :\Longleftrightarrow \overline{a} \in \mathbf{lfp}(\psi^{\mathfrak{A}})$
  $\mathfrak{A} \models [\mathbf{gfp}\, T\overline{x}\,.\,\psi(T, \overline{x})](\overline{a}) \quad :\Longleftrightarrow \overline{a} \in \mathbf{gfp}(\psi^{\mathfrak{A}})$

Modal μ-calculus L_μ

Syntax. L_μ extends ML by fixed point rule:

- With every formula $\psi(X)$, where X occurs only positive in ψ L_μ also contains the formulae $\mu X.\psi$ and $\nu X.\psi$

Semantics. On transition system $\mathcal{K}$, $\psi(X)$ defines operator

$$\psi^{\mathcal{K}} : X \longmapsto [\![\psi]\!]^{(\mathcal{K},X)} = \{v : (\mathcal{K}, X), v \models \psi\}$$

$\psi^{\mathcal{K}}$ is monotone, and therefore has a least and a greatest fixed point

$$\mathbf{lfp}(\psi^{\mathcal{K}}) = \bigcap\{X : \psi^{\mathcal{K}}(X) \subseteq X\}, \qquad \mathbf{gfp}(\psi^{\mathcal{K}}) = \bigcup\{X : X \subseteq \psi^{\mathcal{K}}(X)\}$$

- $[\![\mu X.\psi]\!]^{\mathcal{K}} := \mathbf{lfp}(\psi^{\mathcal{K}})$, $\qquad [\![\nu X.\psi]\!]^{\mathcal{K}} := \mathbf{gfp}(\psi^{\mathcal{K}})$

Inductive generation of fixed points

$\psi(X)$ defines operator $\psi^{\mathcal{K}} : X \mapsto \{v : (\mathcal{K}, X), v \models \psi\}$

$$X^0 := \varnothing \qquad\qquad Y^0 := V$$

$$X^{\alpha+1} := \psi^{\mathcal{K}}(X^\alpha) \qquad\qquad Y^{\alpha+1} := \psi^{\mathcal{K}}(Y^\alpha)$$

$$X^\lambda := \bigcup_{\alpha<\lambda} X^\alpha \qquad (\lambda \text{ limit ordinal}) \qquad Y^\lambda := \bigcap_{\alpha<\lambda} Y^\alpha$$

$$X^0 \subseteq \cdots \subseteq X^\alpha \subseteq X^{\alpha+1} \subseteq \cdots \qquad\qquad Y^0 \supseteq \cdots \supseteq Y^\alpha \supseteq Y^{\alpha+1} \supseteq \cdots$$

These inductive sequences reach fixed points

$$X^\alpha = X^{\alpha+1} =: X^\infty, \qquad\qquad Y^\beta = Y^{\beta+1} =: Y^\infty$$

for some α, β, with $|\alpha|, |\beta| \leq | V|$

$$X^\infty = [\![\mu X.\psi]\!]^{\mathcal{K}} \qquad\qquad Y^\infty = [\![\nu X.\psi]\!]^{\mathcal{K}}$$

L_μ: Examples

- $\mathcal{K}, w \models \nu X . \langle a \rangle X \iff$ there is an infinite a-path from w in $\mathcal{K}$

  $\mathcal{K}, w \models \mu X . P \vee [a]X \iff$ every infinite a-path from w eventually hits P

- $\mathcal{K}, w \models \nu X \mu Y . \Diamond((P \wedge X) \vee Y) \iff$ on some path from w, P occurs infinitely often

- Logics of knowledge: multi-modal propositional logics where $[a]\varphi$ stands for "agent a knows φ"

  add common knowledge:

  everybody knows φ, and everybody knows that everybody knows φ, and everybody knows that everybody knows that everybody knows . . .

  expressible as a greatest fixed point: $C\varphi \equiv \nu X . (\varphi \wedge \bigwedge_a [a]X)$

Finite games and LFP

- Game is definable in LFP / L_μ

Player 0 has winning strategy for game $\mathcal{G}$ from position v

$$\Longleftrightarrow$$

$$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{lfp}\, Wx\,.\,(V_0x \wedge \exists y(Exy \wedge Wy)) \\ \vee (V_1x \wedge \forall y(Exy \rightarrow Wy))](v)$$

$$\Longleftrightarrow$$

$$\mathcal{G}, v \models \mu W\,.\,(V_0 \wedge \Diamond W) \vee (V_1 \wedge \Box W)$$

- Game is complete for LFP
  (via quantifier-free reductions on finite structures)

Duality of least ad greatest fixed points

$$\neg[\mathbf{lfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{z}) \quad\equiv\quad [\mathbf{gfp}\, T\overline{x}\,.\,\neg\varphi(\neg T,\overline{x})](\overline{z})$$
$$\neg\mu X\,.\,\varphi(X) \quad\equiv\quad \nu X\,.\,\neg\varphi(\neg X)$$

Corollary. Every formula in LFP or L_μ can be efficiently translated into an equivalent formula in negation normal form.

Importance of the modal μ-calculus

- encompasses most of the popular logics used in hardware verification: LTL, CTL, CTL*, PDL,. . . , and also many logics from other fields: game logic, description logics, etc.
- reasonably good algorithmic properties:
  - satisfiability problem decidable (Exptime-complete)
  - efficient model checking for practically important fragments of L_μ
  - automata-based algorithms
- nice model-theoretic properties:
  - finite model property
  - tree model property
- L_μ is the bisimulation-invariant fragment of MSO

Disadvantage: Fixed-point formulae are hard to read

Importance of fixed point logics

- general and versatile methodology to capture recursion in a natural and elegant way
- Finite model theory: great variety of fixed-point logics, based on operators that are not necessarily monotone.
- Databases: Datalog and other query languages based on fixed points
- Descriptive complexity: LFP captures PTIME (on ordered structures)

  On unordered finite structures, LFP can express certain PTIME-complete problems (such as GAME), but fails to express all of PTIME.
- ...

Model checking games for LFP and L_μ

LFP-game: extend FO-game by moves

$$[\mathbf{fp}\, T\overline{x}\,.\,\varphi](\overline{a}) \longrightarrow \varphi(T,\overline{a}) \qquad (\mathbf{fp} \in \{\mathbf{lfp}, \mathbf{gfp}\})$$

$$T\overline{b} \longrightarrow \varphi(T,\overline{b})$$

Similarly for L_μ: extend ML-game by moves

$$(\lambda X\,.\,\varphi, u) \longrightarrow (\varphi, u) \qquad (\lambda \in \{\mu, \nu\})$$

$$(X, w) \longrightarrow (\varphi, w)$$

Infinite plays possible

need winning condition for infinite plays

Model checking game for L_μ: Example

$$\psi = \mu X.P \lor \Box X \qquad \equiv \qquad [\mathbf{lfp}\, Tx\,.\, Px \lor \forall y(Exy \rightarrow Ty)](x)$$

$$\mathcal{K}: \quad \begin{tikzcd} \underset{a}{\bullet} \arrow[r] \arrow[rrr, bend left] & \underset{b}{\bullet} \arrow[r] & \underset{c}{\bullet} \arrow[r] & \overset{P}{\underset{d}{\bullet}} \end{tikzcd} \qquad \text{from node } a$$

$$\begin{tikzcd}
\psi, a \arrow[r] & P \lor \Box X, a \arrow[r] \arrow[dl] & \Box X, a \arrow[r] \arrow[dl] & X, d \arrow[r] & P \lor \Box X, d \arrow[r] \arrow[dr] & P, d \\
P, a & X, b \arrow[d] & & & \Box X, c \arrow[ul] & \Box X, d \\
& P \lor \Box X, b \arrow[dl] \arrow[dr] & & & P \lor \Box X, d \arrow[u] \arrow[dr] & \\
P, b & & \Box X, b \arrow[r] & X, c \arrow[ur] & & P, c
\end{tikzcd}$$

Winning conditions

On formulae $[\mathbf{lfp}\ T\overline{x}\,.\,\psi(T,\overline{x})](\overline{a})$ or $\mu X.\psi$ (where ψ has no fixed points), Verifier must win in a finite number of steps.

By forcing a cycle, **Falsifier** wins.

Are cycles always bad for Verifier?

No, not if they correspond to **greatest** fixed points

- **lfp**-cycles: **Falsifier** wins
- **gfp**-cycles: Verifier wins

What about cycles with both least and greatest fixed points?

The **outermost** fixed point on cycle determines the winner

Reminder: Parity games

$G = (V, E, \Omega)$, $\quad V = V_0 \cup V_1$, $\quad \Omega : V \rightarrow \mathbb{N}$

Player 0 moves at positions $v \in V_0$, Player 1 at positions $v \in V_1$

$\Omega(v)$ is the priority of position v

Play: finite or infinite sequence $\pi = v_0 v_1 v_2 \cdots$ with $(v_i, v_{i+1}) \in E$

Winning condition:

– finite plays: who cannot move, loses

– infinite plays: **least priority** seen **infinitely often** determines winner

$$\text{Player 0 wins } \pi \iff \min\{k : (\exists^\infty i)\Omega(v_i) = k\} \text{ is even}$$

Parity games are **positionally determined.**

They can be solved in NP $\cap$ Co-NP. The best deterministic algorithms known require time $O(n^{\sqrt{n}})$ or $O(n^{d/3})$ where d is the number of priorities.

Model checking games for LFP and L_μ

Extend FO-game by moves

$$
\begin{aligned}
[\mathbf{fp}\; T\overline{x}\,.\,\varphi](\overline{a}) &\longrightarrow \varphi(T,\overline{a}) \\
T\overline{a} &\longrightarrow \varphi(T,\overline{a})
\end{aligned}
$$

Parity winning condition, with following priority assignment:

- $\Omega(T\overline{a})$ is $\begin{cases} \text{even} & \text{if } T \textbf{ gfp}\text{-variable} \\ \text{odd} & \text{if } T \textbf{ lfp}\text{-variable} \end{cases}$
- $\Omega(T\overline{a}) \leq \Omega(T'\overline{b})$ if T' depends on T
  (i.e. if T free in $[\mathbf{fp}\; T'\overline{x}\,.\,\varphi(T', T, \overline{x})](\overline{a})$)
- $\Omega(\varphi)$ maximal, for other formulae φ

analogous for L_μ

Note: $|\Omega(V)|$: alternations between least and greatest fixed points

Significance of parity games

Enlarge game graphs to simplify winning conditions

many games with complicated winning strategies can be simulated by parity games (over larger game graphs)

- games with winning conditions formulated in temporal logic (LTL) or monadic second-order logic (S1S)
- Muller games: games where the winner only depends on which priorities are seen infinitely often.
- games that model reactive systems

Parity games admit positional (i.e., memoryless) winning strategies

Parity games arise as model checking games of fixed-point logics

Game-based model checking for fixed point logics

Two orthogonal aspects

- size of game graphs:

$$\|\mathcal{G}(\mathfrak{A}, \psi)\| \leq |\psi| \cdot f((\mathfrak{A}, \mathrm{width}(\psi))$$

  f determined by quantification mechanism of the given logic:
  number of accessible tuples

  small for modal and guarded formulae, and formulae with small width

  large for formulae with unbounded first-order quantification

- structure of game graphs

  determined by possible interaction of **lfp** / **gfp** operators among themselves and with $\exists, \forall, \wedge, \vee$.

Size of game graphs

- **Modal logics: ML, μ-calculus,...**

  accessible tuples: nodes

  $$\|\mathcal{G}(\mathfrak{A}, \psi)\| = O(|\psi| \cdot \|\mathfrak{A}\|)$$

- **Guarded logics: GF, μGF, Datalog LITE,...**

  accessible tuples: atomic facts

  $$\|\mathcal{G}(\mathfrak{A}, \psi)\| = O(|\psi| \cdot \|\mathfrak{A}\|)$$

- **FO, LFP, Datalog... (full first-order quantification)**

  accessible tuples: all

  $$\|\mathcal{G}(\mathfrak{A}, \psi)\| = O(|\psi| \cdot |A|^{\mathrm{width}(\psi)})$$

Model checking complexity of fixed point logics

- bounded width and bounded alternation depth: PTIME
- unbounded width and bounded alternation depth: EXPTIME
- bounded width and unbounded alternation depth: ???

Conjecture. The following problems are solvable in polynomial time:

(1) computing winning sets in parity games

(2) model checking for LFP-formulae of width k (for any $k \geq 2$)

(3) model checking for modal μ-calculus

(4) model checking for μGF

If any of these problems admits a polynomial time algorithm, then all of them do.

Easy cases of parity games

- **Well-founded games:** No cycles, only finite plays.

  Complexity: $O(|V| + |E|)$

- **Dull games:** Loops with minimal priority even / odd are disjoint

  Weak games: Priorities never decrease along edges.

  Weak games and dull games are equivalent

  Complexity: $O(|(V| + |E|)$

- **Nested solitaire games:** On each strongly connected component, only one player makes non-trivial moves

  Complexity $O(d(|V| + |E|))$

Easy fragments of fixed point logics

- **Formulae without fixed points**
  - — lead to well-founded games

- **Alternation free formula**
  - — no free **lfp**-variables in **gfp**-formulae and vice versa
  - — lead to dull games

- **Solitaire-LFP**
  - — in $\psi \wedge \varphi$ and $\forall y \varphi$, no free fixed-point variables inside φ
  - — lead to nested solitaire games

Expressive power and complexity of Solitaire-LFP

- Transitive closure logic (TC) $\leq$ Solitaire-LFP

$$[\mathbf{tc}_{x,y}\ \varphi](a,b) \equiv [\mathbf{lfp}\ Txy\,.\,\varphi(x,y) \vee \exists z(Txz \wedge \varphi(z,y))](a,b)$$

- On finite structures, Solitaire-LFP $\equiv$ TC
- On infinite structures, Solitaire-LFP $\not\leq$ TC

$$[\mathbf{gfp}\ Tx\,.\,\exists y(Exy \wedge Ty)](x): \quad \text{"there is infinite path from } x\text{"}$$

not expressible in TC (not even on countable structures)

- Model checking complexity of Solitaire-LFP

Structure complexity: Nlogspace-complete

combined complexity: Pspace-complete

Complexity of fixed point logics

	no fixpoints	alternation free	solitaire	full
modal	linear	linear	$O(d\,\|\mathfrak{A}\|\,\|\psi\|)$	?
guarded	linear	linear	$O(d\,\|\mathfrak{A}\|\,\|\psi\|)$	?
bounded	Ptime	Ptime	Ptime	?
full	Pspace	Exptime	Pspace	Exptime

Efficiently solvable cases of parity games

Although it is not known yet whether parity games can be solved efficiently, there are important cases for which polynomial-time algorithms have been found:

- parity games with a bounded number of priorities
- parity games where even and odd cycles do not intersect
- nested solitaire games
- parity games of bounded tree width
- parity games with bounded clique-width
- parity games with bounded DAG-width
- parity games with bounded Kelly-width
- parity games with **bounded entanglement**.

Entanglement of a directed graph

Entanglement measures to what extent the cycles of a graph are intertwined,

defined by means of **game**, played by a thief against a number of detectives, similar to games for tree width, directed tree width, and hypertree width. Yet, very significant differences between entanglement and tree width

Entanglement is intimately connected with the **modal μ-calculus**:

Transition systems with entanglement k can be decribed, up to bisimulation, by a μ-calculus formula with k fixed-point variables.

Application: The variable hierarchy of the μ-calculus is strict.

Entanglement captures difficulty of **parity games**

Parity games of bounded entanglement are solvable in polynomial time.

Entanglement: How to catch a thief

$G = (V, E)$ directed graph

Entanglement game: k detectives try to catch a thief on G

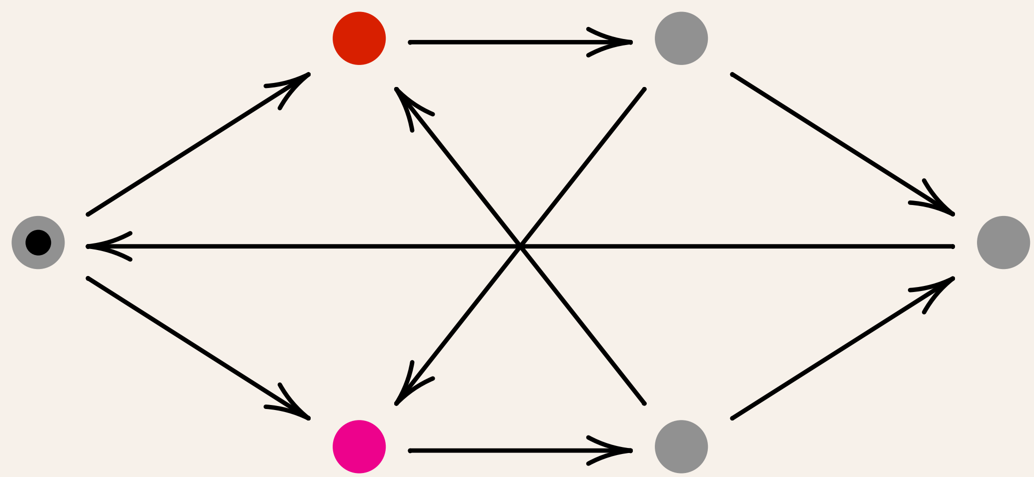
The thief selects an initial position $v_0 \in G$

The detectives are outside the graph

Move:

- the detectives stay where they are, or place one of them on the current position v of the thief.
- the thief moves to a successor $w \in vE$ not occupied by a detective. If no such position exists, the thief is caught and the detectives have won.

$$\mathrm{ent}(G) := \min\{k : k \text{ detectives have a strategy to catch the thief on } G\}$$



Proposition

(1) $\mathrm{ent}(G) = 0$, if and only if G is acyclic.

(2) If G is the graph of a unary function, then $\mathrm{ent}(G) = 1$.

(3) If G is an undirected tree, then $\mathrm{ent}(G) \leq 2$.

(4) The fully connected directed graph with n nodes has entanglement n.

Complexity of entanglement

Proposition. For any fixed $k \in \mathbb{N}$, the problem whether a given graph has entanglement k is solvable in polynomial time.

An **alternating** procedure that just plays the game needs space $(k+1)\log|V|)$ to store the current positions of the thief and the detectives.

$\textsc{Alogspace} \quad = \quad \textsc{Ptime}$

Open problem: Complexity of computing the entanglement

The variable hierarchy of the μ-calculus

$L_\mu[k]$: μ-calculus with at most k different fixed-point variables

- Most of the popular fragments of L_μ, such as CTL, PDL, CTL*, and game logic GL, are actually contained in $L_\mu[2]$.
- Syntactically, a formula in $L_\mu[k]$ is a tree with back-edges with entanglement at most k.
- Model checking for $L_\mu[2]$ is as hard as for the entire μ-calculus.

Theorem (Berwanger, G. , Lenzi)
The variable hierarchy of the μ-calculus is strict.

Since game logic GL is contained in $L_\mu[2]$, this also prove that GL $\subsetneq L_\mu$ and thus solves a problem posed by Parikh in 1985.

The superdetective game

For any parity game $G = (V, V_0, V_1, E, \Omega)$, Player σ, and any number $k \leq |V|$, we define the **Superdetective game** $G[\sigma, k]$:

Superdetective wants to show, using k detectives, that Player σ wins the parity game G. He controls positions $v \in V_\sigma$ and can place the k detectives Challenger controls positions $v \in V_{1-\sigma}$ of the opponent.

Playing the game: Superdetective and Challenger play the parity game. In addition, Superdetective may, at any move, transfer one detective to the current position of the parity game.

Winning condition: The play ends, when the parity game reaches a position occupied by a detective. Superdetective wins if the least position seen since this detective was placed there is even, if $\sigma = 0$ and odd, if $\sigma = 1$. In all other cases, Challenger wins.

Superdetective game and parity game

Proposition.

(1) If Player σ wins the parity game G, then Superdetective wins $G[\sigma, k]$ for $k = \mathrm{ent}(G)$.

Actually $k = \mathrm{ent}(G_f)$ suffices, where G_f is the subgame induced by a positional winning strategy f for Player σ.

(2) If, for some k, Superdetective wins $G[\sigma, k]$, then Player σ wins the parity game G.

Complexity: Computing the winner of the Superdetective game $G[\sigma, k]$ requires alternating space $((2k+1)\log|V|)$

Play the game and record current position of thief and current positions of the k detectives, along with the minimal priority since he was last posted.

Parity games of bounded entanglement

Theorem. The winner of a parity game G can be determined in $\textsc{Aspace}((2k+1)\log|V|)$, where k is the minimum entanglement of a subgame G_f induced by a memoryless winning strategy f.

Input : parity game G, with initial position v

guess $k_0 \leq |V|$

universally choose $k_1 \leq |V|$

if $k_0 \leq k_1$ **then**

  if Superdetective wins $G[0, k_0]$ from v **then** **accept** , **else** **reject**

if $k_1 < k_0$ **then**

  if Superdetective wins $G[1, k_1]$ from v **then** **reject** , **else** **accept**

Corollary.

Parity games of bounded entanglement are solvable in polynomial time.

Parity games and fixed points logics

Parity games arise as the model checking games for the least fixed point logic LFP and the modal μ-calculus L_μ.

For every structure $\mathfrak{A}$ and every formula $\psi(\overline{x}) \in$ LFP we find a parity game $\mathcal{G}(\mathfrak{A}, \varphi)$ such that

$$\mathfrak{A} \models \varphi(\overline{a}) \quad \Longleftrightarrow \quad \text{Player 0 wins } \mathcal{G}(\mathfrak{A}, \varphi) \text{ from initial position } \varphi(\overline{a}).$$

Moreover, $\mathcal{G}(\mathfrak{A}, \varphi)$ is first-order interpretable in $\mathfrak{A}$.

An efficient algorithm for solving parity games would give an efficient model-checking procedure for the modal μ-calculus.

We have also claimed the converse. It is time to prove it.

Descriptive complexity of parity games

Question: In which logics are the winning regions of parity games definable?

In particular, are the winning regions definable in
- monadic second-order logic MSO ?
- least fixed-point logic LFP ?
- modal μ-calculus L_μ ?

The answers very much depend on whether parity games with bounded or unbounded number of prorities are considered, and on whether game graphs are finite or (possibly) infinite.

Parity games as relational structures

Parity games with a bounded number of priorities:

$$(V, V_0, V_1, E, P_0, \dots, P_{d-1})$$

where $P_i = \{v \in V : \Omega(v) = i\}$.

Parity games with an unbounded number of priorities:

$$(V, V_0, V_1, E, \prec, \mathrm{Odd})$$

where $u \prec v$ iff $\Omega(u) < \Omega(v)$ and Odd is the set of positions with an odd priority.

Parity games with a bounded number of priorities

This case is well understood.

Theorem. The winning region of Player 0 on parity games with d priorities is definable in the modal μ-calculus, by the formula

$$\mathrm{Win}_d := \nu X_0\, \mu X_1\, \nu X_2 \cdots \lambda X_{d-1} \bigvee_{i<d} \Big((V_0 \wedge P_i \wedge \Diamond X_i) \vee (V_1 \wedge P_i \wedge \Box X_i) \Big).$$

Proof. The model checking game for Win_d on $\mathcal{G}$ coincides (up to the presence of additional 'stupid' moves) with the game $\mathcal{G}$ itself ! Hence

$$\text{Player 0 wins } \mathcal{G} \text{ from position } u \quad \Longleftrightarrow \quad \mathcal{G}, u \models \mathrm{Win}_d.$$

Corollary. An efficient model-checking procedure for L_μ gives an efficient algorithm for computing winning regions in parity games.

Alternation hierarchies

The formula Win_d, describing winning positions of Player 0 in parity games with d priorities, has alternation depth d.

Theorem. There is no formula on the modal μ-calculus with alternation depth $< d$ that is equivalent to Win_d.

Corollary. (Bradfield, Lenzi, Arnold)
The alternation hierarchy of the modal μ-calculus is strict.

What about alternation hierarchies for LFP ?

Theorem. On finite structures, the alternation hierarchy of LFP collapses.

Theorem. On arithmetic $\mathfrak{N} = (\omega, +, \cdot)$ the alternation hierarchy of LFP is strict.

Definability of parity games on arbitrary game graphs

GSO: Guarded second-order logic

On (game) graphs this coincides with monadic second-order logic with quantification over **sets of edges** (rather than just sets of nodes)

Theorem. Winning regions of parity games in $\mathcal{PG}$ are definable in GSO

On game graphs $\mathcal{G} = (V, V_0, V_1, E, \prec, \mathrm{Odd})$ we express that x is in the winning region of Player 0 as follows:

$(\exists S \subseteq E)$ such that

(1) S is a positional strategy of Player 0

(2) no S-path from x gets stuck at a vertex $y \in V_0$

(3) for every $y \in \mathrm{Odd}$, every S-path from x either contains infinitely many nodes $z \prec y$, or only finitely many nodes with the same priority as y.

(1) is a first-order formula, and (2) and (3) are expressible in monadic LFP, and therefore also in MSO. Since the non-monadic quantifier $(\exists S \subseteq E)$ is guarded, the formula is in GSO.

Definability of parity games on arbitrary game graphs (2)

Fragments of second-order logic:

Σ^1_2: Properties expressible by $\exists R_1 \ldots \exists R_k \forall T_1 \ldots \forall T_m \varphi$ with $\varphi \in$ FO

Π^1_2: similarly, for formulae $\forall R_1 \ldots \exists A_k \exists T_1 \ldots \exists T_m \varphi$

$\Delta^1_2 := \Sigma^1_2 \cap \Pi^1_2$

Fact. (Dawar, Gurevich) $\quad$ LFP $\subseteq \Delta^1_2$

Corollary. $\quad$ Winning regions of parity games in $\mathcal{PG}$ are definable in Δ^1_2.

We have seen that winning regions are definable by formulae $\exists S \varphi(S, x)$ with $\varphi \in$ LFP. This gives us Σ^1_2-definitions. By taking the formula for the opposite player (and using determinacy), we also get Π^1_2-definitions.

Non-definability in LFP

Theorem. Winning regions of parity games in $\mathcal{PG}$ are **not** definable in LFP, even under the assumption that the game graph is countable and the number of priorities is finite.

Suppose that $\mathrm{Win}(x) \in \mathrm{LFP}$ defines the winning region of Player 0 on $\mathcal{PG}$. Use this formula to solve the model checking problem for LFP on $\mathfrak{N} = (\omega, +, \cdot)$.

For any $\varphi(x) \in \mathrm{LFP}$, we have a parity game $\mathcal{G}(\mathfrak{N}, \varphi)$ such that, for all n

$$\mathfrak{N} \models \varphi(n) \quad \Longleftrightarrow \quad \mathcal{G}(\mathfrak{N}, \varphi) \models \mathrm{Win}(v_n)$$

(where v_n is the initial position associated with $\varphi(n)$)

Note: The model checking game $\mathcal{G}(\mathfrak{N}, \varphi)$ is first-order interpretable in $\mathfrak{N}$.

Non-definability in LFP

Hence the formula $\mathrm{Win}(x)$ is mapped, via a first-order translation $\mathfrak{I}_\varphi$, into another LFP-formula $\mathrm{Win}_\varphi(x)$ such that

$$\mathcal{G}(\mathfrak{N}, \varphi) \models \mathrm{Win}(v_n) \quad \Longleftrightarrow \quad \mathfrak{N} \models \mathrm{Win}_\varphi(n).$$

Note that the first-order translation $\mathrm{Win}(x) \mapsto \mathrm{Win}_\varphi(x)$ depends on φ, but does not increase the alternation depth. Hence, on arithmetic, every formula $\varphi(x)$ would be equivalent to one of fixed alternation depth:

$$\mathfrak{N} \models \varphi(n) \quad \Longleftrightarrow \mathfrak{N} \models \mathrm{Win}_\varphi(n).$$

However, it is known that the alternation hierarchy of LFP on arithmetic is strict.

Definability of parity games on finite game graphs

Definability issues on finite game graphs are different and closely related to complexity. For instance, winning regions are definable in Δ^1_1 (rather than Δ^1_2) because they are computable in NP $\cap$ Co-NP.

Question. Are the winning regions of parity games on finite game graphs definable in the modal μ-calculus, or in LFP?

Clearly, a positive answer would imply that parity games are solvable in polynomial time.

Theorem. Winning regions on $\mathcal{PG}$ are **not** definable in the modal μ-calculus.

Theorem. Winning regions are on $\mathcal{PG}$ are definable in LFP if, and **only if** they are computable in polynomial time.

The μ-calculus on $\mathcal{PG}$

On $\mathcal{PG}$, priorities are encoded via a pre-order $\prec$ on the nodes. In modal logics, binary relations are handled via modal operators.

Hence, besides the modal operators $\Diamond$ and $\Box$ related to the moves in the game, we have modal operators $\langle \prec \rangle$ and $[\prec]$.

$\langle \prec \rangle \varphi$ holds at a node x if φ holds at a node y with lower priority than x (we view $\prec$-edges as going downwards).

To make such a μ-calculus more powerful, we can also admit modalities for $\preceq$, $\sim$, $\succeq$ and/or the associated successor relations. However, precise definitions do not matter since **no formula in any such μ-calculus can define winning regions of parity games**.

Non-definability in the μ-calculus

Theorem. Winning regions on $\mathcal{PG}$ are **not** definable in the modal μ-calculus.

Proof. Suppose we had such a μ-calculus formula ψ, with alternation depth m.

For any fixed number d, we can translate ψ into a formula ψ_d of the usual μ-calculus, defining winning regions of parity games with d priorities, encoded by predicates $P_0, \dots P_{d-1}$.

Replace $\langle \prec \rangle \varphi$ by

$$\bigvee_{0 \le i < j < d} P_j \wedge \mu X.((P_i \wedge \varphi) \vee \Diamond X)$$

(there is a reachable node of lower priority at which φ holds).

The translation increases the alternation depth at most by one.

Non-definability in the μ-calculus

Take any strongly connected parity game $\mathcal{G} \in \mathcal{PG}$ with d priorities and let $\mathcal{G}'$ be its presentation with predicates $P_0, \ldots P_{d-1}$. Clearly

$$\mathcal{G}, v \models \psi \quad \Longleftrightarrow \quad \mathcal{G}', v \models \psi_d.$$

Hence, for any d, winning regions of parity games with d priorities on strongly connected, finite game graphs would be definable by a μ-calculus formula with fixed alternation depth $m + 1$.

It is known that this is not true.

Are winning regions definable in LFP?

Assume that winning regions of parity games are computable in polynomial time.

Goal. Show that they are definable in LFP (despite the fact that LFP is, in general, weaker than PTIME).

Idea. Use a theorem by Martin Otto:
The multi-dimensional μ-calculus (which is a fragment of LFP) captures precisely the bisimulation-invariant part of PTIME.

Problem. Winning regions of parity games are invariant under the usual notion of bisimulation, on structures with predicates $P_0, P_1, \ldots$.
To apply Otto's Theorem for parity games with an unbounded number of priorities, we need a different form of bisimulation, namely bisimulation on structures $\mathcal{G} = (V, V_0, V_1, E, \prec, \mathrm{Odd})$.

Bisimulation on structures $\mathcal{G} = (V, V_0, V_1, E, \prec, \text{Odd})$

Let $\tau = (V_0, V_1, E, \prec, \text{Odd})$.

If two finite τ-structures $\mathcal{G}, \mathcal{G}'$ are indeed parity games, then they are bisimilar, as τ-structures, if and only if they are bisimilar in the usual sense.

However,

- not all structures of this form are parity games, and
- the class of parity games is not closed under bisimulation.

To check whether a τ-structure $\mathcal{G} = (V, V_0, V_1, E, \prec, \text{Odd})$ is a parity game, we have to verify that $\prec$ is a pre-order, and Odd is an appropriate union of equivalence classes wrt. $\prec$.

Lemma. A τ-structure is bisimilar to a parity game if, and only if, its bisimulation quotient is a parity game.

Defining winning positions in LFP

Theorem. Let $\mathcal{C}$ be any class of parity games such that winning positions on its bisimulation quotients are decidable in polynomial time. Then, on $\mathcal{C}$, winning regions are LFP-definable.

Let A be a polynomial-time algorithm, which decides winning regions of Player 0 on bisimulations quotients of games in $\mathcal{C}$.

Let B be the algorithm that, given a structure $(\mathcal{G}, v) \in \mathcal{C}$, first computes its bisimulation quotient, and then applies A.

The class X of structures $(\mathcal{G}, v)$ that are accepted by B is invariant under bisimulation (and decidable in PTIME).

Hence, by Otto's theorem there is an LFP-formula $\psi(x)$ such that

$$(\mathcal{G}, v) \in X \quad \Longleftrightarrow \quad \mathcal{G} \models \psi(x).$$

Defining winning positions in LFP

For parity games $\mathcal{G} \in \mathcal{C}$ we further have

$$(\mathcal{G}, v) \in X \quad \Longleftrightarrow \quad \text{Player 0 wins } \mathcal{G} \text{ from } v.$$

Hence $\psi(x)$ defines the winning region of Player 0 for games in $\mathcal{C}$.

Corollary. On the class of **all** finite parity games, winning regions are LFP-definable if, and only if, they are computable in polynomial time.

Let $\mathcal{C}$ be a class of parity games on which a polynomial-time algorithm for computing the winning regions are known. If we can show that the class is closed under taking bisimulation quotients, then we also know that winning regions on $\mathcal{C}$ are LFP-definable.

Entanglement and bisimulation quotients

Lemma. Let G be a directed graph, and $G^\sim$ its bisimulation quotiont. Then $\mathrm{ent}(G^\sim) \leq \mathrm{ent}(G)$.

If the thief has a strategy to escape against k detectives on $G^\sim$, then he has a similar strategy on G, by which he stays clear not only of the nodes occupied by the detectives, but also all nodes bisimilar to these.

Corollary. In any class of parity games of bounded entanglement, winning regions are definable in LFP.

Open problems

Find other classes of parity games, on which winning regions are LFP-definable

(preferably the class of all parity games on finite graphs).

Are winning regions of parity games definable in MSO ?