Modal Fixpoint Logics: When Logic Meets Games, Automata and Topology

Alessandro Facchini & Damian Niwiński

University of Warsaw

Lecture V

1. Complexity aspects of the μ-calculus

2. Probabilistic μ-calculus

3. Perspectives and open problems

ESSLLI Tübingen 2014

Disclaimer. Credits to many authors. Errors (if any) are mine. . .

Part I

Complexity aspects of the μ-calculus

How to evaluate fixed point expressions ?

$$\mu x.\nu y.\Diamond(x \wedge \Box(y \vee \mu z.\Diamond(x \wedge \Box(y \vee z))))$$

Primary problem: compute the value

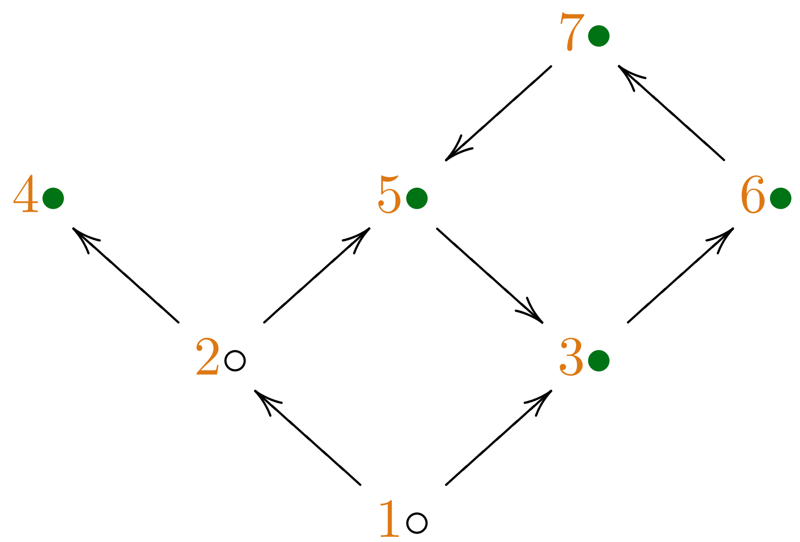
$$\mu x_{2k+1}.\nu x_{2k}.\ldots\mu x_1.\nu x_0.F(x_0, x_1, \ldots, x_{2k}, x_{2k+1})$$

for a monotonic mapping $F : L^{2k+1} \longrightarrow L$ over a finite complete lattice $\langle L, \leq \rangle$.

Typically $L_n = \langle \wp\{1, \ldots, n\}, \subseteq \rangle \equiv \langle \{0, 1\}^n, \leq \rangle$.

Note: $L_k \times L_m \equiv L_{k+m}$.

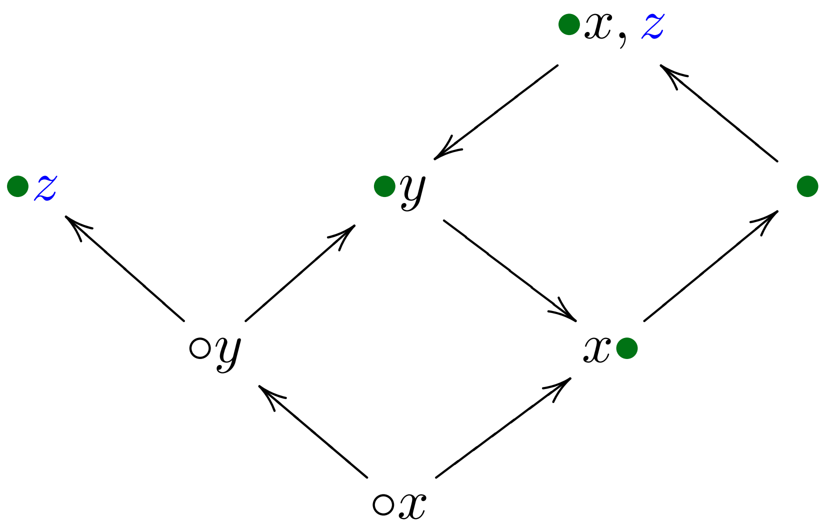
Example revisited



$$\mu x.\nu y.\,\underbrace{\Box y \wedge (\mathit{Happy} \vee \Box x)}_{F(x,y)}$$

$$F : \underbrace{1010001}_{x}, \underbrace{0100100}_{y} \mapsto ?$$

Example revisited



$$F(x, y) = \Box y \wedge (Happy \vee \Box x)$$

$$F : \underbrace{1010001}_{x}, \underbrace{0100100}_{y} \mapsto \underbrace{0001001}_{z}$$

Iteration algorithm

$G : L \to L$. Compute $\mu x.G(x)$.

$$
\begin{array}{ccc}
G(\bot) & = & a_1 \\
G(a_1) & = & a_2 \\
\ldots & \ldots & \ldots \\
G(a_m) & = & a_{m+1} \\
 & & \| \\
 & \textbf{until} & a_m
\end{array}
$$

Then $a_m = \mu x.G(x)$.

$\mathcal{O}(n)$ steps, where n is the **height** of the lattice L (i.e., the length of a maximal chain, the height of $\{0, 1\}^n$ is n).

Iteration algorithm

Compute $\mu x.\nu y.F(x,y)$.

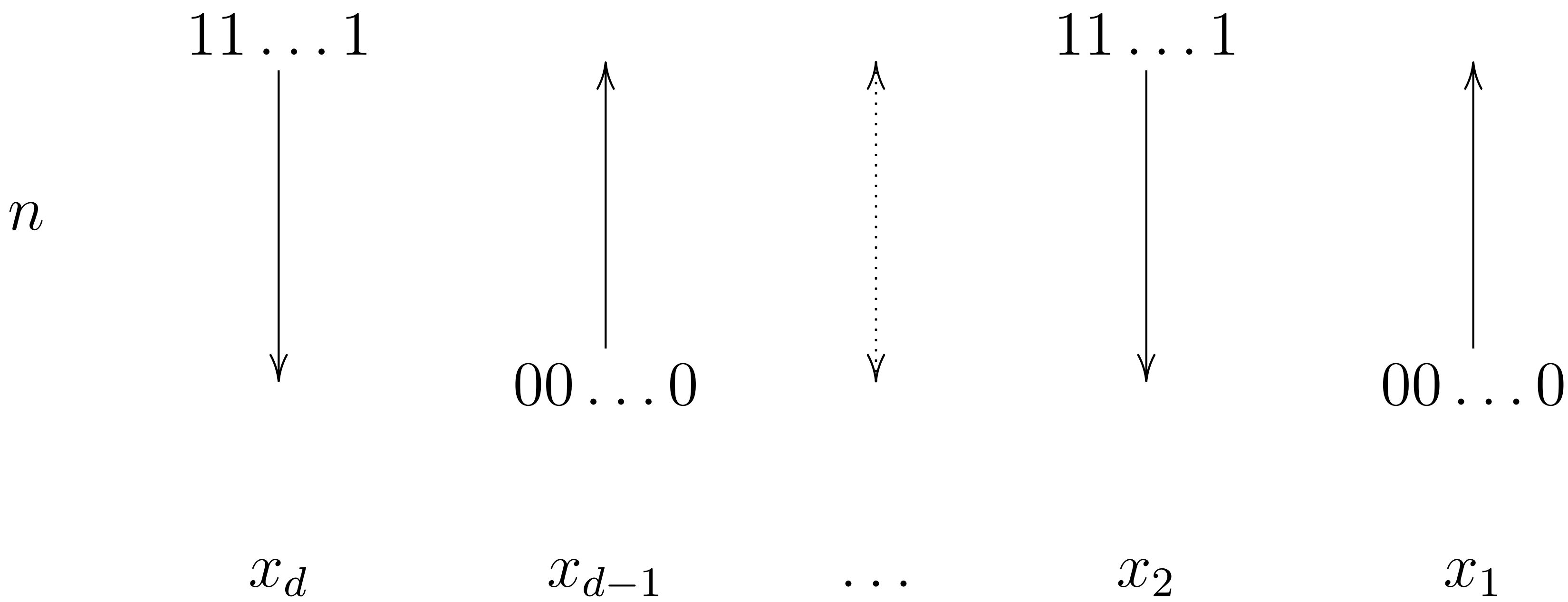
$$
\begin{array}{lllll}
F(\bot,\top) & F(\bot,F(\bot,\top)) & \ldots\ldots & \nu y.F(\bot,y) = & a_1 \\
F(a_1,\top) & F(a_1,F(\bot,\top)) & \ldots\ldots & \nu y.F(a_1,y) = & a_2 \\
\ldots\ldots & \ldots\ldots\ldots\ldots & \ldots\ldots & \ldots\ldots\ldots & \ldots\ldots \\
F(a_m,\top) & F(a_m,F(\bot,\top)) & \ldots\ldots & \nu y.F(a_m,y) = & a_{m+1} \\
 & & & & \| \\
 & & & \textbf{until} & a_m
\end{array}
$$

Then $a_m = \mu x.\nu y.F(x,y)$.

$\mathcal{O}(n^2)$ steps, where n is the height of the lattice.

Black-box algorithms

$$\nu x_d.\mu x_{d-1}.\ldots.\nu x_2.\mu x_1.F(x_1, x_2, \ldots x_d)$$



The (naive) iteration algorithm makes $\mathcal{O}(n^d)$ calls of F.

Calculate F_d

$$\underbrace{\nu x_d.\ \overbrace{\mu x_{d-1}.\dots.\ \underbrace{\nu x_2.\ \overbrace{\mu x_1.F(x_1, x_2, \dots x_d)}^{F_1}}_{F_2}}^{F_{d-1}}}_{F_d}$$

Calculate $F_i(x_{i+1}, \dots, x_d)$ ($i > 0$)

$x_i = 00\dots0$, for odd i / $11\dots1$, for even i

repeat

$x_i =$ **Calculate** $F_{i-1}(x_i, x_{i+1}, \dots, x_d)$

until x_i stops changing

return x_i

Calculate $F_0(x_1, \dots, x_d)$ **return** $F(x_1, \dots, x_d)$

$$\underbrace{\nu x_d . \overbrace{\mu x_{d-1} . \ldots . \underbrace{\nu x_2 . \overbrace{\mu x_1 . F(x_1, x_2, \ldots x_d)}^{F_1}}_{F_2}}^{F_{d-1}}}_{F_d}$$

Calculate $F_i(x_{i+1}, \ldots, x_d)$ ($i > 0$)

% $x_i = 00\ldots0$, for odd i / $11\ldots1$, for even i

Initialize x_i `(closer to the fixpoint)`

repeat

$x_i =$ **Calculate**$F_{i-1}(x_i, x_{i+1}, \ldots, x_d)$

until x_i stops changing

return x_i

We can reduce the number of calls from n^d to $n^{\frac{d}{2}+1}$.

We can reduce the number of calls from n^d to $n^{\frac{d}{2}+1}$ at the expense of increasing computational space.

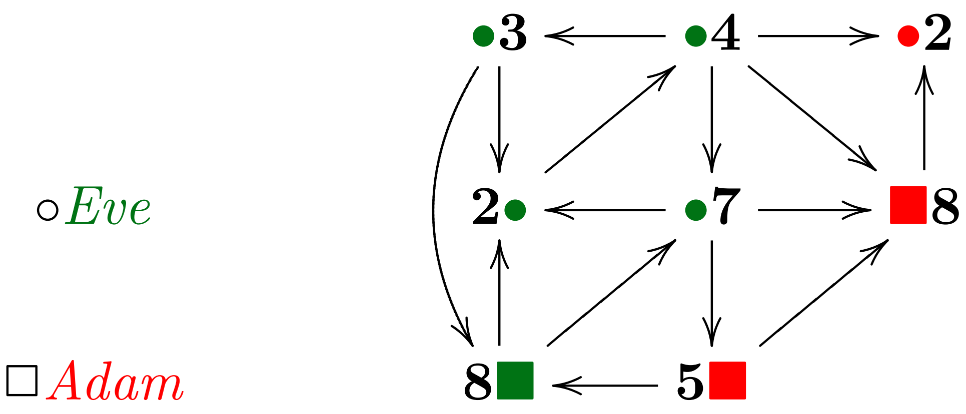
Known **lower bound** in the black box model:
$\Omega\left(\frac{n^2}{\log n}\right)$ iterations; already for $\nu y.\mu x.F(x,y)$.

Question. Is there a $n^{c \cdot d}$ lower bound for each $\nu x_d.\mu x_{d-1}.\ldots.\nu x_2.\mu x_1.F(x_1, x_2, \ldots x_d)$?

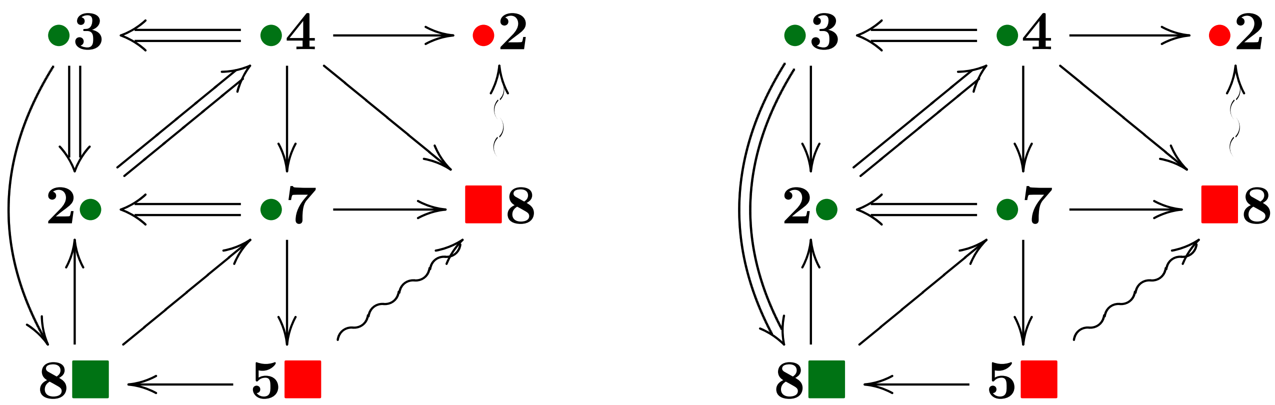
Solving parity games

Recall $s \in [\![\varphi]\!]_{\mathcal{K}}$ (in other words $\mathcal{K}, s \models \varphi$) iff Eve wins the game $\mathcal{G}(\mathcal{K}, \varphi)$ from position (s, φ).

Thus the evaluation of $[\![\varphi]\!]_{\mathcal{K}}$ boils down to computing the winning regions in parity games.

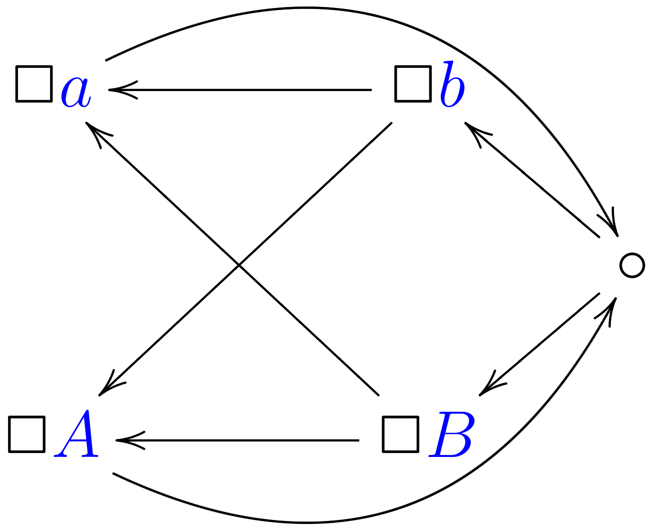


Theorem. (Emerson & Jutla, A.W.Mostowski)
Parity games are **positionally determined**, i.e., the winner may always use a strategy, which depends only on the actual position.



Parenthesis

Not every game is positionally determined.



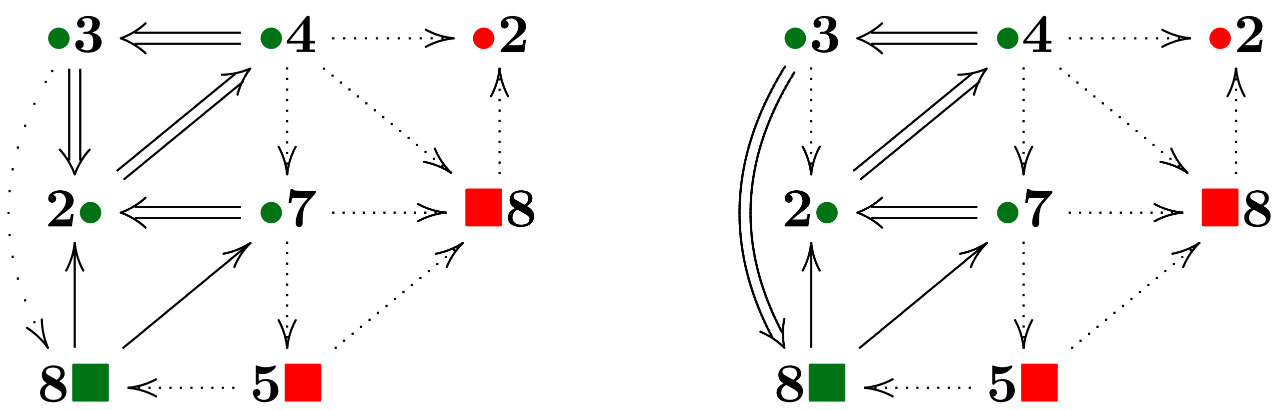
Eve wins if both a, b or both A, B occur infinitely often.

Eve can make it, but one bit of memory is needed.

Winning strategies in parity games

It can be verified in polynomial time whether a positional strategy is winning.

(Check the parity of **max rank** on strongly connected subgraphs of the strategy.)



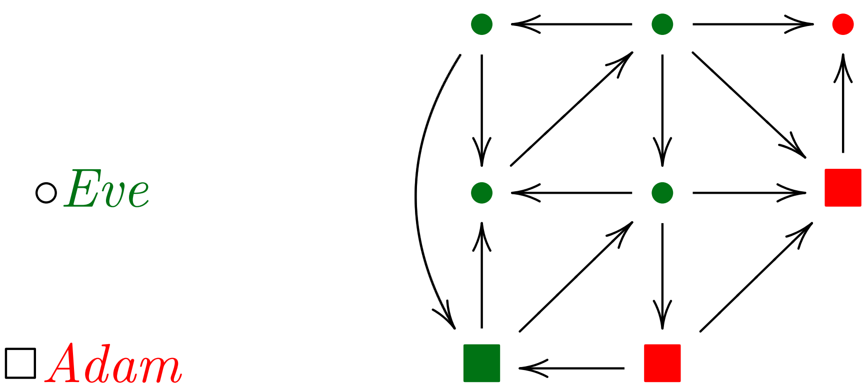
Thus the problem of determining the winning regions in parity games is in **NP** $\cap$ **co-NP**.

It is even in **UP** $\cap$ **co-UP**.

UP $=$ unambiguous **NP**.

Survival game

All ranks are 0, so that Eve wins any infinite play.



The winning region of Eve is the maximal set $W \subseteq Pos$, such that

$$W \subseteq (E \cap \Diamond W) \cup (A \cap \Box W)$$

Note: W is a **fixed point** ($\longrightarrow$ Knaster-Tarski Theorem, 1st lecture).

Eve's (positional) strategy: remain in W.

This game can be solved in **linear time**.

Solving parity games deterministically

Recall $Win_E =$

$$\nu X_8.\mu X_7.\ldots.\mu X_1.\nu X_0.(E \cap rank_0 \cap \Diamond X_0) \cup (E \cap rank_1 \cap \Diamond X_1) \cup \ldots$$

$$\ldots \cup (E \cap rank_7 \cap \Diamond X_7) \cup (E \cap rank_8 \cap \Diamond X_8) \cup$$

$$\cup (A \cap rank_0 \cap \Box X_0) \cup (A \cap rank_1 \cap \Box X_1) \cup \ldots \cup (A \cap rank_8 \cap \Box X_8)$$

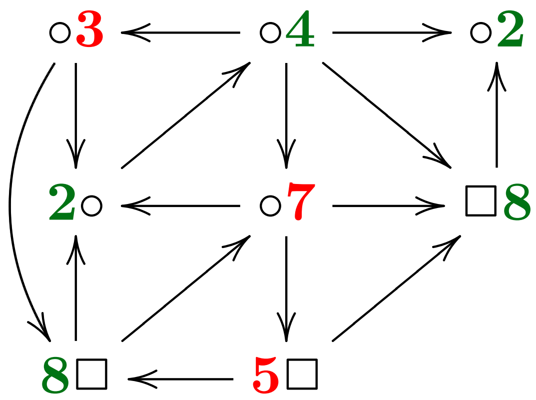
$Win_A =$ has a dual formula.

By the *naive algorithm*, we can compute the winning region in a game with d ranks and n positions in time $n^{d+\mathcal{O}(1)}$ and space $\mathcal{O}(d \cdot n)$.

By improving the naive algorithm ($\longrightarrow$ initialization), we can reduce time to $n^{\frac{d}{2}+\mathcal{O}(1)}$ at the expense of increasing the computation space.

Can we do better ?

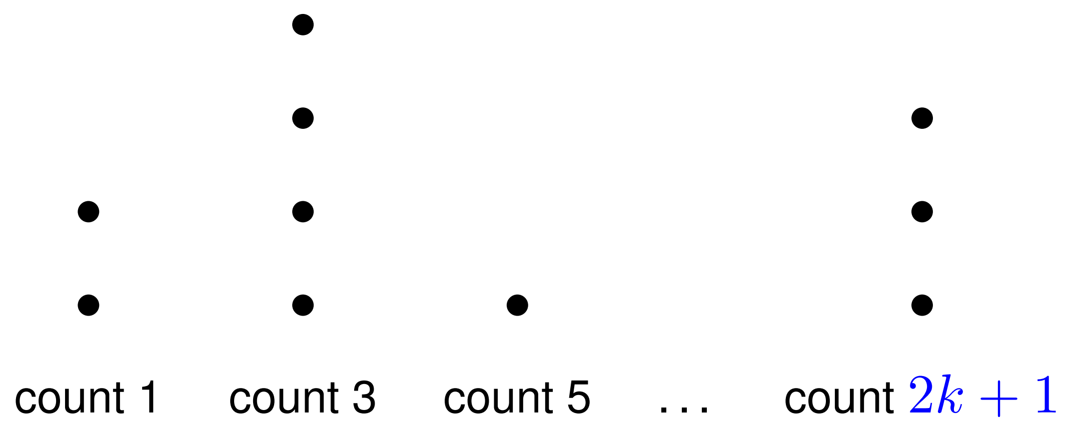
Solving parity games – another view

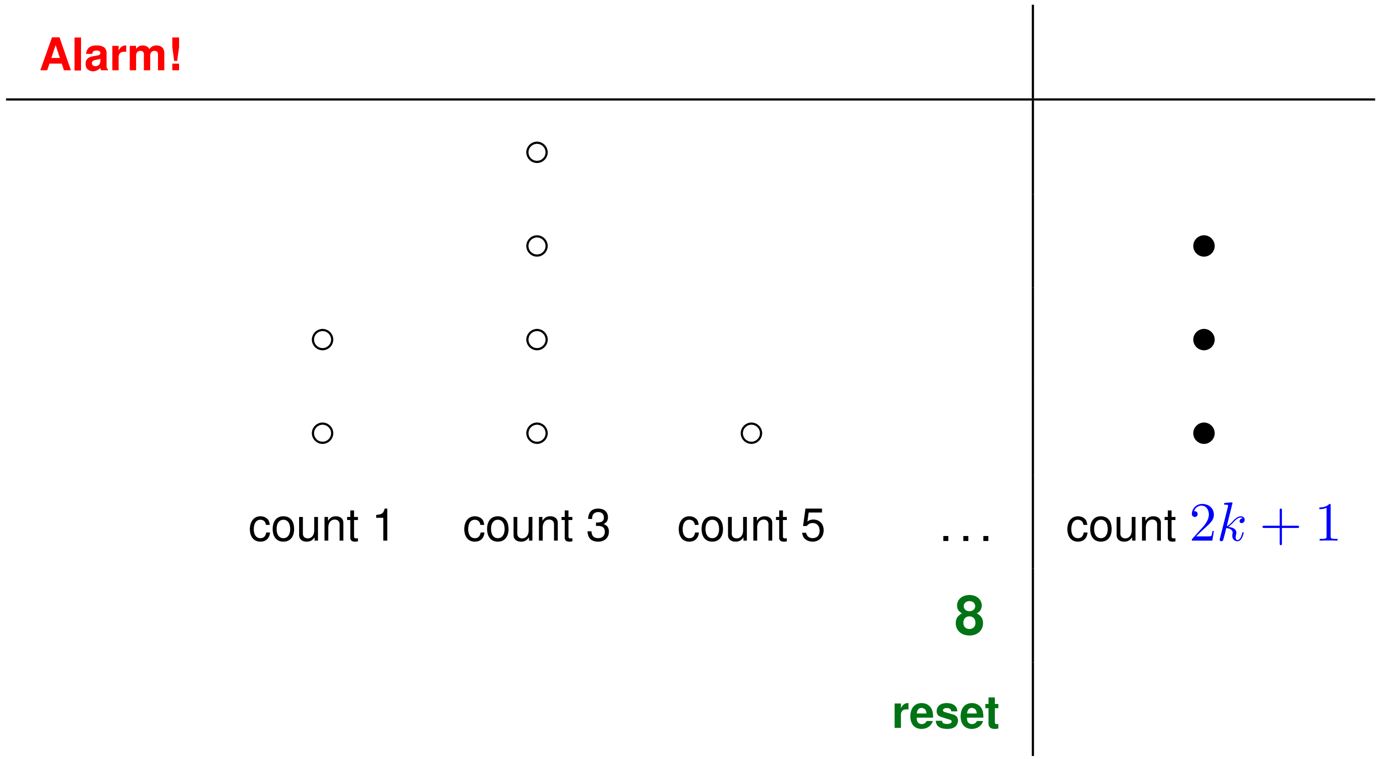


Playing a winning positional strategy, Eve never sees an **odd** rank more than $|Pos|$ times, without seeing some **higher even** rank in the meantime.

Eve never sees an **odd** rank too many times, without seeing some **higher even** rank in the meantime.

Alarm!





Eve wins the game iff Alarm! is never reached.

The game G^+.

For a parity game G with $|Pos| = n$ and ranks in $\{0, 1, \ldots, 2k+1\}$, create

$$Pos \times \{0, 1, \ldots, n\}^{k+1} \cup \{Alarm\}$$

The **update** $rank \times counters \mapsto counters'$:

$$\begin{aligned} up(\mathbf{2i+1}, c_1 c_3 \ldots c_{2k+1}) &= c_1 c_3 \ldots \overbrace{(c_{2i+1} + 1)}^{\leq n} \ldots c_{2k+1} \\ &= \top \text{ otherwise} \\ up(\mathbf{2i}, c_1 c_3 \ldots c_{2k+1}) &= 00 \ldots 0 c_{2i+1} \ldots c_{2k+1} \end{aligned}$$

Moves: if $v \longrightarrow v'$ in G then

$$(v, c_1 \ldots c_{2k+1}) \longrightarrow (v', \overbrace{up(rank(v), c_1 \ldots c_{2k+1})}^{\neq \top})$$

$(v, c_1 \ldots c_{2k+1}) \longrightarrow$ Alarm (Eve looses)

The game G^+ (continued).

Positions: $Pos \times \{0, 1, \ldots, n\}^{k+1} \cup \{Alarm\}$.

Moves: if $v \longrightarrow v'$ in G then

$$(v, c_1 \ldots c_{2k+1}) \longrightarrow (v', \overbrace{up(rank(v), c_1 \ldots c_{2k+1})}^{\neq \top})$$

$\searrow$ Alarm (Eve looses)

G^+ is a survival game.

The following conditions are equivalent.

(i) Eve wins the game G from position p.

(ii) Eve wins the game G^+ from position $(p, \vec{0})$.

The size of G^+

$$|Pos \times \{0,1,\ldots,n\}^{k+1} \cup \{Alarm\}| = n^{\frac{d}{2}+\mathcal{O}(1)}$$

(where $d = 2k+2 =$ the number of ranks in the original game).

It can be solved in linear time, which yields the time

$$n^{\frac{d}{2}+\mathcal{O}(1)}$$

for the original game ☺, but with the computation space of the same order, in contrast to the space $\mathcal{O}(d \cdot n)$ used by the naive algorithm ☹.

The space complexity blow-up can be avoided by using a more “patient” alarming policy.

The game G^{++}.

View elements of $\{0,1,\ldots,n\}^{k+1}$ as $k+1$-digit numbers **in base** $n+1$,

$$a_0 a_1 \ldots a_k \quad = \quad a_0 + a_1 \cdot (n+1) + a_2 \cdot (n+1)^2 + \ldots + a_k \cdot (n+1)^k.$$

Let *Overflow* $= (n+1)^{k+1}$.

$$\begin{aligned} up(\mathbf{2i+1}, \overbrace{a_0 \ldots a_k}^{m}) &= m + (n+1)^i, \text{ if } < \mathit{Overflow} \\ &= \top \text{ otherwise} \\ up(\mathbf{2i}, a_0 \ldots a_{k\,m}) &= 00\ldots0a_i \ldots a_k \end{aligned}$$

Positions and moves in G^{++} are like in G^{+} (with the **new update** function).

G^{++} is a survival game.

The following conditions are equivalent.

(i) Eve wins the game G from position p.

(ii) Eve wins the game G^{++} from position $(p, 0)$.

(iii) $\max(p) > \bot$, where

$$\max(p) = \sup\{x : \text{ Eve wins the game } G^{++} \text{ from } (p, x)\}$$

with $\bot = \sup \emptyset$.

Thus, to solve the original game G, it is enough to compute $\max(p)$, for all positions p.

The algorithm computes $F : Pos \to \{0, 1, \ldots, \mathit{Overflow} - 1\} \cup \{\bot\}$.

For all $p \in Pos$ **do** $F(p) := \mathit{Overflow} - 1$.

While $(\exists p) \neg \mathit{Well}(p, F(p), F)$ **do**

Choose such p.

$F(p) := \sup\{x : \mathit{Well}(p, x, F)\}$.

Return F.

Where

$$
\begin{aligned}
\mathit{Well}(p, x, F) \quad \iff \quad & x = \bot, \text{ or } & p \in Pos_{\exists}, \text{ and} \\
& up(rank(p)x) \leq & \max\{F(q) : q \in Succ(p)\} \\
& \text{or} & p \in Pos_{\forall}, \text{ and} \\
& up(rank(p)x) \leq & \min\{F(q) : q \in Succ(p)\}
\end{aligned}
$$

$F = \max$. Computation time is $n^{\frac{d}{2}+\mathcal{O}(1)}$, space $\mathcal{O}(d \cdot poly(n))$.

Correctness of the algorithm

For all $p \in Pos$ **do** $F(p) := \mathit{Overflow} - 1$.

While $(\exists p) \neg \mathit{Well}(p, F(p), F)$ **do**

Choose such p.

$F(p) := \sup\{x : \mathit{Well}(p, x, F)\}$.

Return F.

$F \overset{??}{=} \max$.

$\max \leq F$. This is an invariant of the computation.

$F \leq \max$. If $(\forall q)\ \mathit{Well}(q, F(q), F)$ and $F(p) > \bot$ then Eve wins in G^{++} from position $(p, F(p))$. Hence $F(p) \leq \max(p)$.

Satisfiability problem for Lμ

Given: φ.

Question: does there exist $\mathcal{K}$ and s, such that $\mathcal{K}, s \models \varphi$?

Reduction

$\varphi \to A_\varphi$ a μ-automaton (alternating) of $\mathcal{O}(|\varphi|)$ states recognizing tree models of φ

$A_\varphi \to A'_\varphi$ an equivalent **non-deterministic** automaton of $2^{\mathcal{O}(|\varphi|)}$ states, but **only** $\mathcal{O}(\varphi)$ ranks (Simulation Theorem)

$A'_\varphi \to G_\varphi$ parity game with $2^{\mathcal{O}(|\varphi|)}$ positions and $\mathcal{O}(\varphi)$ ranks.

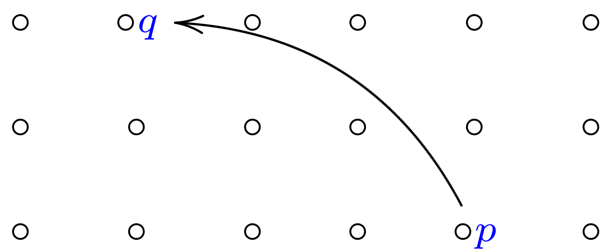
This yields a **single exponential-time** algorithm for the problem.

Part II

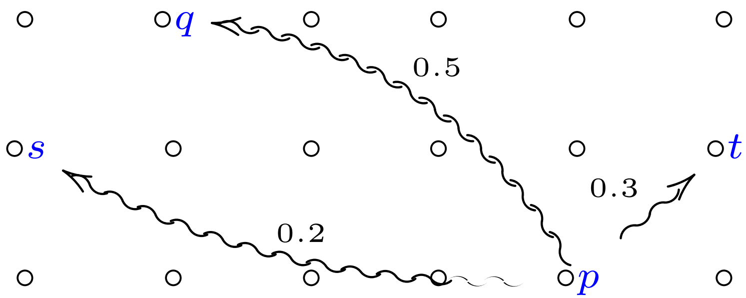
Probabilistic μ-calculus

Probabilistic model of computation

Instaed of a single transition, e.g., $p \to q$



we have a **probabilistic distribution** on all transitions from p



e.g., $d(p,q) = 0.5$, $d(p,s) = 0.2$, $d(p,t) = 0.3$, and $d(p,w) = 0$, for all others w's.

Classical Kripke structure $\mathcal{K} = \langle S, R, \rho \rangle$,

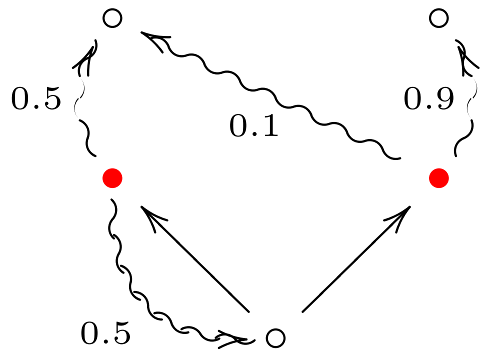
with $R \subseteq S \times S$, and ρ : Prop $\to \wp S$.

Probabilistic Kripke structure $\mathcal{K} = \langle S, \mathcal{R}, \rho \rangle$,

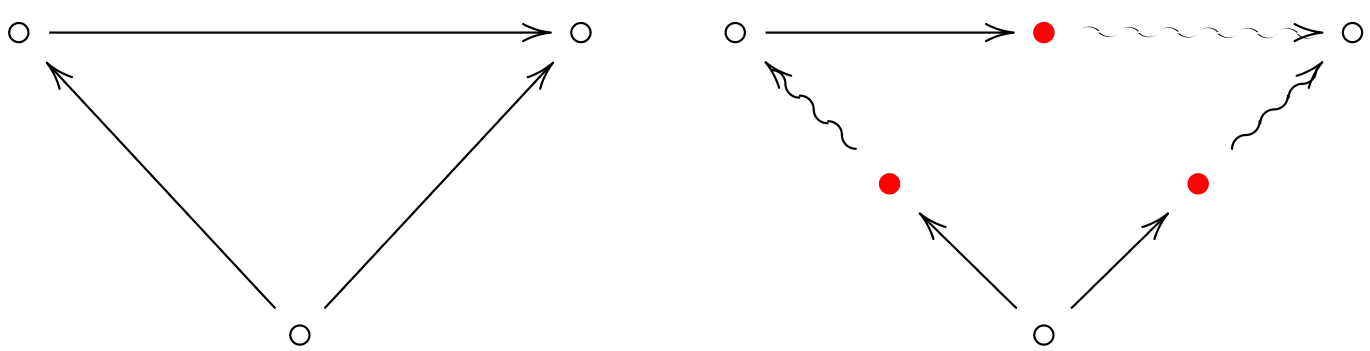
with $\mathcal{R} \subseteq S \times \mathcal{D}(S)$, and ρ : Prop $\to [0,1]^S$.

where $\mathcal{D}(S) = \{d \in [0,1]^S : \sum_{s \in S} d(s) = 1\}$.

From each state, a distribution can be non-deterministically chosen.



Classical Kripke structure as a probabilistic one



Classical interpretation $[\![\varphi]\!]_{\mathcal{K}} \subseteq S$, i.e.,

$[\![\varphi]\!]_{\mathcal{K}} : S \to \{0, 1\}$

Probabilistic interpretation $[\![\varphi]\!]_{\mathcal{K}} : S \to [0, 1]$

Idea: $[\![\varphi]\!]_{\mathcal{K}}(s) =$ probability that φ holds true in s (classically 0 or 1).

More generally, $[\![\varphi]\!]_{\mathcal{K}} v : S \to [0, 1]$, where $v : Prop \to [0, 1]^S$.

Logic pLμ

Model $\mathcal{K} = \langle S, \mathcal{R}, \rho \rangle$, with $\mathcal{R} \subseteq S \times \mathcal{D}(S)$, and ρ : Prop $\to [0,1]^S$, $v : Prop \to [0,1]^S$.

Note: $[0,1]^S$ is a complete lattice, hence Knaster-Tarski's Theorem applies.

Syntax and interpretation

$[\![x]\!]_{\mathcal{K}} v(s) = v(x)(s)$

$[\![p]\!]_{\mathcal{K}} v(s) = \rho(p)(s)$ $\quad$ $[\![\neg p]\!]_{\mathcal{K}} v(s) = 1 - \rho(p)(s)$

$[\![\varphi \vee \psi]\!]_{\mathcal{K}} v(s) = \max([\![\varphi]\!]_{\mathcal{K}} v(s), [\![\psi]\!]_{\mathcal{K}} v(s))$ $\quad$ $[\![\varphi \wedge \psi]\!]_{\mathcal{K}} v(s) = \min([\![\varphi]\!]_{\mathcal{K}} v(s), [\![\psi]\!]_{\mathcal{K}} v(s))$

$[\![\mu x.\varphi]\!]_{\mathcal{K}} v(s) = \mu X.[\![\varphi]\!]_{\mathcal{K}} v[X/x](s)$ $\quad$ $[\![\nu x.\varphi]\!]_{\mathcal{K}} v(s) = \nu X.[\![\varphi]\!]_{\mathcal{K}} v[X/x](s)$

$[\![\Diamond \varphi]\!]_{\mathcal{K}} v(s) = \bigvee \{[\![\varphi]\!]_{\mathcal{K}} v(d) : R(s,d)\}$ $\quad$ $[\![\Box \varphi]\!]_{\mathcal{K}} v(s) = \bigwedge \{[\![\varphi]\!]_{\mathcal{K}} v(d) : R(s,d)\}$

where $[\![\varphi]\!]_{\mathcal{K}} v(d) = \sum_{q \in S} d(q) \cdot [\![\varphi]\!]_{\mathcal{K}} v(q)$ (mean value).

The mappings $\vee, \wedge : \{0,1\}^2 \to \{0,1\}$

Or	0	1
0	0	1
1	1	1

And	0	1
0	0	0
1	0	1

have several (meaningful) extensions to $[0,1]^2 \to [0,1]$, e.g.

Or	**And**
$\max(x,y)$	$\min(x,y)$
$x+y-x\cdot y$	$x\cdot y$
$\min(x+y,1)$	$\max(0,x+y-1)$

Which one to choose ?

The idea of **Matteo Mio** (Ackermann Award 2013):

to combine 2 or 3 operations in one logic.

Logic pL$\mu^{\odot}$

$$
\begin{aligned}
[\![\varphi \odot \psi]\!]_{\mathcal{K}} v(s) &= [\![\varphi]\!]_{\mathcal{K}} v(s) + [\![\psi]\!]_{\mathcal{K}} v(s) - [\![\varphi]\!]_{\mathcal{K}} v(s) \cdot [\![\psi]\!]_{\mathcal{K}} v(s) \\
[\![\varphi \cdot \psi]\!]_{\mathcal{K}} v(s) &= [\![\varphi]\!]_{\mathcal{K}} v(s) \cdot [\![\psi]\!]_{\mathcal{K}} v(s)
\end{aligned}
$$

Logic pL$\mu^{\odot}_{\oplus}$

$$
\begin{aligned}
[\![\varphi \oplus \psi]\!]_{\mathcal{K}} v(s) &= \min\left([\![\varphi]\!]_{\mathcal{K}} v(s) + [\![\psi]\!]_{\mathcal{K}} v(s), 1\right) \\
[\![\varphi \ominus \psi]\!]_{\mathcal{K}} v(s) &= \max\left(0, [\![\varphi]\!]_{\mathcal{K}} v(s) + [\![\psi]\!]_{\mathcal{K}} v(s) - 1\right)
\end{aligned}
$$

Łukasiewicz μ-calculus Łμ Mio & Simpson 2013

pL $\mu_{\oplus}$ extended by $[\![\neg\varphi]\!]_{\mathcal{K}} v(s) = 1 - [\![\varphi]\!]_{\mathcal{K}} v(s)$, plus $[\![\lambda\varphi]\!]_{\mathcal{K}} = \lambda \cdot [\![\varphi]\!]_{\mathcal{K}}$.

Expressive power of pL$\mu^{\odot}$

$$\mathbb{P}_{>0}\varphi \stackrel{def}{=} \mu x.(\varphi \odot x) \qquad\qquad \mathbb{P}_{=1}\varphi \stackrel{def}{=} \nu y.(\varphi \cdot y)$$

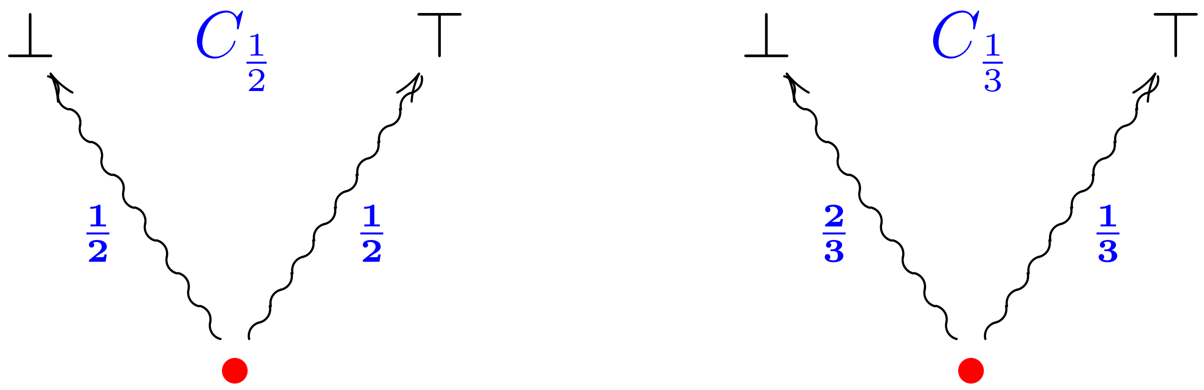
where

$$\begin{aligned} [\![\mathbb{P}_{>0}\varphi]\!]_{\mathcal{K}}(s) &= 1 && \text{if } [\![\varphi]\!]_{\mathcal{K}}(s) > 0 \\ &= 0 && \text{otherwise} \end{aligned}$$

$$\begin{aligned} [\![\mathbb{P}_{=1}\varphi]\!]_{\mathcal{K}}(s) &= 1 && \text{if } [\![\varphi]\!]_{\mathcal{K}}(s) = 1 \\ &= 0 && \text{otherwise} \end{aligned}$$

In particular, **pL**$\mu^{\odot}$ subsumes the probabilistic version of **CTL**.

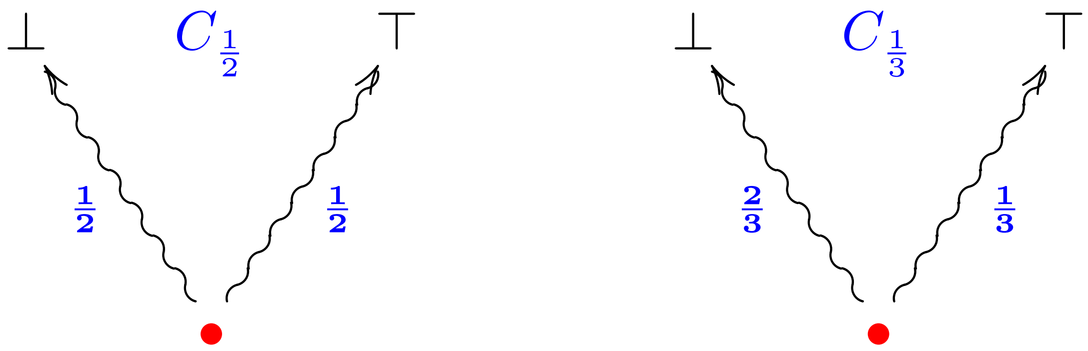
Example (Mio)



Game $C_{\frac{1}{2}} \vee C_{\frac{1}{3}}$ Eve selects a game $C_{\frac{1}{2}}$ or $C_{\frac{1}{3}}$, and this game is played.

Game $C_{\frac{1}{2}} \odot C_{\frac{1}{3}}$ Both games are played independently, and Eve wins if she wins in at least one of them.

Example continued



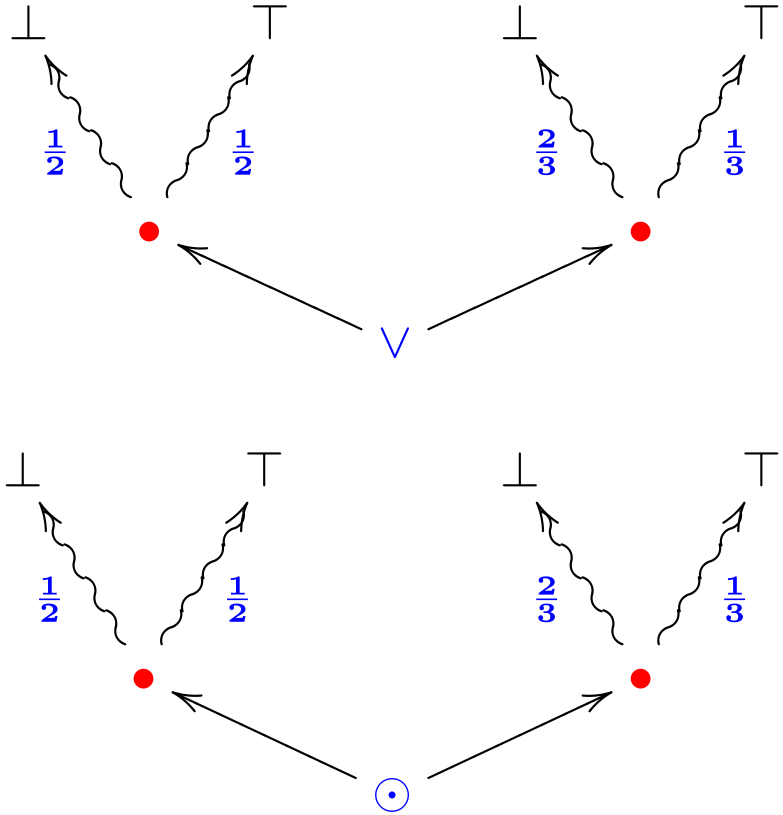
Let $\mathbb{P}_\exists(C) =$ probability that Eve wins the game C.

$$\mathbb{P}_\exists(C_{\frac{1}{2}} \vee C_{\frac{1}{3}}) = \frac{1}{2} = \max\left(\mathbb{P}_\exists(C_{\frac{1}{2}}), \mathbb{P}_\exists(C_{\frac{1}{3}})\right)$$

$$\mathbb{P}_\exists(C_{\frac{1}{2}} \odot C_{\frac{1}{3}}) = 1 - \frac{1}{2} \cdot \frac{2}{3} = \frac{2}{3} = \mathbb{P}_\exists(C_{\frac{1}{2}}) \odot \mathbb{P}_\exists(C_{\frac{1}{3}})$$

Example continued

This suggests the game interpretation of the connectives $\vee$ and $\odot$.



Game semantics for pL$\mu^{\odot}$: stochastic meta-parity games (Mio 2012)

The arena of the **outer game** comprises Eve's positions, Adam's positions, random positions, and **branching positions**.

The result of the game is a **tree**, not a path.

This tree is an arena of an **inner game**, which is a standard **parity game**.

The branching nodes (of the outer game) are assigned to Eve or Adam in the inner game.

Who wins the inner game, wins the whole game.

The original proof by Mio 2012 of **determinacy** of these games involved Martin's Axiom; eliminated in 2014.

Future directions and open problems
in the μ calculs.

Number of iterations

An ordinal α is a **convergence limit** of a formula $\mu x.\varphi$ if, for any model $\mathcal{K}$,

$$[\![\mu x.\varphi]\!]_{\mathcal{K}} = \bigvee_{\xi<\alpha} [\![\varphi]\!]^{\xi}_{\mathcal{K}}(\emptyset),$$

and, for some model, the number α of iterations is required.

Here we view $[\![\varphi]\!]_{\mathcal{K}} : \wp S \to \wp S$, with $[\![\varphi]\!]_{\mathcal{K}}(Z) = [\![\varphi]\!]_{\mathcal{K}}[Z/x]$.

E.g., $\mu x.\Diamond x \vee p$ has the convergence limit ω, but $\mu x.\Box x$ has **no** convergence limit. (It holds in a well founded tree of any height.)

The formula $\mu x.(\Diamond x \wedge \Box p \wedge p) \vee (\Box x \wedge \Box p \wedge \neg p) \vee \Box\bot$ has convergence limit $\omega + 1$ (suggested by M.Bojańczyk). M.Czarnecki showed that, for any $\alpha < \omega^2$, there is a formula with convergence limit α.

Conjecture. There are **no** formulas with convergence limit $\alpha \geq \omega^2$. That is, if a formula requires $\geq \omega^2$ in steps in some model then it may require arbitrary many steps in some model.

Algorithms and complexity

The **model checking** problem for Lμ is

Given $\mathcal{K}, \varphi$.

Question $\mathcal{K} \models \varphi$ **?**

This problem is polynomially equivalent to solving **parity games**.

Is there a polynomial algorithm to solve parity games ?

The best known upper bound is $n^{\mathcal{O}(\sqrt{n})}$.

The **expression complexity** is a problem for a fixed $\mathcal{K}$.

Given φ.

Question $\mathcal{K} \models \varphi$ **?**

Is this problem in P ?

Is there a lower bound over $\Omega(n^2)$ for the **black box model** ?

Algorithms and complexity continued

It is known *via* reduction to simple stochastic games that the parity game problem is in the class **PPAD**. Give a direct proof of this result.

It is known that, for graphs with bounded *tree width* (J.Obdrzalek), the problem is polynomial, however no FPT algorithm is known. Is the problem **FTP** tractable ?

Decidability of the hierarchy

Given an Lμ-formula φ, can we compute the **minimal alternation depth** of a formula $\psi \equiv \varphi$?

An analogous question for the powerset algebra of trees.

Probabilistic μ-calculus

What is the complexity of the model-checking problem ?

Decidability known for Łμ.

What is the expressive power of the probabilistic μ-calculi compared to **PCTL*** ?

Fixpoint logics in the general picture

Does the Janin-Walukiewicz Theorem hold for finite structures ?

Is there an analogous thoerem for probabilistic μ-calculus ?

Can one extend the μ-calculus to non-monotonic operations ?

References – Part I

J.Bernet, D.Janin and I.Walukiewicz. *Permissive strategies: from parity games to safety games.* RAIRO – ITA 36(3), 2002.

E.A. Emerson and C.S. Jutla. *The Complexity of Tree Automata and Logics of Programs*. FOCS, 1988.

E.A. Emerson, C.S. Jutla, and A.P. Sistla. *On model-checking for fragments of the μ-calculus*. CAV 1993.

M. Jurdziński. *Deciding the Winner in Parity Games is in* UP $\cap$ co-Up. IPL 68(3), 1998.

M. Jurdziński. *Small Progress Measures for Solving Parity Games*. STACS 2000.

M. Jurdziński, M.Paterson, U.Zwick. *A Deterministic Subexponential Algorithm for Solving Parity Games*. SIAM J. Comput. 38(4), 2008.

D.E. Long, A. Browne, E.M. Clarke, S. Jha, and W.R. Marrero. *An improved algorithm for the evaluation of fixpoint expressions*. CAV 1994.

P.Parys. *Some results on complexity of μ-calculus evaluation in the black-box model*. RAIRO – Theor. Inf. and Applic. 2013.

References – Part II

A. Bianco and L. de Alfaro. *Model checking of probabilistic and nondeterministic systems*. FSTTCS 1995.

D. Fischer, E. Grädel, Ł. Kaiser. *Model Checking Games for the Quantitative -Calculus*. Theory Comput. Syst. 47(3), 2010.

T. Gogacz, H. Michalewski, M. Mio, and M. Skrzypczak. *Measure Properties of Game Tree Languages*. MFCS 2014.

M. Huth and M. Kwiatkowska. *Quantitative analysis and model checking*. LICS, 1997.

M. Mio. *Game Semantics for Probabilistic μ-Calculi*. PhD thesis, Univ. Edinburgh, 2012.

M. Mio and A. Simpson. *Łukasiewicz μ-calculus.* FICS 2013.

C. Morgan and A. McIver. *A probabilistic temporal calculus based on expectations*. In Proc. Formal Methods, 1997.