Countdown logic, games and automata

bisimulation-invariant approach to (un)boundedness

Jędrzej Kołodziejski
(& Bartek Klin)

4 VII 2023
eTokio

μ-calculus = modal logic + fixpoints

Syntax:

Syntax:

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

Syntax:

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

Syntax:

a, b from fixed Act

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

Syntax:

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

Syntax:

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

Semantics:

Syntax:

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

- interpreted in points of a modal model $\mathcal{M}$

Semantics:

Syntax:

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

- interpreted in points of a modal model $\mathcal{M}$

directed graph $(M, (\xrightarrow{a})_{a \in \mathsf{Act}})$, edges labelled with Act

Semantics:

Syntax:

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

- interpreted in points of a modal model $\mathcal{M}$

directed graph $(M, (\xrightarrow{a})_{a \in \mathsf{Act}})$, edges labelled with Act

- "$\langle a \rangle \varphi$" means "there exists an a-child satisfying φ"

Semantics:

Syntax:

if trivial $\mathsf{Act} = \{a\}$, denote

$\langle a \rangle = \Diamond$ and $[a] = \Box$

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \overbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

- interpreted in points of a modal model $\mathcal{M}$

directed graph $(M, (\xrightarrow{a})_{a \in \mathsf{Act}})$, edges labelled with Act

- "$\langle a \rangle \varphi$" means "there exists an a-child satisfying φ"

Semantics:

Syntax:

if trivial Act $= \{a\}$, denote

$\langle a \rangle = \Diamond$ and $[a] = \Box$

a, b from fixed Act

and x from fixed Var

$$\varphi ::= \underbrace{\top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi}^{\text{boolean}} \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

- interpreted in points of a modal model $\mathcal{M}$

directed graph $(M, (\xrightarrow{a})_{a \in \mathsf{Act}})$, edges labelled with Act

plus val : Var $\to \mathcal{P}(M)$

- "$\langle a \rangle \varphi$" means "there exists an a-child satisfying φ"

Semantics:

- $\Diamond x$ induces an operation $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x ::= S} = \{\mathsf{m} \mid \exists_{\mathsf{m} \to \mathsf{n}} \mathsf{n} \in S\}$$

- $\Diamond x$ induces an operation $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$

- since x appears only positively in $\Diamond x$, F is monotone...

- $\Diamond x$ induces an operation $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

$$S \stackrel{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$

- since x appears only positively in $\Diamond x$, F is monotone...

$$S \subseteq S' \implies F(S) \subseteq F(S')$$

- $\Diamond x$ induces an operation $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m} \to \mathsf{n}} \mathsf{n} \in S\}$$

- since x appears only positively in $\Diamond x$, F is monotone...

$$S \subseteq S' \implies F(S) \subseteq F(S')$$

- ...and so F has the greatest and the least fixpoint!

- $\Diamond x$ induces an operation $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$

- since x appears only positively in $\Diamond x$, F is monotone...

$$S \subseteq S' \implies F(S) \subseteq F(S')$$

- ...and so F has the greatest and the least fixpoint!

$$[\![\nu x.\Diamond x]\!] = \mathrm{GFP}.F$$

- $\Diamond x$ induces an operation $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$

- since x appears only positively in $\Diamond x$, F is monotone...

$$S \subseteq S' \implies F(S) \subseteq F(S')$$

- ...and so F has the greatest and the least fixpoint!

$$[\![\nu x.\Diamond x]\!] = \mathrm{GFP}.F \qquad [\![\mu x.\Diamond x]\!] = \mathrm{LFP}.F$$

Knaster-Tarski Theorem:

Knaster-Tarski Theorem:

Every monotone map $F : \mathcal{P}(M) \to \mathcal{P}(M)$ has the least and the greatest (w.r.t. $\subseteq$) fixpoint LFP.F and GFP.F.

Knaster-Tarski Theorem:

Every monotone map $F : \mathcal{P}(M) \to \mathcal{P}(M)$ has the least and the greatest (w.r.t. $\subseteq$) fixpoint LFP.F and GFP.F.

Both are computed as the limits of (transfinite) sequences:

Knaster-Tarski Theorem:

Every monotone map $F : \mathcal{P}(M) \to \mathcal{P}(M)$ has the least and the greatest (w.r.t. $\subseteq$) fixpoint LFP.F and GFP.F.

Both are computed as the limits of (transfinite) sequences:

$$F_\mu^\alpha = \bigcup_{\beta<\alpha} F_\mu^\beta \qquad \text{and} \qquad F_\nu^\alpha = \bigcap_{\beta<\alpha} F_\nu^\beta$$

with α ranging over ordinal numbers

Knaster-Tarski Theorem:

Every monotone map $F : \mathcal{P}(M) \to \mathcal{P}(M)$ has the least and the greatest (w.r.t. $\subseteq$) fixpoint LFP.F and GFP.F.

Both are computed as the limits of (transfinite) sequences:

$$F_\mu^\alpha = \bigcup_{\beta<\alpha} F_\mu^\beta \qquad \text{and} \qquad F_\nu^\alpha = \bigcap_{\beta<\alpha} F_\nu^\beta$$

with α ranging over ordinal numbers

(note: $F_\mu^0 = \bigcup \emptyset = \emptyset$ and $F_\nu^0 = \bigcap \emptyset = M$)

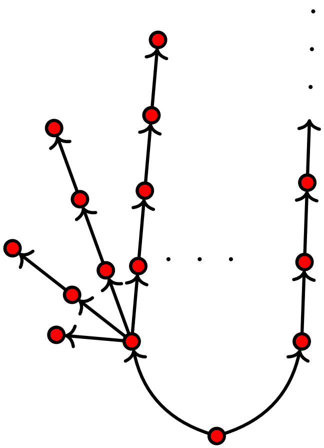
- we compute fixpoints by (transfinite) iteration of F:

- we compute fixpoints by (transfinite) iteration of F:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x ::= S} = \{\mathsf{m} \mid \exists_{\mathsf{m} \to \mathsf{n}} \mathsf{n} \in S\}$$

- we compute fixpoints by (transfinite) iteration of F:

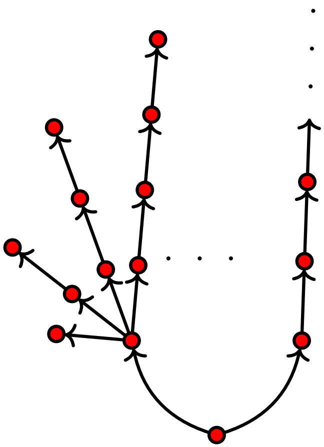
$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$



$F_\nu^0 = M$

- we compute fixpoints by (transfinite) iteration of F:

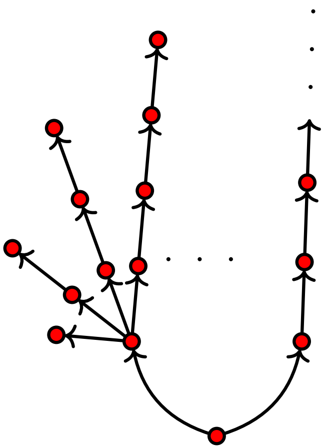
$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}}\mathsf{n} \in S\}$$



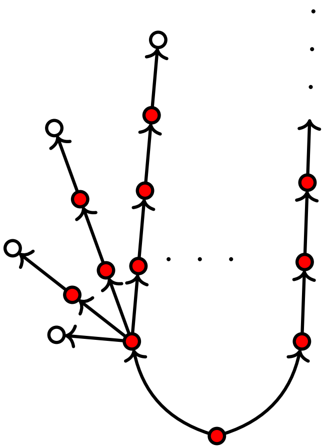
$F^0_\nu = M$ $\qquad$ $F^1_\nu = F(F^0_\nu)$

- we compute fixpoints by (transfinite) iteration of F:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x ::= S} = \{\mathsf{m} \mid \exists_{\mathsf{m} \to \mathsf{n}} \mathsf{n} \in S\}$$



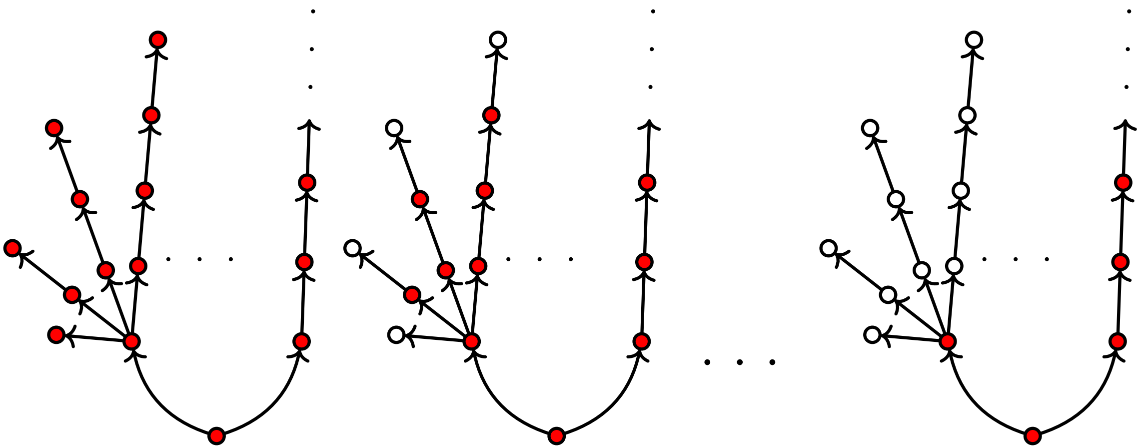
$F_\nu^0 = M$



$F_\nu^1 = F(F_\nu^0) =$ "$\exists$ a child"

- we compute fixpoints by (transfinite) iteration of F:

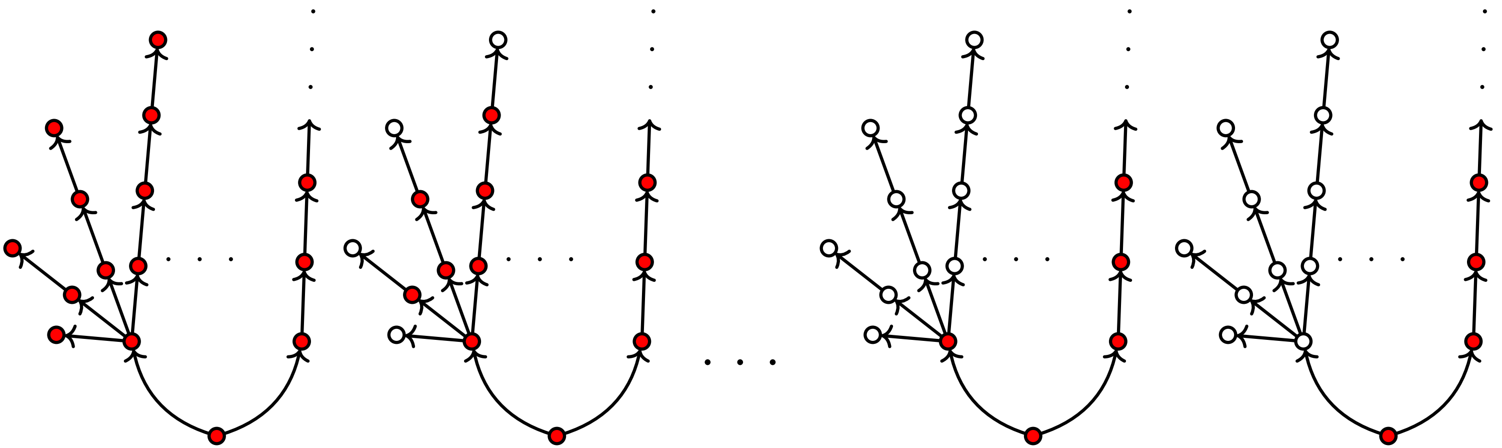
$$S \stackrel{F}{\mapsto} [\![\Diamond x]\!]^{x ::= S} = \{\mathsf{m} \mid \exists_{\mathsf{m} \to \mathsf{n}} \mathsf{n} \in S\}$$



$F^0_\nu = M$ $\quad$ $F^1_\nu = F(F^0_\nu) =$"$\exists$ a child" $\quad$ $F^\omega_\nu =$ "arbitrarily long paths"

- we compute fixpoints by (transfinite) iteration of F:

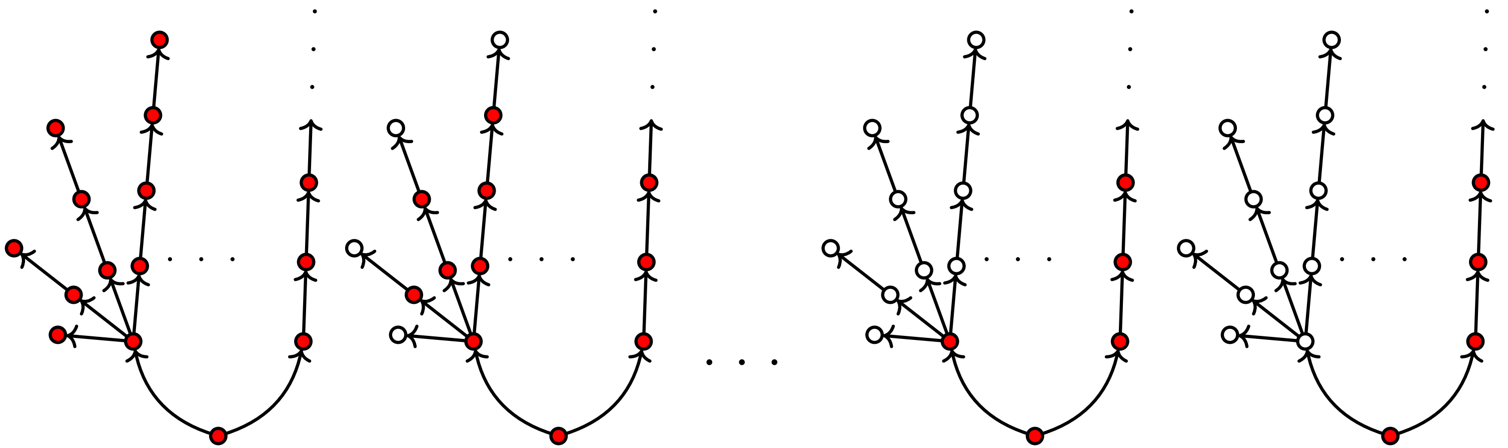
$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x ::= S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$



$F_\nu^0 = M$ $\quad$ $F_\nu^1 = F(F_\nu^0) =$ "$\exists$ a child" $\quad$ $F_\nu^\omega =$ "arbitrarily long paths" $\quad$ $F_\nu^{\omega+1} = F_\nu^{\omega+2} = \mathrm{GFP}.F$

▶ we compute fixpoints by (transfinite) iteration of F:

$$S \overset{F}{\mapsto} [\![\Diamond x]\!]^{x::=S} = \{\mathsf{m} \mid \exists_{\mathsf{m}\to\mathsf{n}} \mathsf{n} \in S\}$$



$F_\nu^0 = M$ $\quad$ $F_\nu^1 = F(F_\nu^0) =$ "$\exists$ a child" $\quad$ $F_\nu^\omega =$ "arbitrarily long paths" $\quad$ $F_\nu^{\omega+1} = F_\nu^{\omega+2} = \mathrm{GFP}.F$

$$\| \\ [\![\nu x.\Diamond x]\!]$$

μ-calculus = modal logic + fixpoints

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid \mathsf{x} \mid \mu\mathsf{x}.\varphi \mid \nu\mathsf{x}.\varphi$$

<u>μ-calculus = modal logic + fixpoints</u>

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

μ-calculus = modal logic + fixpoints

$$[\![\top]\!]^{\mathsf{val}} = M \quad \text{and} \quad [\![\bot]\!]^{\mathsf{val}} = \emptyset$$

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

<u>μ-calculus = modal logic + fixpoints</u>

$$[\![\top]\!]^{\mathsf{val}} = M \quad \text{and} \quad [\![\bot]\!]^{\mathsf{val}} = \emptyset$$

$$[\![\varphi_1 \vee \varphi_2]\!]^{\mathsf{val}} = [\![\varphi_1]\!]^{\mathsf{val}} \cup [\![\varphi_2]\!]^{\mathsf{val}} \quad \text{and} \quad [\![\varphi_1 \wedge \varphi_2]\!]^{\mathsf{val}} = [\![\varphi_1]\!]^{\mathsf{val}} \cap [\![\varphi_2]\!]^{\mathsf{val}}$$

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

μ-calculus = modal logic + fixpoints

$$\llbracket \top \rrbracket^{\mathsf{val}} = M \quad \text{and} \quad \llbracket \bot \rrbracket^{\mathsf{val}} = \emptyset$$

$$\llbracket \varphi_1 \vee \varphi_2 \rrbracket^{\mathsf{val}} = \llbracket \varphi_1 \rrbracket^{\mathsf{val}} \cup \llbracket \varphi_2 \rrbracket^{\mathsf{val}} \quad \text{and} \quad \llbracket \varphi_1 \wedge \varphi_2 \rrbracket^{\mathsf{val}} = \llbracket \varphi_1 \rrbracket^{\mathsf{val}} \cap \llbracket \varphi_2 \rrbracket^{\mathsf{val}}$$

$$\llbracket \langle a \rangle \varphi \rrbracket^{\mathsf{val}} = \{ m \in M \mid \exists_{m \xrightarrow{a} n}\; n \in \llbracket \varphi \rrbracket^{\mathsf{val}} \} \quad \text{and} \quad \llbracket [a] \varphi \rrbracket^{\mathsf{val}} = \{ m \in M \mid \forall_{m \xrightarrow{a} n}\; n \in \llbracket \varphi \rrbracket^{\mathsf{val}} \}$$

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [b]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

μ-calculus = modal logic + fixpoints

$$[\![\top]\!]^{val} = M \quad \text{and} \quad [\![\bot]\!]^{val} = \emptyset$$

$$[\![\varphi_1 \vee \varphi_2]\!]^{val} = [\![\varphi_1]\!]^{val} \cup [\![\varphi_2]\!]^{val} \quad \text{and} \quad [\![\varphi_1 \wedge \varphi_2]\!]^{val} = [\![\varphi_1]\!]^{val} \cap [\![\varphi_2]\!]^{val}$$

$$[\![\langle a \rangle \varphi]\!]^{val} = \{m \in M \mid \exists_{m \xrightarrow{a} n}\ n \in [\![\varphi]\!]^{val}\} \quad \text{and} \quad [\![[a]\varphi]\!]^{val} = \{m \in M \mid \forall_{m \xrightarrow{a} n}\ n \in [\![\varphi]\!]^{val}\}$$

$$[\![x]\!]^{val} = val(x)$$

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

μ-calculus = modal logic + fixpoints

$$[\![\top]\!]^{\mathsf{val}} = M \quad \text{and} \quad [\![\bot]\!]^{\mathsf{val}} = \emptyset$$

$$[\![\varphi_1 \vee \varphi_2]\!]^{\mathsf{val}} = [\![\varphi_1]\!]^{\mathsf{val}} \cup [\![\varphi_2]\!]^{\mathsf{val}} \quad \text{and} \quad [\![\varphi_1 \wedge \varphi_2]\!]^{\mathsf{val}} = [\![\varphi_1]\!]^{\mathsf{val}} \cap [\![\varphi_2]\!]^{\mathsf{val}}$$

$$[\![\langle \mathsf{a} \rangle \varphi]\!]^{\mathsf{val}} = \{\mathsf{m} \in M \mid \exists_{\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}}\ \mathsf{n} \in [\![\varphi]\!]^{\mathsf{val}}\} \quad \text{and} \quad [\![[\mathsf{a}]\varphi]\!]^{\mathsf{val}} = \{\mathsf{m} \in M \mid \forall_{\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}}\ \mathsf{n} \in [\![\varphi]\!]^{\mathsf{val}}\}$$

$$[\![x]\!]^{\mathsf{val}} = \mathsf{val}(x)$$

$$[\![\mu x.\varphi]\!]^{\mathsf{val}} = \mathsf{LFP}.F \quad \text{and} \quad [\![\nu x.\varphi]\!]^{\mathsf{val}} = \mathsf{GFP}.F$$

$$\varphi ::= \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \mathsf{a} \rangle \varphi \mid [\mathsf{b}]\varphi \mid x \mid \mu x.\varphi \mid \nu x.\varphi$$

μ-calculus = modal logic + fixpoints

$$[\![\top]\!]^{\mathsf{val}} = M \quad \text{and} \quad [\![\bot]\!]^{\mathsf{val}} = \emptyset$$

$$[\![\varphi_1 \vee \varphi_2]\!]^{\mathsf{val}} = [\![\varphi_1]\!]^{\mathsf{val}} \cup [\![\varphi_2]\!]^{\mathsf{val}} \quad \text{and} \quad [\![\varphi_1 \wedge \varphi_2]\!]^{\mathsf{val}} = [\![\varphi_1]\!]^{\mathsf{val}} \cap [\![\varphi_2]\!]^{\mathsf{val}}$$

$$[\![\langle \mathsf{a} \rangle \varphi]\!]^{\mathsf{val}} = \{\mathsf{m} \in M \mid \exists_{\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}}\ \mathsf{n} \in [\![\varphi]\!]^{\mathsf{val}}\} \quad \text{and} \quad [\![[\mathsf{a}]\varphi]\!]^{\mathsf{val}} = \{\mathsf{m} \in M \mid \forall_{\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}}\ \mathsf{n} \in [\![\varphi]\!]^{\mathsf{val}}\}$$

$$[\![x]\!]^{\mathsf{val}} = \mathsf{val}(x)$$

$$[\![\mu x.\varphi]\!]^{\mathsf{val}} = \mathsf{LFP}.F \quad \text{and} \quad [\![\nu x.\varphi]\!]^{\mathsf{val}} = \mathsf{GFP}.F$$

$$\text{with} \quad F(S) = [\![\varphi]\!]^{\mathsf{val}[x::=S]}$$

Why we like it so much?

Why we like it so much?

μ-calculus $\sim$ parity games

Why we like it so much?

μ-calculus $\sim$ parity games

- algorithmicaly feasible & expressive

Why we like it so much?

$$\mu\text{-calculus} \sim \text{parity games}$$

- algorithmicaly feasible & expressive
- equivalent to automata (of various types),

Why we like it so much?

μ-calculus $\sim$ parity games

- algorithmicaly feasible & expressive
- equivalent to automata (of various types), monadic second-order logic MSO, algebras...

Why we like it so much?

μ-calculus $\sim$ parity games

- algorithmicaly feasible & expressive
- equivalent to automata (of various types), monadic second-order logic MSO, algebras...

regular languages

Why we like it so much?

μ-calculus $\sim$ parity games

- algorithmicaly feasible & expressive
- equivalent to automata (of various types), monadic second-order logic MSO, algebras...

regular languages

(of finite/infinite words, trees... or up to bisimulation)

Parity Games:

Parity Games:

$$\underline{V, E, \text{rank} : V \to \mathcal{R}}$$

Parity Games:

$V_\exists \sqcup V_\forall$

$\parallel$

$$\underline{V, E, \mathrm{rank} : V \to \mathcal{R}}$$

Parity Games:

$$\begin{array}{ccc}
V_\exists \sqcup V_\forall & & V \times V \\
\rotatebox{-60}{$=$} & & \rotatebox{90}{$\subseteq$} \\
V, & E, & \mathrm{rank} : V \to \mathcal{R}
\end{array}$$

Parity Games:

$$\begin{array}{ccc} V_\exists \sqcup V_\forall & V \times V & \mathcal{R}_\exists \sqcup \mathcal{R}_\forall \\ \rotatebox{-45}{$\parallel$} & \rotatebox{90}{$\subseteq$} & \rotatebox{45}{$\parallel$} \end{array}$$

$$\underline{V, E, \mathrm{rank} : V \to \mathcal{R}}$$

Parity Games:

$$\begin{array}{ccc} V_\exists \sqcup V_\forall & V \times V & \mathcal{R}_\exists \sqcup \mathcal{R}_\forall \\ \rotatebox{-45}{$\parallel$} & \rotatebox{-90}{$\subseteq$} & \rotatebox{45}{$\parallel$} \end{array}$$

$$V, E, \mathrm{rank} : V \to \mathcal{R}$$

- $\exists$ve and $\forall$dam move between positions, round by round

Parity Games:

$$
\begin{array}{ccccc}
V_\exists \sqcup V_\forall & & V \times V & & \mathcal{R}_\exists \sqcup \mathcal{R}_\forall \\
\| & & \cup\!| & & /\!/ \\
\multicolumn{5}{c}{V, E, \mathrm{rank} : V \to \mathcal{R}}
\end{array}
$$

- $\exists$ve and $\forall$dam move between positions, round by round
- from position v its owner chooses vEw & the game moves to w

Parity Games:

$$\begin{array}{cccc} V_\exists \sqcup V_\forall & V \times V & & \mathcal{R}_\exists \sqcup \mathcal{R}_\forall \\ \| & \cup\!\!\!\mid & & \| \\ V, & E, & \mathrm{rank} : V \to & \mathcal{R} \end{array}$$

- $\exists$ve and $\forall$dam move between positions, round by round
- from position v its owner chooses vEw & the game moves to w
- if a player is stuck (has no legal move) looses immediately

Parity Games:

$$\begin{array}{ccc} V_\exists \sqcup V_\forall & V \times V & \mathcal{R}_\exists \sqcup \mathcal{R}_\forall \\ \rotatebox{-45}{$\parallel$} & \rotatebox{90}{$\subseteq$} & \rotatebox{45}{$\parallel$} \\ \end{array}$$

$$\underline{V, E, \mathrm{rank} : V \to \mathcal{R}}$$

- $\exists$ve and $\forall$dam move between positions, round by round
- from position v its owner chooses vEw & the game moves to w
- if a player is stuck (has no legal move) looses immediately
- otherwise an infinite play π: look at the greatest rank r appearing infinitely often in π — the owner of r looses

Parity Games:

$$
\begin{array}{ccc}
V_\exists \sqcup V_\forall & V \times V & \mathcal{R}_\exists \sqcup \mathcal{R}_\forall \\
\| & \cup\!| & \| \\
\end{array}
$$

$$\underline{V, E, \mathrm{rank} : V \to \mathcal{R}}$$

- $\exists$ve and $\forall$dam move between positions, round by round
- from position v its owner chooses vEw & the game moves to w
- if a player is stuck (has no legal move) looses immediately
- otherwise an infinite play π: look at the greatest rank r appearing infinitely often in π — the owner of r looses

(strategies, winning strategies, etc. defined as usual)

Game Semantics:

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathsf{SubFor}(\varphi)}$

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathsf{SubFor}(\varphi)}$

$$\exists\text{ve wins from } (\mathrm{m}, \varphi) \iff \mathrm{m} \in [\![\varphi]\!]$$

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathsf{SubFor}(\varphi)}$

$$\exists\mathsf{ve\ wins\ from\ }(\mathsf{m},\varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

- possible moves E depend on the topmost connective:

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathrm{SubFor}(\varphi)}$

$$\exists\text{ve wins from } (\mathrm{m}, \varphi) \iff \mathrm{m} \in [\![\varphi]\!]$$

- possible moves E depend on the topmost connective:
    - in $(\mathrm{m}, \psi \vee \psi')$ $\exists$ve chooses (m, ψ) or (m, ψ'),

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathrm{SubFor}(\varphi)}$

$$\exists\text{ve wins from } (\mathrm{m}, \varphi) \iff \mathrm{m} \in [\![\varphi]\!]$$

- possible moves E depend on the topmost connective:
    - in $(\mathrm{m}, \psi \vee \psi')$ $\exists$ve chooses (m, ψ) or (m, ψ'),
    - in $(\mathrm{m}, \langle \mathrm{a} \rangle \psi)$ $\exists$ve chooses (n, ψ) with $\mathrm{m} \xrightarrow{\mathrm{a}} \mathrm{n}$,

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathrm{SubFor}(\varphi)}$

$$\exists\text{ve wins from } (\mathrm{m}, \varphi) \iff \mathrm{m} \in [\![\varphi]\!]$$

- possible moves E depend on the topmost connective:
  - in $(\mathrm{m}, \psi \vee \psi')$ $\exists$ve chooses (m, ψ) or (m, ψ'),
  - in $(\mathrm{m}, \langle \mathrm{a} \rangle \psi)$ $\exists$ve chooses (n, ψ) with $\mathrm{m} \xrightarrow{\mathrm{a}} \mathrm{n}$,
  - with $\wedge$ and $[\mathrm{a}]$ in place of $\vee$ and $\langle \mathrm{a} \rangle$: same but $\forall$dam chooses

Game Semantics:

- given $\mathcal{M}$ and φ, positions $\underline{V = M \times \mathsf{SubFor}(\varphi)}$

$$\exists\text{ve wins from } (\mathsf{m}, \varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

- possible moves E depend on the topmost connective:
  - in $(\mathsf{m}, \psi \vee \psi')$ $\exists$ve chooses (m, ψ) or (m, ψ'),
  - in $(\mathsf{m}, \langle \mathsf{a} \rangle \psi)$ $\exists$ve chooses (n, ψ) with $\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}$,
  - with $\wedge$ and $[\mathsf{a}]$ in place of $\vee$ and $\langle \mathsf{a} \rangle$: same but $\forall$dam chooses
  - from $(\mathsf{m}, \mu \mathsf{x}.\psi)$ and $(\mathsf{m}, \nu \mathsf{x}.\psi)$ to (m, ψ)

Game Semantics:

- given $\mathcal{M}$ and φ, positions $V = M \times \mathsf{SubFor}(\varphi)$

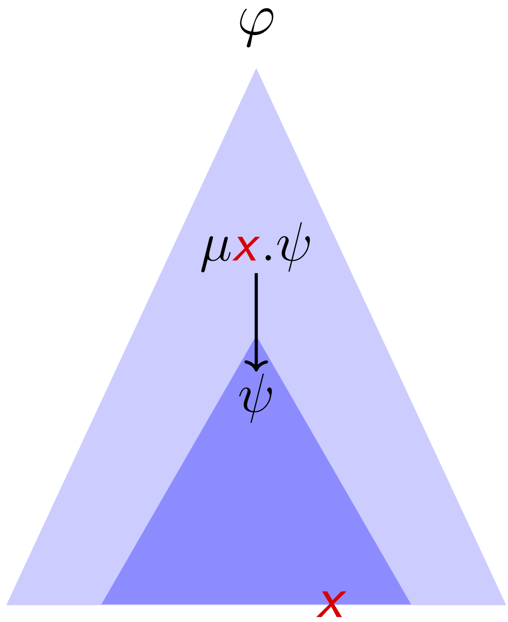
$$\exists\text{ve wins from } (\mathsf{m}, \varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

- possible moves E depend on the topmost connective:
  - in $(\mathsf{m}, \psi \vee \psi')$ ∃ve chooses (m, ψ) or (m, ψ'),
  - in $(\mathsf{m}, \langle \mathsf{a} \rangle \psi)$ ∃ve chooses (n, ψ) with $\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}$,
  - with $\wedge$ and $[\mathsf{a}]$ in place of $\vee$ and $\langle \mathsf{a} \rangle$: same but ∀dam chooses
  - from $(\mathsf{m}, \mu x.\psi)$ and $(\mathsf{m}, \nu x.\psi)$ to (m, ψ)
  - plus x unfolds!

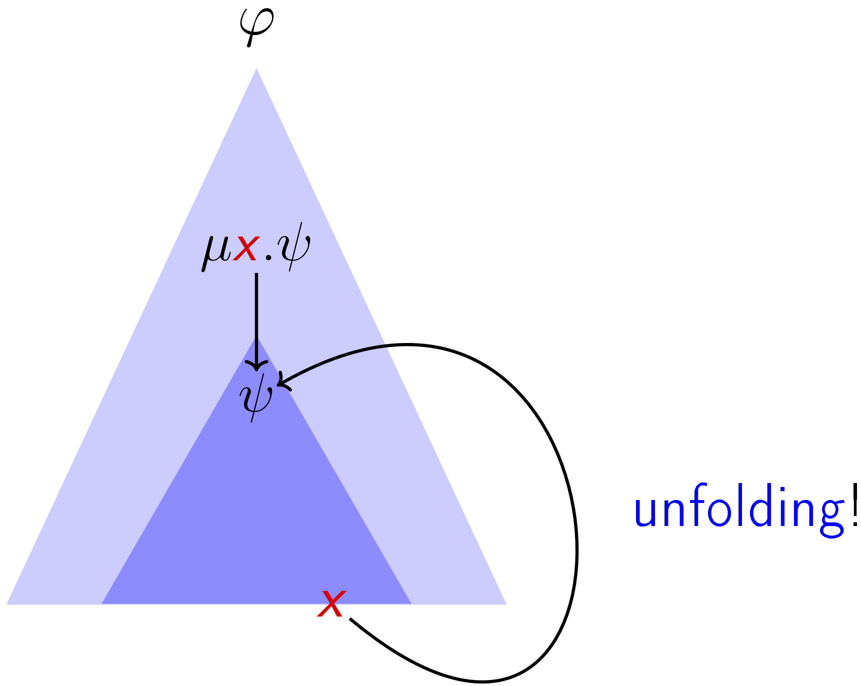
- operators $\mu x.$ and $\nu x.$ <u>bind</u> variable x

- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)

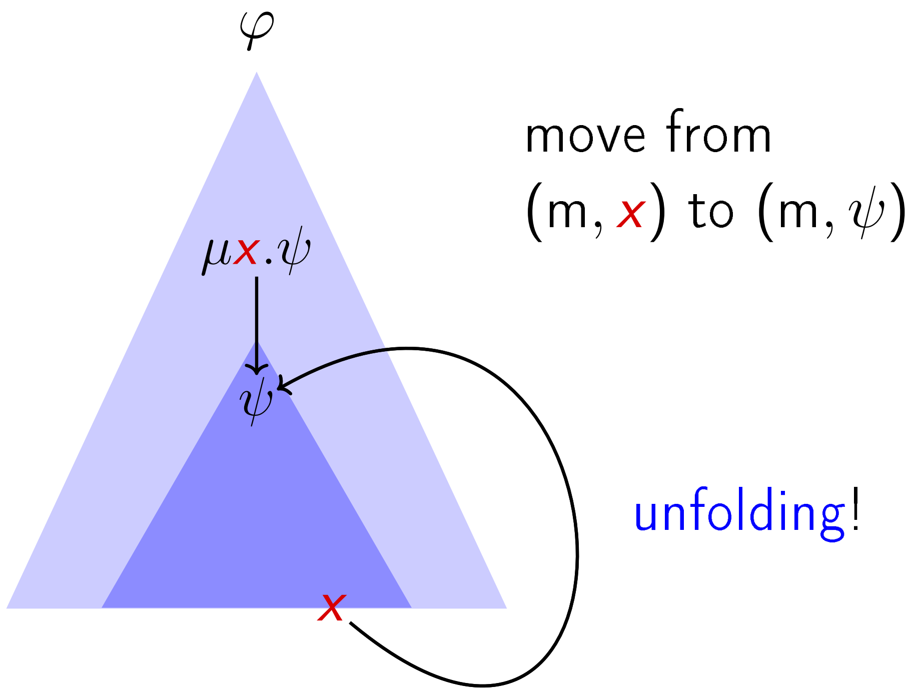
- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)



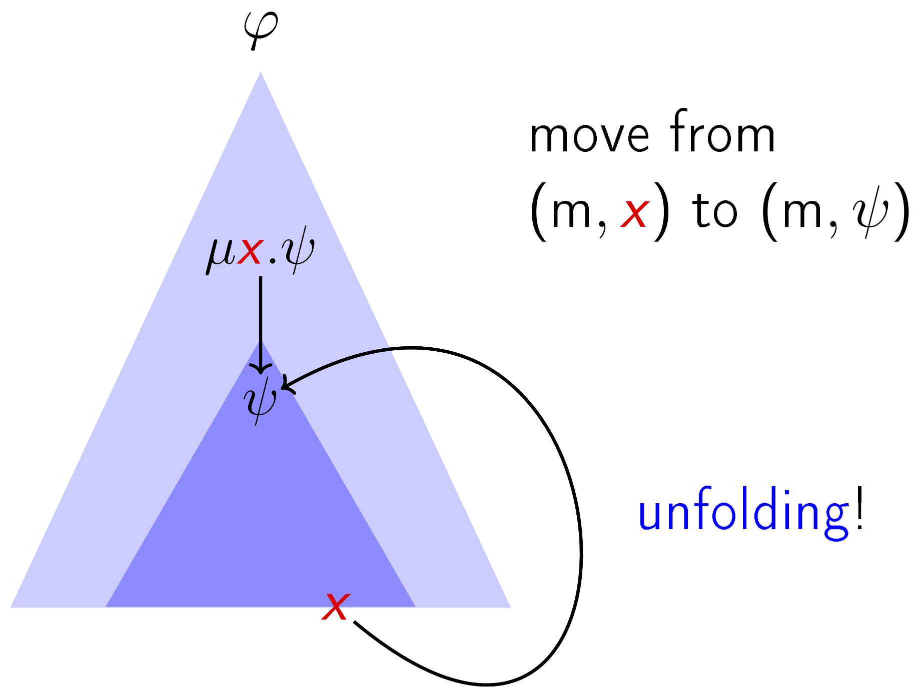
- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)



- operators $\mu x.$ and $\nu x.$ <u>bind</u> variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)

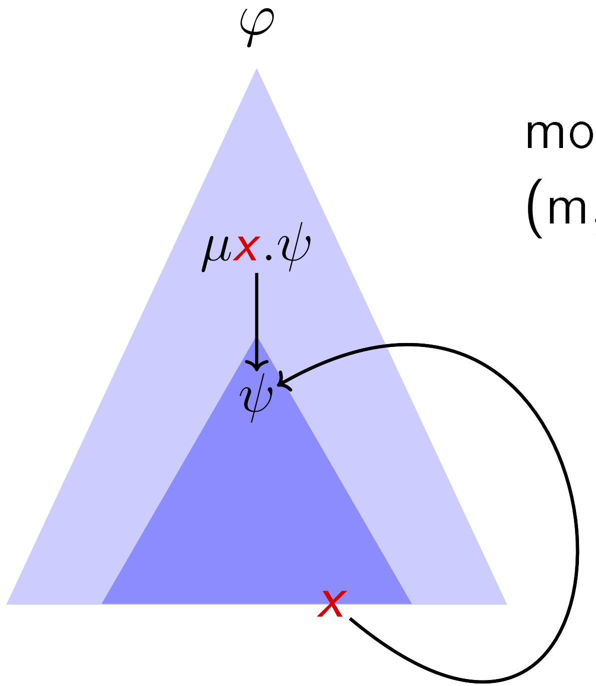


- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)



- unfolding may lead to infinite plays:

- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)

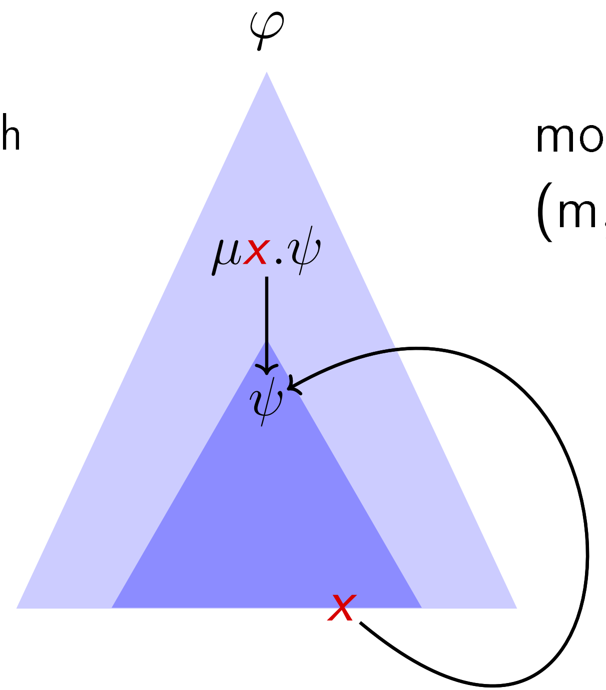


move from (m, x) to (m, ψ)

unfolding!

- unfolding may lead to infinite plays:
- $\exists$ve looses if the outermost operator unfolded infinitely often is μ

- operators $\mu x.$ and $\nu x.$ <u>bind</u> variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)



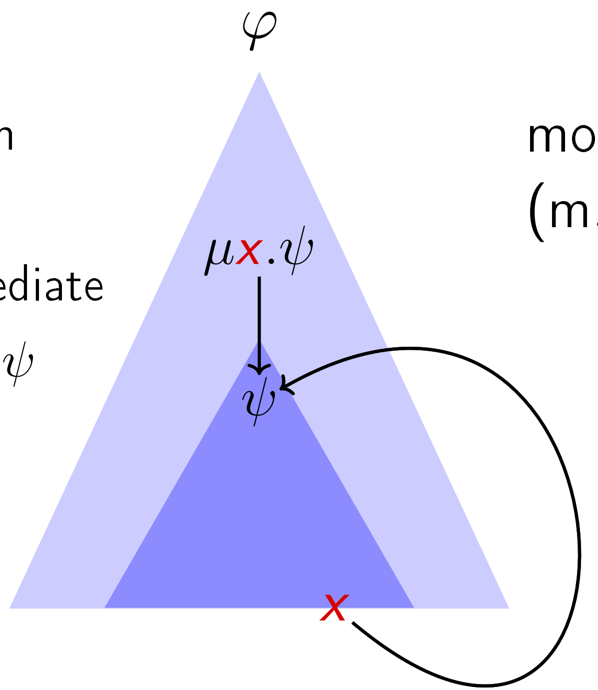
move from (m, x) to (m, ψ)

unfolding!

- rank compatible with subformula order

- unfolding may lead to infinite plays:
- $\exists$ve looses if the outermost operator unfolded infinitely often is μ

- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)



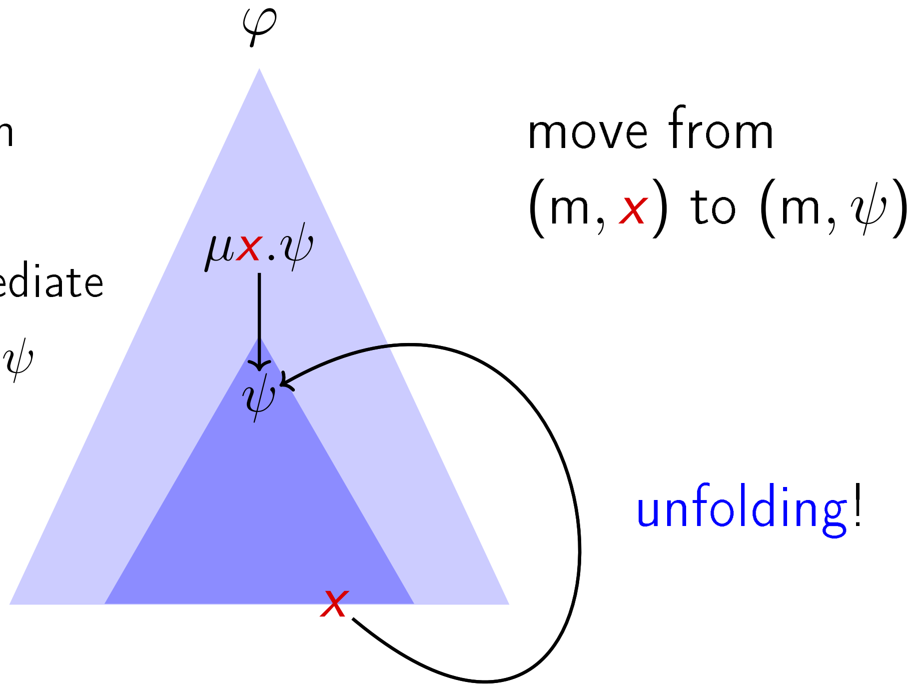
move from (m, x) to (m, ψ)

unfolding!

- rank compatible with subformula order
- rank(m, ψ) for immediate subformula ψ of $\mu x.\psi$ belongs to ∃ve

- unfolding may lead to infinite plays:
- ∃ve looses if the outermost operator unfolded infinitely often is μ

- operators $\mu x.$ and $\nu x.$ bind variable x
- from (m, x) with x bound in $\mu x.\psi$ or $\nu x.\psi$ the game moves to (m, ψ)



- rank compatible with subformula order
- $\mathrm{rank}(\mathrm{m}, \psi)$ for immediate subformula ψ of $\mu x.\psi$ belongs to ∃ve
- symmetrically with ν and ∀dam

- unfolding may lead to infinite plays:
- ∃ve looses if the outermost operator unfolded infinitely often is μ

Example: $\nu x.\Diamond x$

Example: $\nu x.\Diamond x$

$$\begin{tikzcd}
\nu x.\Diamond x \arrow[d] \\
\Diamond x \arrow[d] \\
x \arrow[u, bend right]
\end{tikzcd}$$

Example: $\nu x.\Diamond x$

$$\begin{tikzcd}
\nu x.\Diamond x \arrow[d] \\
\Diamond x \arrow[d] \\
x \arrow[u, bend right=90]
\end{tikzcd}$$

∃ve chooses successor
in the model

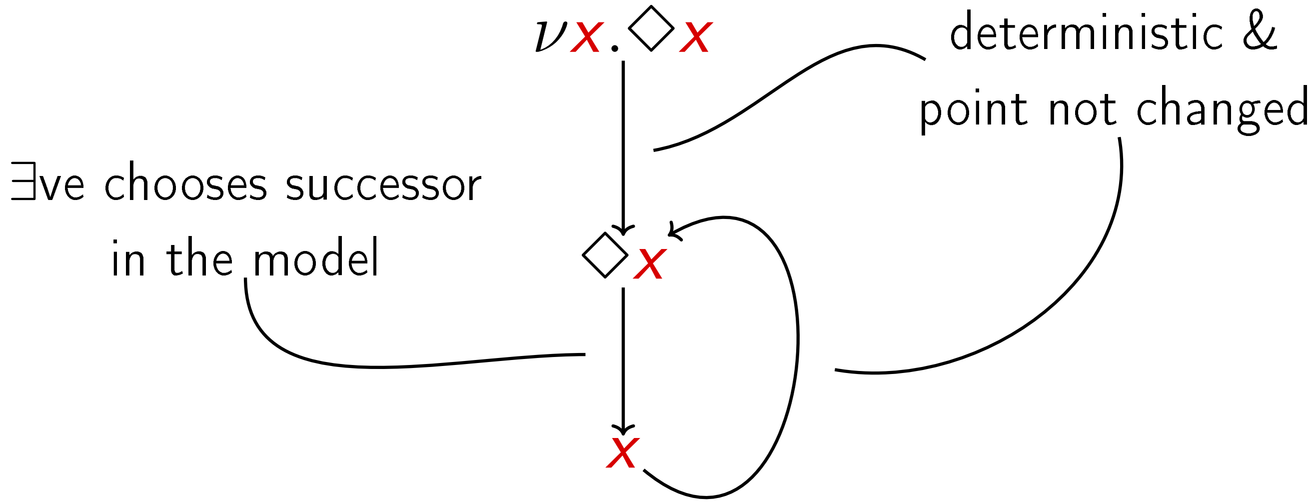
Example: $\nu x.\Diamond x$

deterministic & point not changed

$$\begin{tikzcd} \nu x.\Diamond x \arrow[d] \\ \Diamond x \arrow[d] \\ x \arrow[u, bend right] \end{tikzcd}$$

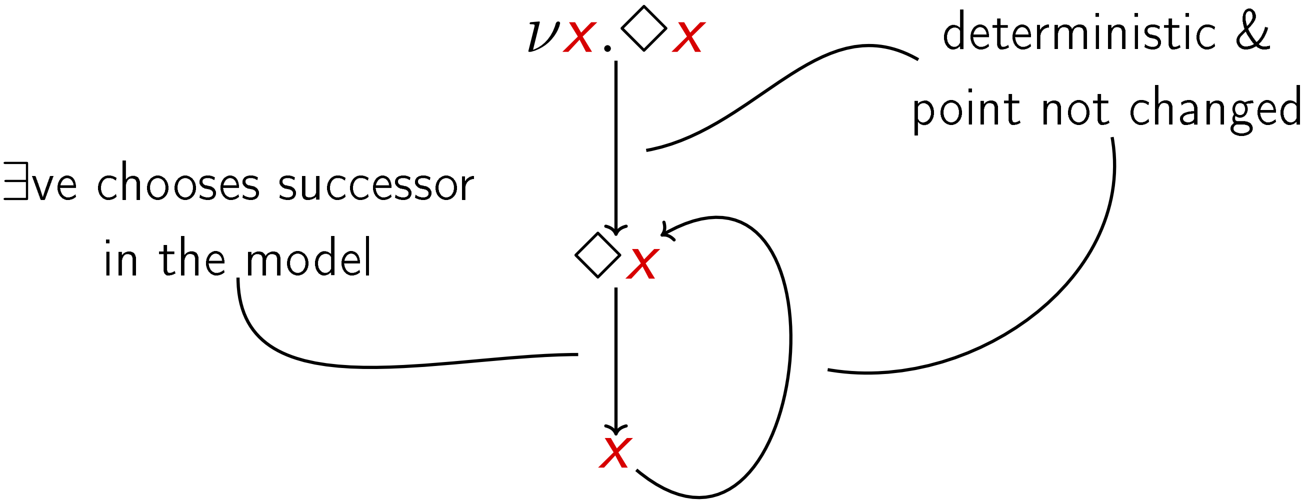
$\exists$ve chooses successor in the model

Example: $\nu x.\Diamond x$



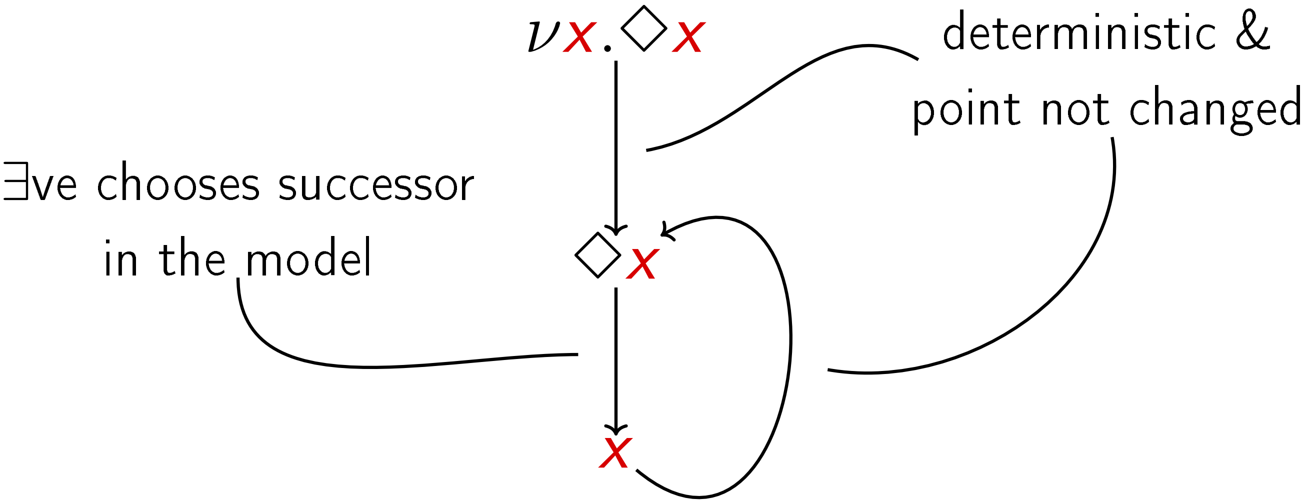
- ∃ve wins all infinie plays

Example: $\nu x.\Diamond x$



- ∃ve wins all infinie plays
- initially a deterministic move from $(\mathrm{m}, \nu x.\Diamond x)$ to $(\mathrm{m}, \Diamond x)$; then

Example: $\nu x.\Diamond x$



- $\exists$ve wins all infinie plays
- initially a deterministic move from $(\mathrm{m}, \nu x.\Diamond x)$ to $(\mathrm{m}, \Diamond x)$; then
- $(\mathrm{m}, \Diamond x)E(\mathrm{m}', x)E(\mathrm{m}', \Diamond x)E(\mathrm{m}'', x)\ldots$, every second deterministic and $\mathrm{m} \to \mathrm{m}' \to \mathrm{m}''\ldots$ chosen by $\exists$ve

As desired:

As desired:

$$\exists\text{ve wins from } (\mathsf{m}, \varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

As desired:

$$\exists\text{ve wins from } (\mathsf{m}, \varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

works for every $\mathcal{M}$ and φ!

Parity Automata:

Parity Automata:

$$Q, q_I, \mathrm{rank} : Q \to \mathcal{R}$$

Parity Automata:

$Q_\exists \sqcup Q_\forall$

$\mathbin{/\!/}$

$Q, q_I, \mathrm{rank} : Q \to \mathcal{R}$

Parity Automata:

$$\begin{array}{ccc} Q_\exists \sqcup Q_\forall & & Q \\ \rotatebox{-45}{$=$} & & \rotatebox{90}{$\in$} \\ Q, & q_I, & \mathrm{rank} : Q \to \mathcal{R} \end{array}$$

Parity Automata:

$$\overset{Q_\exists \sqcup Q_\forall}{\overset{\shortparallel}{Q}},\ \overset{Q}{\overset{\cup}{q_I}},\ \mathrm{rank} : Q \to \mathcal{R}$$

$$\delta : Q \to \mathcal{P}(Q) \cup (\mathrm{Act} \times Q)$$

and a transition function

Parity Automata:

$Q_\exists \sqcup Q_\forall \qquad Q$

$\rotatebox{-45}{$=$} \qquad \rotatebox{90}{$\in$}$

$$Q, q_I, \mathrm{rank} : Q \to \mathcal{R}$$

$$\delta : Q \to \overbrace{\mathcal{P}(Q)}^{\epsilon\text{-transitions}} \cup (\mathrm{Act} \times Q)$$

and a transition function

Parity Automata:

$$Q_\exists \sqcup Q_\forall \qquad Q$$

$$\| \qquad \cup$$

$$\underline{Q, q_I, \mathrm{rank} : Q \to \mathcal{R}}$$

$$\delta : Q \to \overbrace{\mathcal{P}(Q)}^{\epsilon\text{-transitions}} \cup \overbrace{(\mathrm{Act} \times Q)}^{\text{modal transitions}}$$

and a transition function

Parity Automata:

$$Q_\exists \sqcup Q_\forall \qquad Q$$

$$\rotatebox{-45}{$=$} \qquad \rotatebox{90}{$\in$}$$

$$\underline{Q, q_I, \mathrm{rank} : Q \to \mathcal{R}}$$

the semantics of the automton $\mathcal{A}$ defined by a game

$$\delta : Q \to \overbrace{\mathcal{P}(Q)}^{\epsilon\text{-transitions}} \cup \overbrace{(\mathrm{Act} \times Q)}^{\text{modal transitions}}$$

and a transition function

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:
  - (m, p) with $p \in \delta(q)$ if $\delta(q) \subseteq Q$,

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:
  - (m, p) with $p \in \delta(q)$ if $\delta(q) \subseteq Q$,
  - (n, p) with $\mathsf{m} \xrightarrow{\mathsf{a}} \mathsf{n}$ if $\delta(q) = (\mathsf{a}, p)$.

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:
  - (m, p) with $p \in \delta(q)$ if $\delta(q) \subseteq Q$,
  - (n, p) with $\mathrm{m} \xrightarrow{\mathrm{a}} \mathrm{n}$ if $\delta(q) = (\mathrm{a}, p)$.
- ownership and ranks inherited from Q

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:
  - (m, p) with $p \in \delta(q)$ if $\delta(q) \subseteq Q$,
  - (n, p) with $\mathrm{m} \xrightarrow{\mathrm{a}} \mathrm{n}$ if $\delta(q) = (\mathrm{a}, p)$.
- ownership and ranks inherited from Q
  - $V_\exists = M \times Q_\exists$, $V_\forall = M \times Q_\forall$

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:
  - (m, p) with $p \in \delta(q)$ if $\delta(q) \subseteq Q$,
  - (n, p) with $\mathrm{m} \xrightarrow{\mathrm{a}} \mathrm{n}$ if $\delta(q) = (\mathrm{a}, p)$.
- ownership and ranks inherited from Q
  - $V_\exists = M \times Q_\exists$, $V_\forall = M \times Q_\forall$
  - $\mathrm{rank}(\mathrm{m}, q) = \mathrm{rank}(q)$

Semantic Game for automaton $\mathcal{A}$ and model $\mathcal{M}$:

- positions $V = M \times Q$
- from (m, q) moves to:
  - (m, p) with $p \in \delta(q)$ if $\delta(q) \subseteq Q$,
  - (n, p) with $\mathrm{m} \xrightarrow{\mathrm{a}} \mathrm{n}$ if $\delta(q) = (\mathrm{a}, p)$.
- ownership and ranks inherited from Q
  - $V_\exists = M \times Q_\exists$, $V_\forall = M \times Q_\forall$
  - $\mathrm{rank}(\mathrm{m}, q) = \mathrm{rank}(q)$

language of $\mathcal{A}$:

$\mathcal{A}$ accepts $\mathrm{m} \in M$ $=$ $\exists$ve wins the game from (m, q_I)

$\underline{\mu\text{-ML formulae}}$ $\underline{\text{parity automata}}$

$$
\begin{tikzcd}
\underline{\mu\text{-ML formulae}} \arrow[rr, bend left, "\underline{\text{game semantics}}"] & & \underline{\text{parity automata}}
\end{tikzcd}
$$

$Q = \mathsf{SubFor}(\varphi)$

$$\begin{tikzcd} \underline{\mu\text{-ML formulae}} \arrow[rr, bend left, "\underline{\text{game semantics}}"] & & \underline{\text{parity automata}} \end{tikzcd}$$

game semantics

$Q = \mathsf{SubFor}(\varphi)$

μ-ML formulae

parity automata

μ-ML describes arbitrary automata

$Q = \mathsf{SubFor}(\varphi)$

$$\begin{tikzcd}[column sep=huge]
\mu\text{-ML formulae} \arrow[rr, bend left=60, "\text{game semantics}"] & = & \text{parity automata} \arrow[ll, bend left=60, "\mu\text{-ML describes arbitrary automata}"]
\end{tikzcd}$$

Limitations:

Limitations:

- μ-ML has the finite model property: if a formula φ is true in a point m of some model $\mathcal{M}$, then it is true in some point n of a finite model $\mathcal{N}$

Limitations:

- μ-ML has the finite model property: if a formula φ is true in a point m of some model $\mathcal{M}$, then it is true in some point n of a finite model $\mathcal{N}$
- in general this is a good thing, but limits expressive power

Limitations:

- μ-ML has the finite model property: if a formula φ is true in a point m of some model $\mathcal{M}$, then it is true in some point n of a finite model $\mathcal{N}$
- in general this is a good thing, but limits expressive power
- for instance, (un)boundedness properties such as:

"there exist arbitrarily long paths originating in a given point"

cannot be defined

Limitations:

- μ-ML has the finite model property: if a formula φ is true in a point m of some model $\mathcal{M}$, then it is true in some point n of a finite model $\mathcal{N}$
- in general this is a good thing, but limits expressive power
- for instance, (un)boundedness properties such as:

  "there exist arbitrarily long paths originating in a given point"

  cannot be defined
- well-foundedness definable with φ_{WF}, so if there was φ_{U} defining the above prope
  then $\varphi_{\mathsf{WF}} \wedge \varphi_{\mathsf{U}}$ would be satisfiable but not in a finite model (König's Lemma)

Limitations:

- μ-ML has the finite model property: if a formula φ is true in a point m of some model $\mathcal{M}$, then it is true in some point n of a finite model $\mathcal{N}$
- in general this is a good thing, but limits expressive power
- for instance, (un)boundedness properties such as:

"there exist arbitrarily long paths originating in a given point"

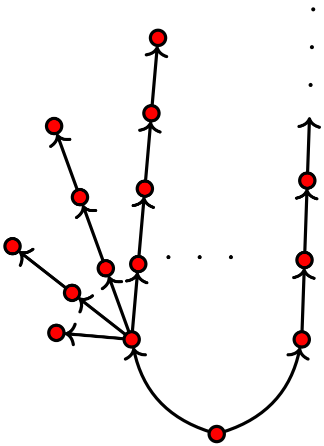
cannot be defined

- well-foundedness definable with φ_{WF}, so if there was φ_{U} defining the above prope
  then $\varphi_{\mathsf{WF}} \wedge \varphi_{\mathsf{U}}$ would be satisfiable but not in a finite model (König's Lemma)

we want to extend μ-ML!!!

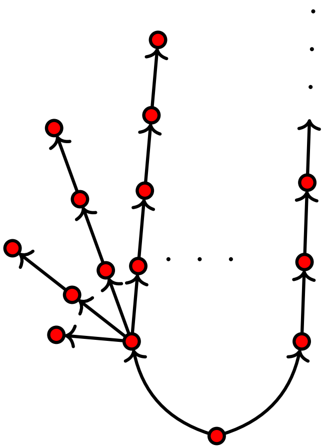
- we compute fixpoints by (transfinite) iteration of F:

- we compute fixpoints by (transfinite) iteration of F:



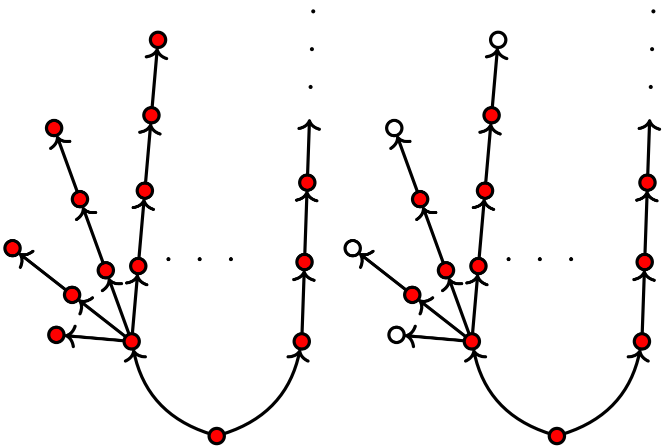
$F_\nu^0 = M$

- we compute fixpoints by (transfinite) iteration of F:



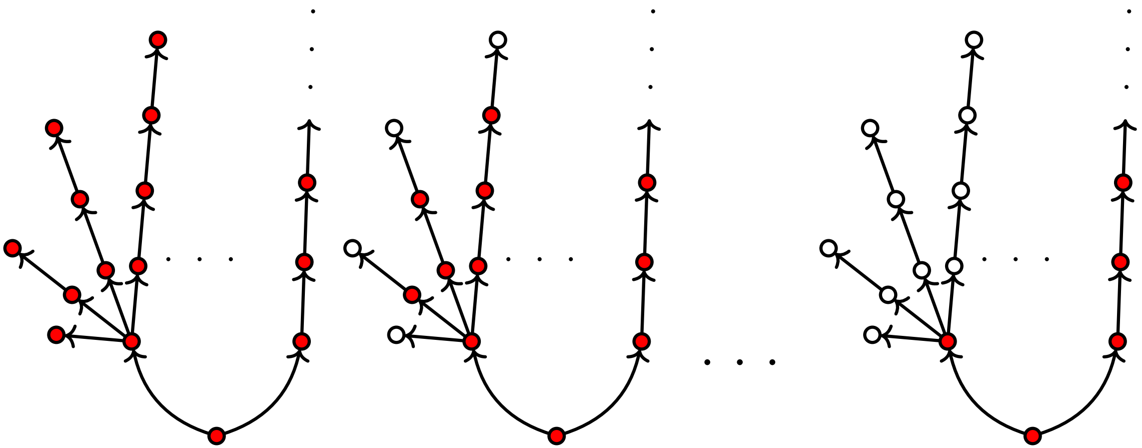
$F^0_\nu = M$ $\qquad$ $F^1_\nu = F(F^0_\nu)$

- we compute fixpoints by (transfinite) iteration of F:



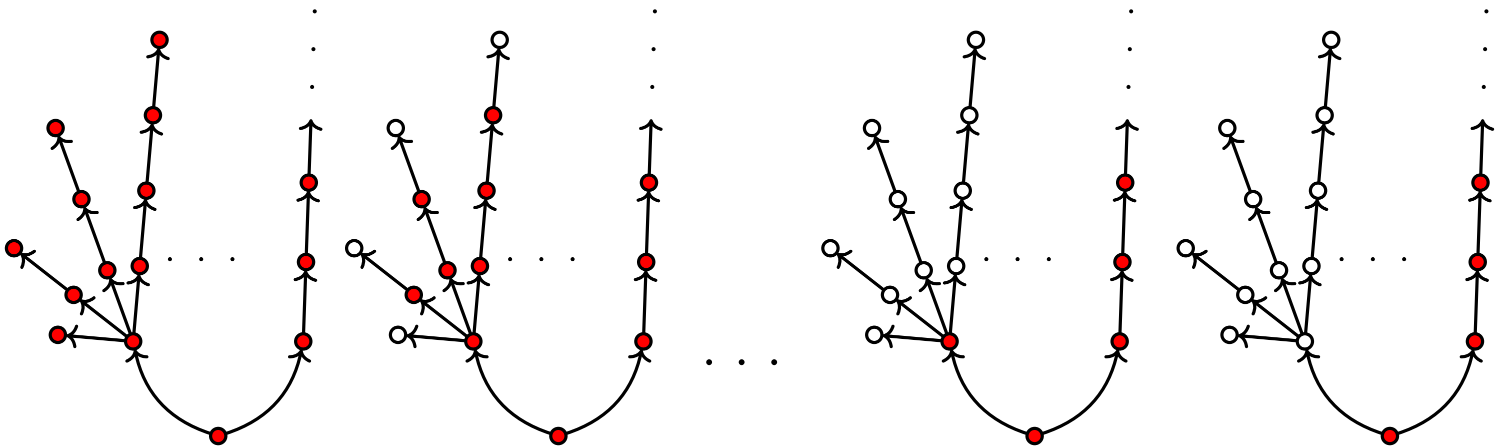
$F^0_\nu = M$ $\quad$ $F^1_\nu = F(F^0_\nu) =$"$\exists$ a child"

- we compute fixpoints by (transfinite) iteration of F:



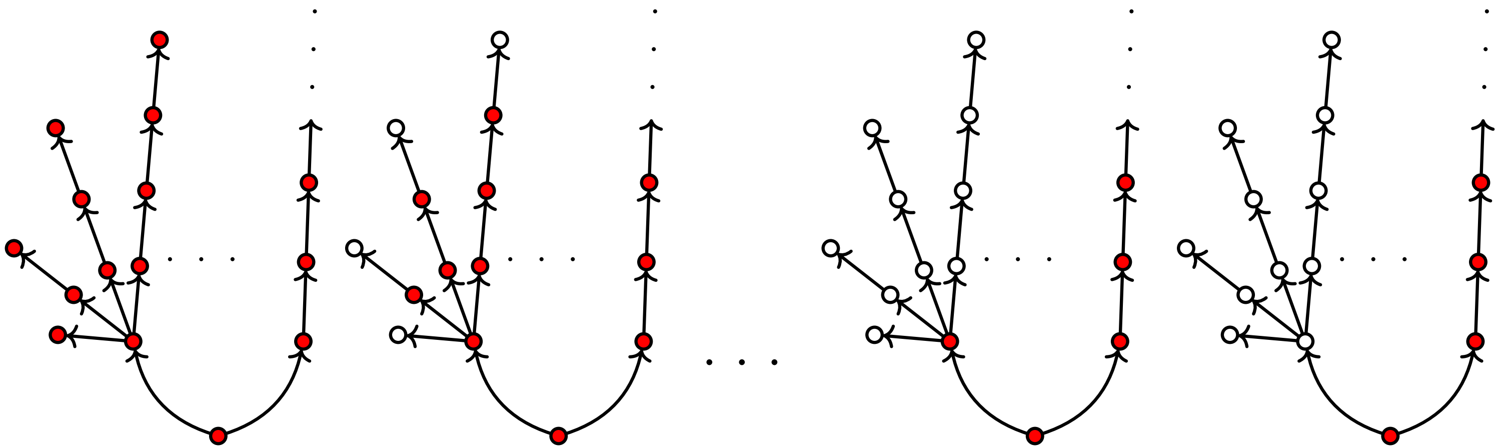
$F_\nu^0 = M$ $\quad$ $F_\nu^1 = F(F_\nu^0) =$ "$\exists$ a child" $\quad$ $F_\nu^\omega =$ "arbitrarily long paths"

- we compute fixpoints by (transfinite) iteration of F:



$F_\nu^0 = M$ $\quad$ $F_\nu^1 = F(F_\nu^0) =$ "$\exists$ a child" $\quad$ $F_\nu^\omega =$ "arbitrarily long paths" $\quad$ $F_\nu^{\omega+1} = F_\nu^{\omega+2} = \mathrm{GFP}.F$

- we compute fixpoints by (transfinite) iteration of F:



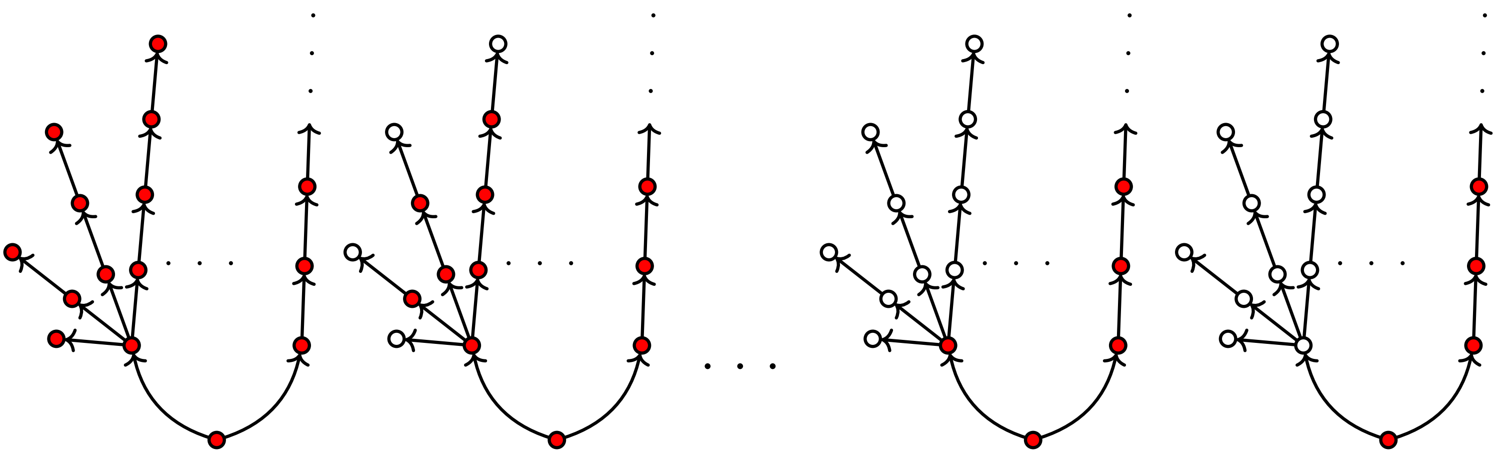
$F^0_\nu = M$ $\quad$ $F^1_\nu = F(F^0_\nu) =$ "$\exists$ a child" $\quad$ $F^\omega_\nu =$ "arbitrarily long paths" $\quad$ $F^{\omega+1}_\nu = F^{\omega+2}_\nu = \mathrm{GFP}.F$

$$\| $$

$$[\![\nu x.\Diamond x]\!]$$

- we compute fixpoints by (transfinite) iteration of F:

add countdown operator $\nu^{\omega} x.\Diamond x$ to the syntax!



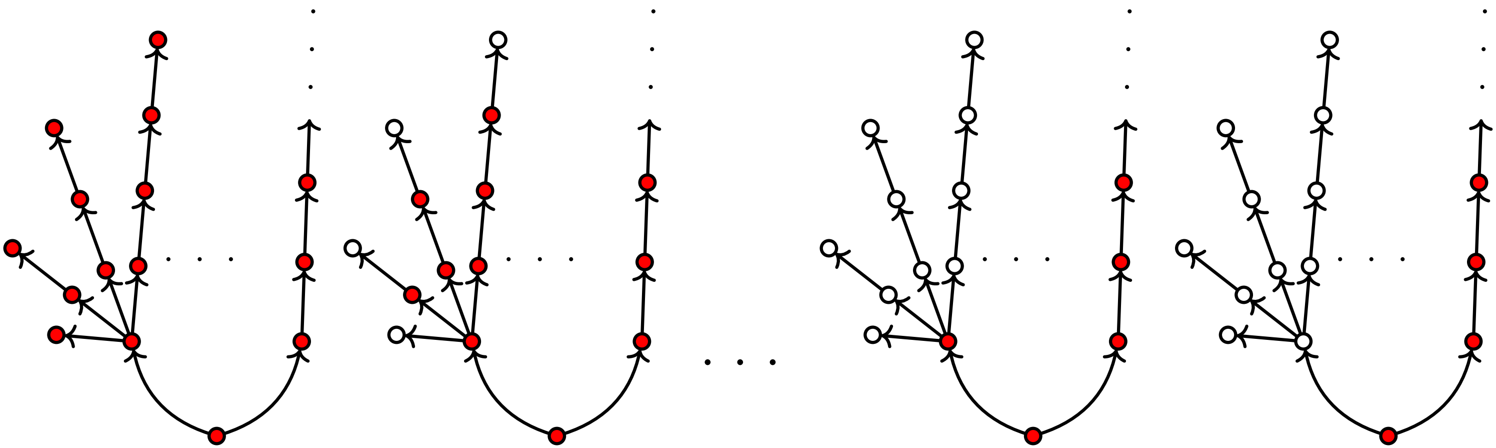
$F_\nu^0 = M$ $\quad$ $F_\nu^1 = F(F_\nu^0) =$ "$\exists$ a child" $\quad$ $F_\nu^\omega =$ "arbitrarily long paths" $\quad$ $F_\nu^{\omega+1} = F_\nu^{\omega+2} = \mathrm{GFP}.F$

$$\parallel$$

$$[\![\nu x.\Diamond x]\!]$$

- we compute fixpoints by (transfinite) iteration of F:

add countdown operator $\nu^{\omega} x.\Diamond x$ to the syntax!



$F^0_\nu = M$ $\quad$ $F^1_\nu = F(F^0_\nu) =$ "$\exists$ a child" $\quad$ . . . $\quad$ $F^\omega_\nu =$ "arbitrarily long paths" $\quad$ $F^{\omega+1}_\nu = F^{\omega+2}_\nu = \mathrm{GFP}.F$

$\parallel$ $\quad$ $\parallel$

$[\![\nu^{\omega} x.\Diamond x]\!]$ $\quad$ $[\![\nu x.\Diamond x]\!]$

μ-calculus + countdown operators μ^ω, ν^ω

$$\frac{\mu\text{-calculus} + \text{countdown operators } \mu^{\omega}, \nu^{\omega}}{=}$$

$$\underline{\text{countdown } \mu\text{-calculus}}$$

Syntax:

extended with μ^ω and ν^ω

μ-calculus + countdown operators μ^ω, ν^ω

=

countdown μ-calculus

Syntax:

extended with μ^ω and ν^ω

$$\frac{\mu\text{-calculus} + \text{countdown operators } \mu^\omega, \nu^\omega}{=}$$

$$\underline{\text{countdown } \mu\text{-calculus}}$$

$$[\![\mu^\omega x.\varphi]\!]^{\mathsf{val}} = F_\mu^\omega \quad \text{and} \quad [\![\nu^\omega x.\varphi]\!]^{\mathsf{val}} = F_\nu^\omega$$

Semantics:

μ-calculus $\sim$ parity games

equivalent to alternating parity automata

μ-calculus $\sim$ parity games

equivalent to alternating parity automata

μ-calculus $\sim$ parity games

wining regions definable in logic

equivalent to alternating parity automata

countdown μ-calculus $\sim$ parity games

wining regions definable in logic

equivalent to alternating ~~parity~~ automata

countdown!

countdown μ-calculus $\sim$ ~~parity~~ games

countdown!

wining regions definable in logic

Game for $\nu x. \Diamond x$:

Game for $\nu x.\Diamond x$:

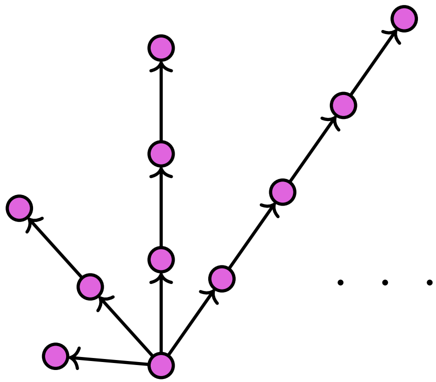
- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point

Game for $\nu x. \Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \dots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

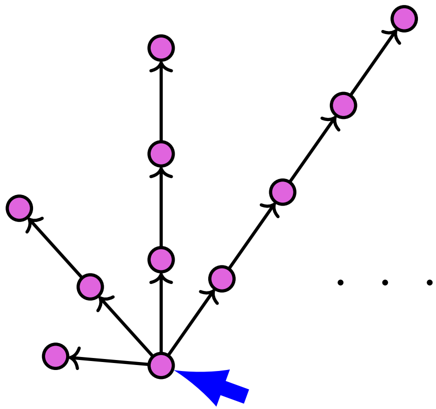
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



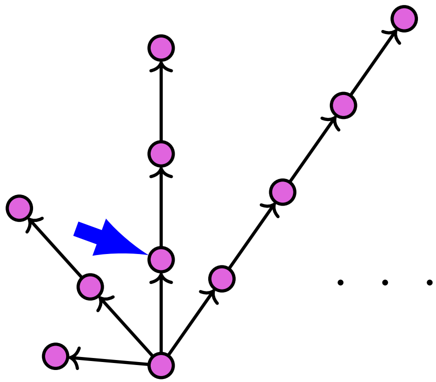
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



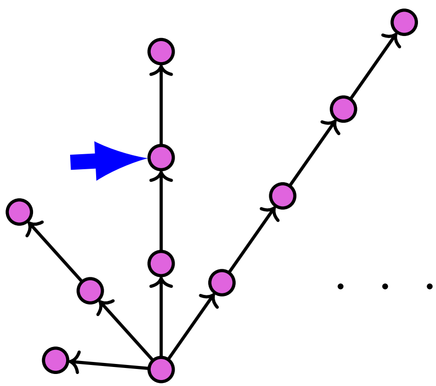
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



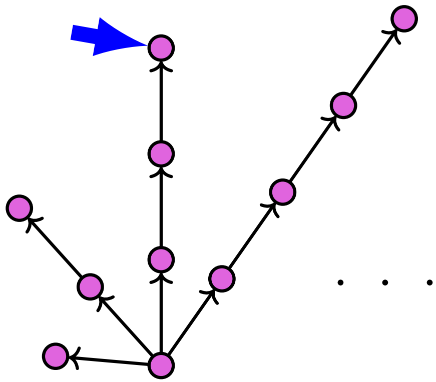
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



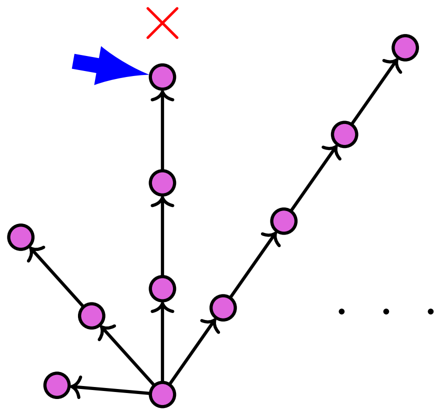
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



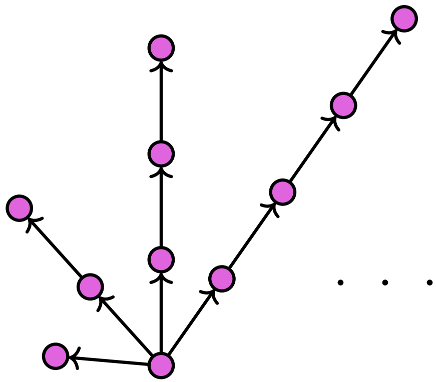
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \dots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

$\exists$ infinite path



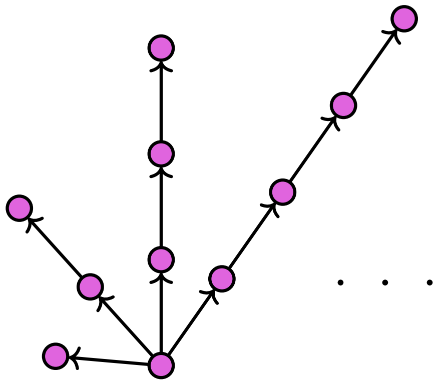
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$$\iff$$

$\exists$ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

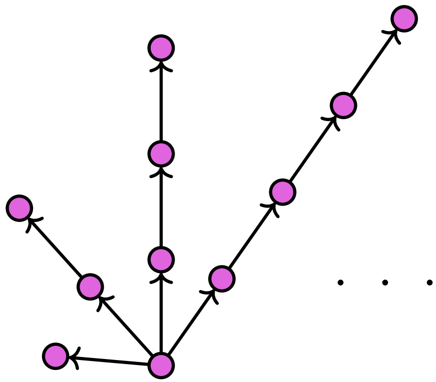
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\iff$

$\exists$ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω

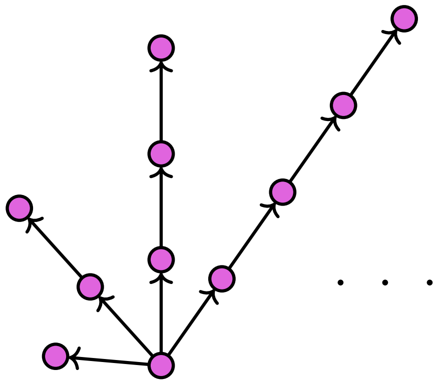
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \dots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

$\exists$ infinite path



Game for $\nu^\omega x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

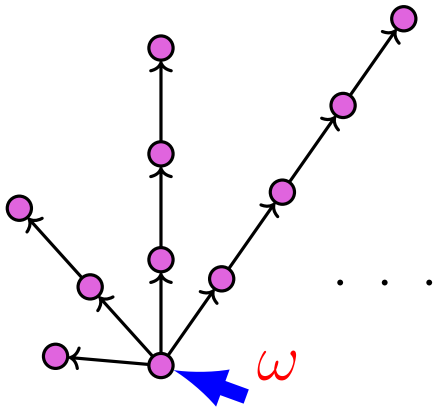
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\iff$

$\exists$ infinite path



Game for $\nu^\omega x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

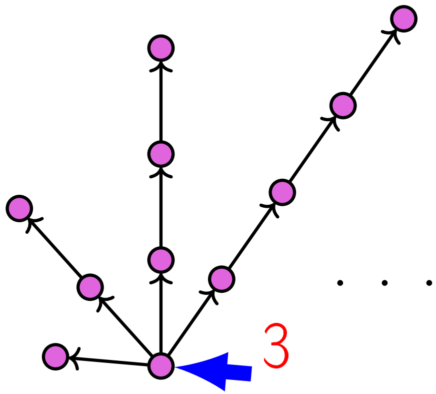
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to ∀dam (meaning: he looses all infinite plays)

∃ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

∃ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) ∀dam decrements the counter (picks smaller value); (ii) ∃ve picks an edge $m \to m'$ to a new point m'.

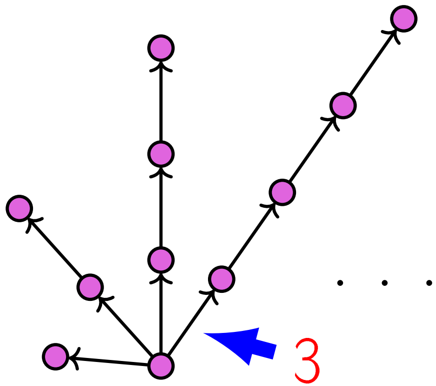
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

$\exists$ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

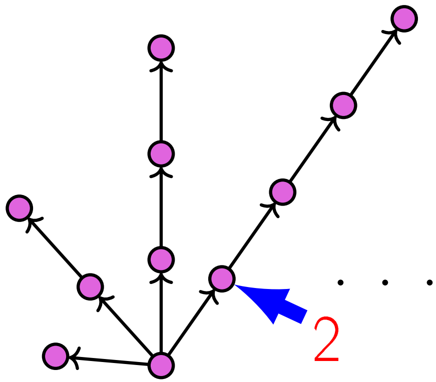
Game for $\nu x.\Diamond x$:

- ▶ ∃ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- ▶ one rank r, belonging to ∀dam (meaning: he looses all infinite plays)

∃ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\iff$

∃ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- ▶ same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- ▶ In each round: (i) ∀dam decrements the counter (picks smaller value); (ii) ∃ve picks an edge $m \to m'$ to a new point m'.

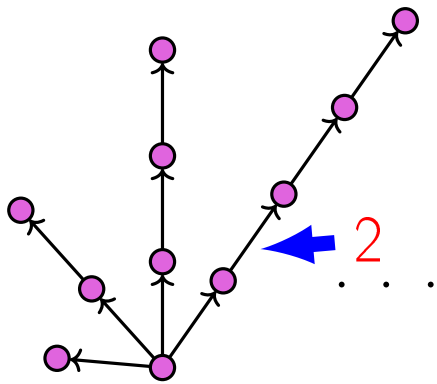
Game for $\nu x. \Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x. \Diamond x)$

$\iff$

$\exists$ infinite path



Game for $\nu^{\omega} x. \Diamond x$:

- same as for $\nu x. \Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

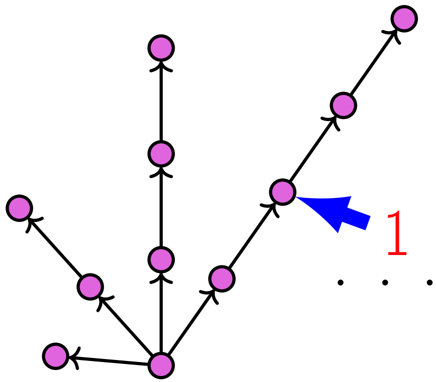
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \dots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

$\exists$ infinite path



Game for $\nu^\omega x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

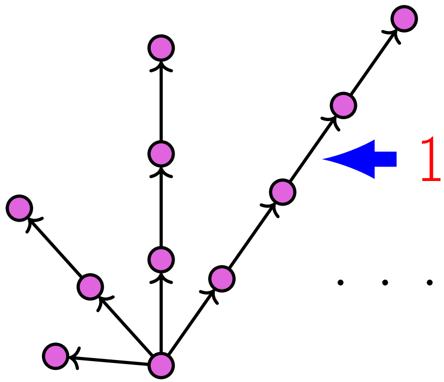
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $m_1 \to m_2 \to \dots$ point by point
- one rank r, belonging to ∀dam (meaning: he looses all infinite plays)

∃ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

∃ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) ∀dam decrements the counter (picks smaller value); (ii) ∃ve picks an edge $m \to m'$ to a new point m'.

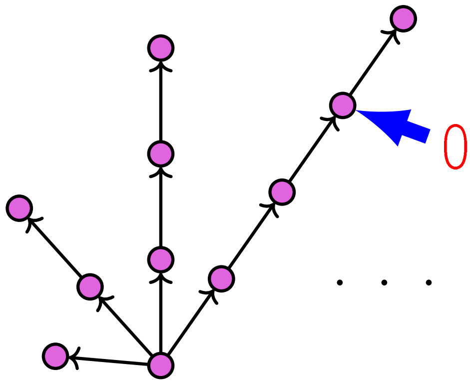
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\iff$

$\exists$ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

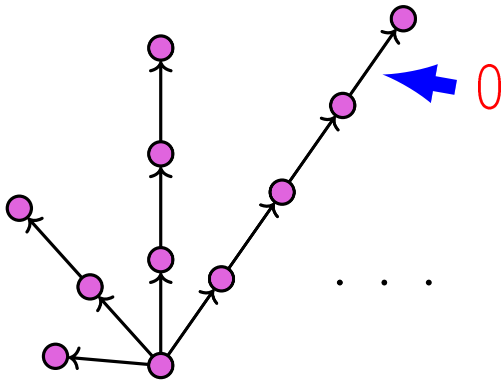
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\iff$

$\exists$ infinite path



Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

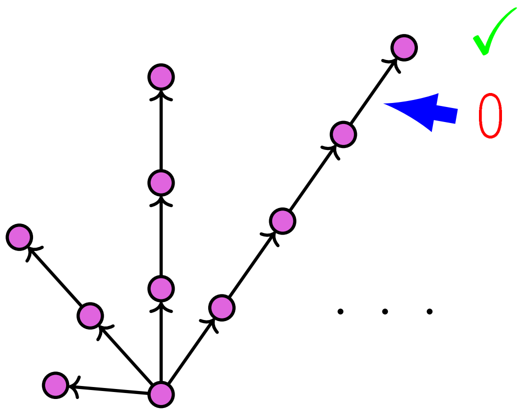
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\iff$

$\exists$ infinite path



Game for $\nu^\omega x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

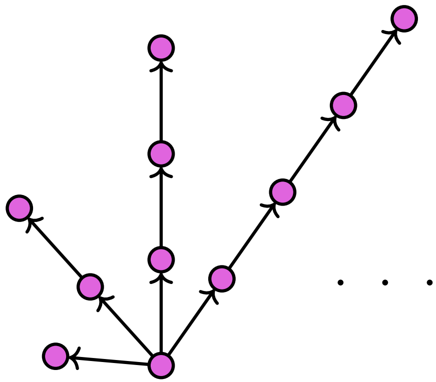
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $m_1 \to m_2 \to \ldots$ point by point
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)

$\exists$ve wins $\mathcal{G}(\nu x.\Diamond x)$

$\Longleftrightarrow$

$\exists$ infinite path



$\exists$ve wins $\mathcal{G}(\nu^{\omega} x.\Diamond x)$

$\Longleftrightarrow$

$\exists$ arbitrarily long paths

Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ plus ordinal-valued counter C initialized to ω
- In each round: (i) $\forall$dam decrements the counter (picks smaller value); (ii) $\exists$ve picks an edge $m \to m'$ to a new point m'.

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game $+$ subset $\mathcal{D} \subseteq \mathcal{R}$

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game $=$ parity game $+$ subset $\mathcal{D} \subseteq \mathcal{R}$

nonstandard ranks

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

$$\underline{\text{countdown game} = \text{parity game} + \text{subset } \mathcal{D} \subseteq \mathcal{R}}$$

- counter $\mathsf{C}_r \in \{0, 1, \ldots \omega\}$ for each $r \in \mathcal{D}$

nonstandard ranks

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

- counter $\mathsf{C}_r \in \{0, 1, ...\omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω

nonstandard ranks

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

- counter $\mathsf{C}_r \in \{0, 1, \ldots \omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathsf{C}_r})$:

nonstandard ranks

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

- counter $\mathsf{C}_r \in \{0, 1, \ldots \omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathsf{C}_r})$: ← a configuration

nonstandard ranks

$$\mathcal{G} = (V, E, \mathrm{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

nonstandard ranks

- counter $\mathsf{C}_r \in \{0, 1, \ldots\omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathsf{C}_r})$: ← a configuration
  - first, counters are updated depending on $\mathrm{rank}(v)$:

$$\mathcal{G} = (V, E, \mathrm{rank} : V \to \mathcal{R}, \mathcal{D})$$

<u>countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$</u>

nonstandard ranks

- counter $\mathrm{C}_r \in \{0, 1, \ldots\omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathrm{C}_r})$: ← a <u>configuration</u>
  - first, counters are updated depending on $\mathrm{rank}(v)$:
    - $\mathrm{C}'_r = \mathrm{C}_r$ for $r > \mathrm{rank}(v)$, [unchanged]

$$\mathcal{G} = (V, E, \text{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

- counter $C_r \in \{0, 1, \ldots \omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{C_r})$: ← a configuration

  - first, counters are updated depending on $\text{rank}(v)$:
    - $C'_r = C_r$ for $r > \text{rank}(v)$, [unchanged]
    - $C'_r = \omega$ for $r < \text{rank}(v)$, [reset]

nonstandard ranks

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

nonstandard ranks

- counter $\mathsf{C}_r \in \{0, 1, \ldots \omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathsf{C}_r})$: ← a configuration
  - first, counters are updated depending on $\mathsf{rank}(v)$:
    - $\mathsf{C}'_r = \mathsf{C}_r$ for $r > \mathsf{rank}(v)$, [unchanged]
    - $\mathsf{C}'_r = \omega$ for $r < \mathsf{rank}(v)$, [reset]
    - if $\mathsf{rank}(v) \in \mathcal{D}$, the owner of $\mathsf{rank}(v)$ chooses:

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

- counter $\mathsf{C}_r \in \{0, 1, ...\omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathsf{C}_r})$: ← a configuration
  - first, counters are updated depending on $\mathsf{rank}(v)$:
    - $\mathsf{C}'_r = \mathsf{C}_r$ for $r > \mathsf{rank}(v)$, [unchanged]
    - $\mathsf{C}'_r = \omega$ for $r < \mathsf{rank}(v)$, [reset]
    - if $\mathsf{rank}(v) \in \mathcal{D}$, the owner of $\mathsf{rank}(v)$ chooses:

$$\mathsf{C}'_{\mathsf{rank}(v)} < \mathsf{C}_{\mathsf{rank}(v)} \qquad \text{[decremented]}$$

nonstandard ranks

$$\mathcal{G} = (V, E, \mathrm{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

nonstandard ranks

- counter $\mathrm{C}_r \in \{0, 1, \ldots \omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathrm{C}_r})$: ← a configuration
  - first, counters are updated depending on $\mathrm{rank}(v)$:
    - $\mathrm{C}'_r = \mathrm{C}_r$ for $r > \mathrm{rank}(v)$, [unchanged]
    - $\mathrm{C}'_r = \omega$ for $r < \mathrm{rank}(v)$, [reset]
    - if $\mathrm{rank}(v) \in \mathcal{D}$, the owner of $\mathrm{rank}(v)$ chooses:
      $$\mathrm{C}'_{\mathrm{rank}(v)} < \mathrm{C}_{\mathrm{rank}(v)} \quad \text{[decremented]}$$
  - then, owner of v chooses vEw

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \mathcal{R}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \mathcal{R}$

- counter $\mathsf{C}_r \in \{0, 1, \ldots\omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\mathsf{C}_r})$: ← a configuration

nonstandard ranks

- first, counters are updated depending on $\mathsf{rank}(v)$:
  - $\mathsf{C}'_r = \mathsf{C}_r$ for $r > \mathsf{rank}(v)$, [unchanged]
  - $\mathsf{C}'_r = \omega$ for $r < \mathsf{rank}(v)$, [reset]
  - if $\mathsf{rank}(v) \in \mathcal{D}$, the owner of $\mathsf{rank}(v)$ chooses:

$$\mathsf{C}'_{\mathsf{rank}(v)} < \mathsf{C}_{\mathsf{rank}(v)} \qquad \text{[decremented]}$$

- then, owner of v chooses vEw
- and the game moves to $(w, \overline{\mathsf{C}'_r})$.

Game Semantics for countdown μ-ML:

Game Semantics for countdown μ-ML:

- countdown game $= (V, E, \mathsf{rank} : V \to \mathcal{R})$ plus $\mathcal{D} \subseteq \mathcal{R}$

Game Semantics for countdown μ-ML:

- countdown game $= \overbrace{(V, E, \mathsf{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\mathcal{D} \subseteq \mathcal{R}$

Game Semantics for countdown μ-ML:

- countdown game $= \overbrace{(V, E, \mathsf{rank} : V \to \mathcal{R})}^{\text{parity game}}$ $\underline{\text{plus}}$ $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

Game Semantics for countdown μ-ML:

- countdown game $= \overbrace{(V, E, \mathsf{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- semantic games for countdown μ-ML $=$ same as for μ-ML (as if μ^ω and ν^ω were μ and ν) plus nonstandard ranks $\mathcal{D}$: ranks of all immediate subformulae of countdown operators

Game Semantics for countdown μ-ML:

- countdown game $= \overbrace{(V, E, \mathsf{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- semantic games for countdown μ-ML $=$ same as for μ-ML (as if μ^ω and ν^ω were μ and ν) plus nonstandard ranks $\mathcal{D}$: ranks of all immediate subformulae of countdown operators

$$\exists\text{ve wins from } (\mathsf{m}, \varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

Game Semantics for countdown μ-ML:

- countdown game $= \overbrace{(V, E, \mathsf{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- semantic games for countdown μ-ML $=$ same as for μ-ML (as if μ^ω and ν^ω were μ and ν) plus nonstandard ranks $\mathcal{D}$: ranks of all immediate subformulae of countdown operators

$$\exists\text{ve wins from } (\mathsf{m}, \varphi) \iff \mathsf{m} \in [\![\varphi]\!]$$

works for every $\mathcal{M}$ and φ!

Countdown automata:

Countdown automata:

- countdown game $= \overbrace{(V, E, \text{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

Countdown automata:

- countdown game $= \overbrace{(V, E, \text{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- countdown automaton $= (Q, \delta, q_I, \text{rank})$ plus $\mathcal{D} \subseteq \mathcal{R}$

Countdown automata:

- countdown game $= \overbrace{(V, E, \mathrm{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- countdown automaton $= \overbrace{(Q, \delta, q_I, \mathrm{rank})}^{\text{parity automaton}}$ plus $\mathcal{D} \subseteq \mathcal{R}$

Countdown automata:

- countdown game $= \overbrace{(V, E, \mathrm{rank} : V \to \mathcal{R})}^{\text{parity game}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- countdown automaton $= \overbrace{(Q, \delta, q_I, \mathrm{rank})}^{\text{parity automaton}}$ plus $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

Countdown automata:

- countdown game $= \overbrace{(V, E, \mathrm{rank} : V \to \mathcal{R})}^{\text{parity game}}$ <u>plus</u> $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- countdown automaton $= \overbrace{(Q, \delta, q_I, \mathrm{rank})}^{\text{parity automaton}}$ <u>plus</u> $\underbrace{\mathcal{D}}_{\text{nonstandard ranks}} \subseteq \mathcal{R}$

- semantic via a countdown game: the parity game for $(Q, \delta, q_I, \mathrm{rank})$ but the ranks $\mathcal{D} \subseteq \mathcal{R}$ are now nonstandard!

countdown μ-ML

countdown automata

countdown game semantics

countdown μ-ML

countdown automata

countdown game semantics

$$\begin{tikzcd} \text{countdown } \mu\text{-ML} \arrow[r, bend left] & \text{countdown automata} \arrow[l, bend left] \end{tikzcd}$$

countdown μ-ML describes arbitrary automata

countdown game semantics

countdown μ-ML = countdown automata

countdown μ-ML describes arbitrary automata

equivalent to alternating ~~parity~~ automata
countdown!

countdown μ-calculus $\sim$ ~~parity~~ games
countdown!

wining regions definable in logic

equivalent to alternating ~~parity~~ automata
countdown!

countdown μ-calculus $\sim$ ~~parity~~ games
countdown!

COMPLICATIONS!!!

wining regions definable in logic

equivalent to alternating ~~parity~~ automata

countdown!

no simple nondeterministic model!!!

countdown μ-calculus $\sim$ ~~parity~~ games

countdown!

wining regions definable in logic

equivalent to alternating ~~parity~~ automata

countdown!

no simple nondeterministic model!!!

countdown μ-calculus $\sim$ ~~parity~~ games

countdown!

vectorial, i.e. multiple

variables bound simultaneously

wining regions definable in logic

equivalent to alternating ~~parity~~ automata

countdown!

no simple nondeterministic model!!!

countdown μ-calculus $\sim$ ~~parity~~ games

countdown!

equivalent for μ-ML

vectorial, i.e. multiple variables bound simultaneously

wining regions definable in logic

equivalent to alternating ~~parity~~ automata
countdown!

no simple nondeterministic model!!!

countdown μ-calculus $\sim$ ~~parity~~ games
countdown!

equivalent for μ-ML

vectorial, i.e. multiple variables bound simultaneously

but not for μ^{ω}-ML!!!

wining regions definable in logic

NO nondeterministic model:

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)
- this is arguably a good news: every extension of MSO closed under projections and boolean operations is too strong (contains MSO + U)

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)
- this is arguably a good news: every extension of MSO closed under projections and boolean operations is too strong (contains MSO + U)
- but the lack of nondeterministic model prevents us from copying clasical proofs

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)
- this is arguably a good news: every extension of MSO closed under projections and boolean operations is too strong (contains MSO + U)
- but the lack of nondeterministic model prevents us from copying clasical proofs
- still, alternating automata are extremely useful:

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)
- this is arguably a good news: every extension of MSO closed under projections and boolean operations is too strong (contains MSO + U)
- but the lack of nondeterministic model prevents us from copying clasical proofs
- still, alternating automata are extremely useful:
  - guarded normal form

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)
- this is arguably a good news: every extension of MSO closed under projections and boolean operations is too strong (contains MSO + U)
- but the lack of nondeterministic model prevents us from copying clasical proofs
- still, alternating automata are extremely useful:
  - guarded normal form
  - model theory (e.g. countable model property)

NO nondeterministic model:

- parity games positionally determined: nondeterministic automaton guesses the strategy
- but countdown games not positionally determined: players need to look at the counters
- countdown μ-ML provably not closed under projections (due to low topological complexity)
- this is arguably a good news: every extension of MSO closed under projections and boolean operations is too strong (contains $\mathrm{MSO} + \mathrm{U}$)
- but the lack of nondeterministic model prevents us from copying clasical proofs
- still, alternating automata are extremely useful:
  - guarded normal form
  - model theory (e.g. countable model property)
  - some decidability results

Decidability results:

Decidability results:

- (finite) model checking: given φ and m in $\mathcal{M}$, does m satisfy φ?

  decidable but not that interesting

Decidability results:

- (finite) model checking: given φ and m in $\mathcal{M}$, does m satisfy φ?

  decidable but not that interesting
- satisfiability: given φ, does there exists $\mathcal{M}$ with m satisfying φ?

Decidability results:

- (finite) model checking: given φ and m in $\mathcal{M}$, does m satisfy φ?

  decidable but not that interesting

- satisfiability: given φ, does there exists $\mathcal{M}$ with m satisfying φ?

CONJECTURE: satisfiability decidable

Decidability results:

- (finite) model checking: given φ and m in $\mathcal{M}$, does m satisfy φ?

  decidable but not that interesting

- satisfiability: given φ, does there exists $\mathcal{M}$ with m satisfying φ?

CONJECTURE: satisfiability decidable

- for now, proven in special cases:

Decidability results:

- (finite) model checking: given φ and m in $\mathcal{M}$, does m satisfy φ?

  decidable but not that interesting
- satisfiability: given φ, does there exists $\mathcal{M}$ with m satisfying φ?

CONJECTURE: satisfiability decidable

- for now, proven in special cases:
  - formulae with positive countdown, i.e. no ν^ω used

Decidability results:

- (finite) model checking: given φ and m in $\mathcal{M}$, does m satisfy φ?

  decidable but not that interesting
- satisfiability: given φ, does there exists $\mathcal{M}$ with m satisfying φ?

CONJECTURE: satisfiability decidable

- for now, proven in special cases:
  - formulae with positive countdown, i.e. no ν^ω used
  - Büchi countdown automata: only two ranks $r^\exists < r^\forall$, over infinite words

Some facts and results:

Some facts and results:

- nothing special about ω, take your favourite ordinal instead!

Some facts and results:

- nothing special about ω, take your favourite ordinal instead!
- more nesting of countdown operators $\implies$ more power

Some facts and results:

- nothing special about ω, take your favourite ordinal instead!
- more nesting of countdown operators $\implies$ more power
- fragment without nesting of countdown operators = certain multi-valued μ-ML

Some facts and results:

- nothing special about ω, take your favourite ordinal instead!
- more nesting of countdown operators $\implies$ more power
- fragment without nesting of countdown operators = certain multi-valued μ-ML

μ-ML, but with logical values from $[0, 1]$ instead of just $\{0, 1\}$
and the function $t \overset{f}{\mapsto} \frac{1}{2}t$ as an extra unary connective:

Some facts and results:

- nothing special about ω, take your favourite ordinal instead!
- more nesting of countdown operators $\implies$ more power
- fragment without nesting of countdown operators = certain multi-valued μ-ML

μ-ML, but with logical values from $[0,1]$ instead of just $\{0,1\}$
and the function $t \overset{f}{\mapsto} \frac{1}{2}t$ as an extra unary connective:

$\underline{\nu x. \Diamond \widetilde{f}(x)}$ has value 1 $\iff$ there are arbitrarily long paths

Some facts and results:

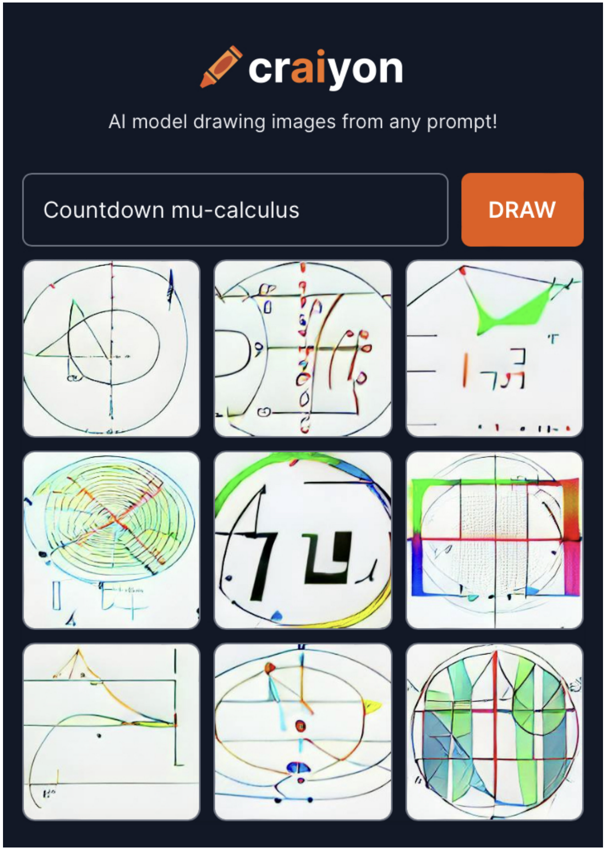
- nothing special about ω, take your favourite ordinal instead!
- more nesting of countdown operators $\implies$ more power
- fragment without nesting of countdown operators = certain multi-valued μ-ML

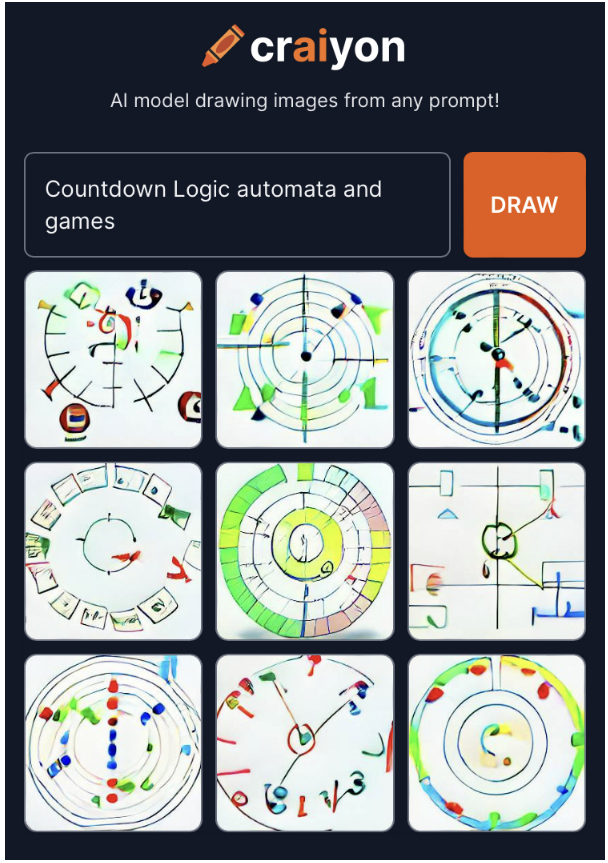
μ-ML, but with logical values from $[0,1]$ instead of just $\{0,1\}$
and the function $t \overset{f}{\mapsto} \frac{1}{2}t$ as an extra unary connective:

$\underline{\nu x.\Diamond\widetilde{f}(x)}$ has value 1 $\iff$ there are arbitrarily long paths

("$\Diamond$" means "supremum over children"; $\widetilde{f}(t) = \frac{1}{2}t + \frac{1}{2}$ is dual to f)

Thank you! :)


craiyon
AI model drawing images from any prompt!
Countdown mu-calculus
DRAW


craiyon
AI model drawing images from any prompt!
Countdown Logic automata and games
DRAW