Countdown μ-calculus

(with Automata and Games)

Jędrzej Kołodziejski (+ Bartek Klin)

MFCS 2022
Vienna

μ-calculus = modal logic + fixpoints

boolean operations + $\Diamond(\_)$: $\Diamond x$, $x \wedge \Diamond y$, ...

μ-calculus = modal logic + fixpoints

boolean operations + $\Diamond(\_)$: $\Diamond x$, $x \wedge \Diamond y$, ...

$(x, y, ... \in \mathsf{Var})$

μ-calculus = modal logic + fixpoints

boolean operations + $\Diamond(\_)$: $\Diamond x$, $x \wedge \Diamond y$, ...

$(x, y, ... \in \mathsf{Var})$

μ-calculus = modal logic + fixpoints

- interpreted in vertices of a directed graph $\mathcal{M} = (M, \to)$
  (labelled with interpretation $S \subseteq M$ for each $x \in \mathsf{Var}$)

boolean operations + $\Diamond(\_)$: $\Diamond x$, $x \wedge \Diamond y$, ...

$(x, y, \ldots \in \mathrm{Var})$

μ-calculus = modal logic + fixpoints

- interpreted in vertices of a directed graph $\mathcal{M} = (M, \rightarrow)$
  (labelled with interpretation $S \subseteq M$ for each $x \in \mathrm{Var}$)
- "$\Diamond x$" means "there exists a child satisfying x"

boolean operations + $\Diamond(\_)$: $\Diamond x$, $x \wedge \Diamond y$, ...

($x, y, \ldots \in \mathrm{Var}$)

μ-calculus = modal logic + fixpoints

$\nu x.\Diamond x, \quad \mu y.x \wedge \Diamond y \ldots$

- interpreted in vertices of a directed graph $\mathcal{M} = (M, \to)$
  (labelled with interpretation $S \subseteq M$ for each $x \in \mathrm{Var}$)
- "$\Diamond x$" means "there exists a child satisfying x"

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

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

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

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

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

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

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

- 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 $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

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

- 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 $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

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

- 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 $F : \mathcal{P}(M) \to \mathcal{P}(M)$:

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

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

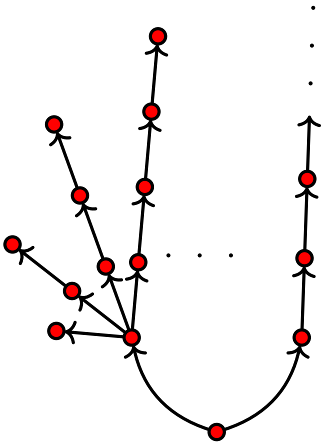
$$S \subseteq S' \implies F(S) \subseteq F(S')$$

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

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

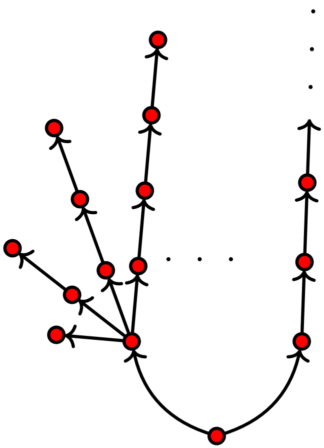
- we compute fixpoints by (transfinite) iteration of F:

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



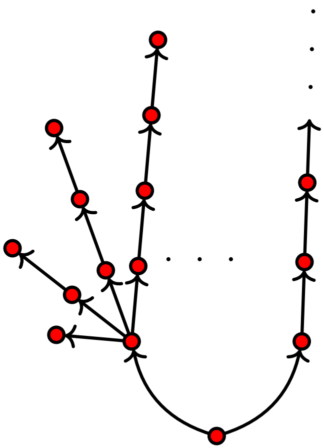
$S_0 = M$

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

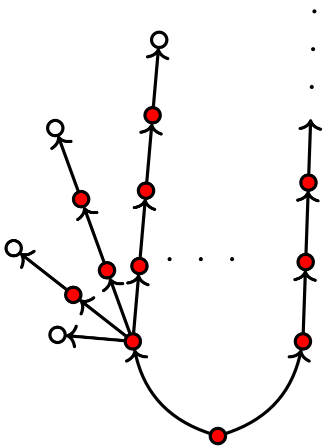


$S_0 = M$ $S_1 = F(S_0)$

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

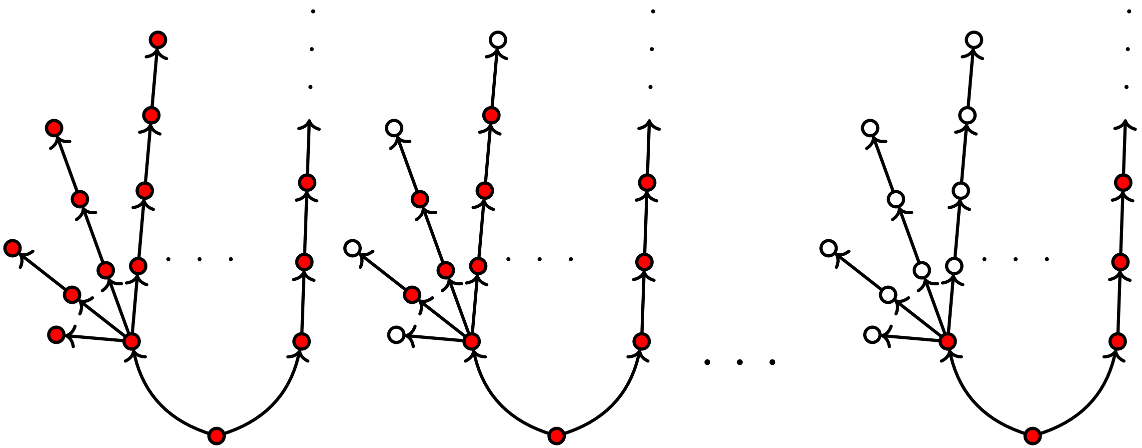


$S_0 = M$



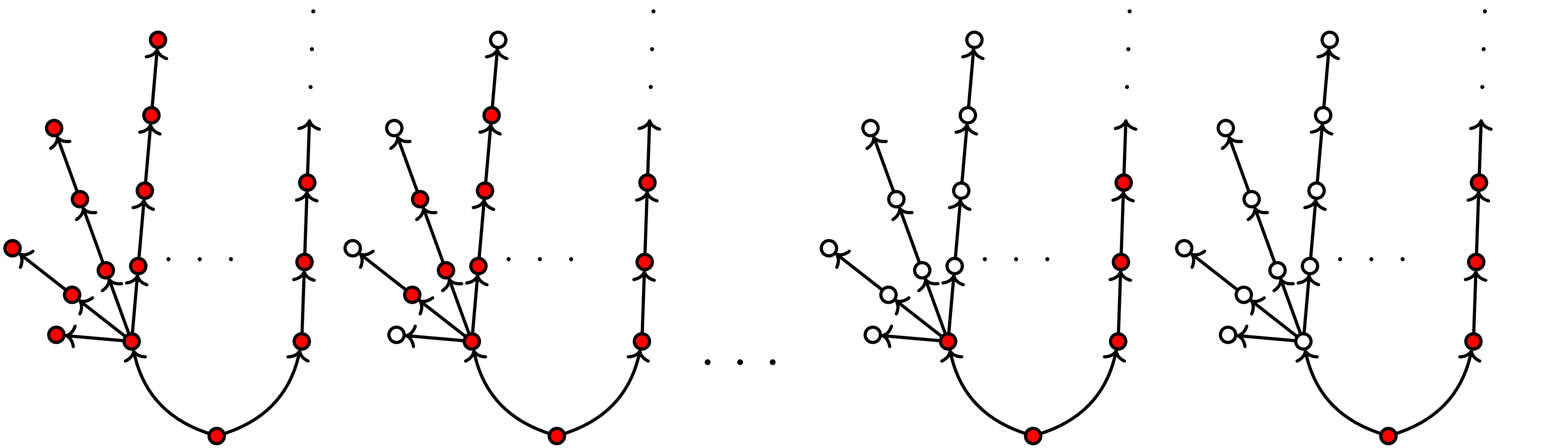
$S_1 = F(S_0) =$ "$\exists$ a child"

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



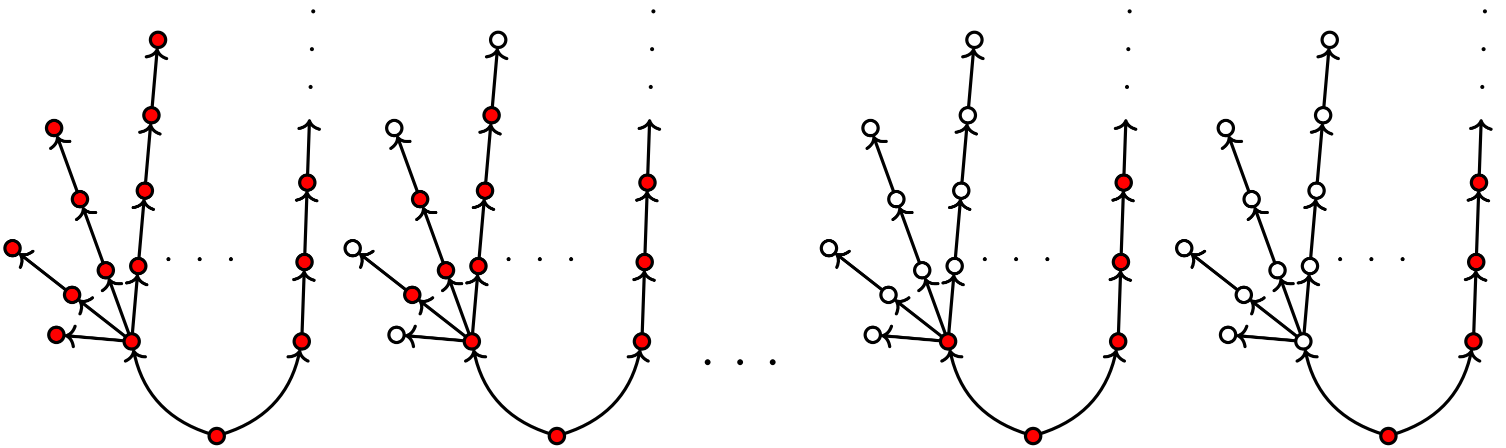
$S_0 = M$ $S_1 = F(S_0) =$"$\exists$ a child" $S_\omega =$ "arbitrarily long paths"

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



$S_0 = M$ $S_1 = F(S_0) =$ "∃ a child" $S_\omega =$ "arbitrarily long paths" $S_{\omega+1} = S_{\omega+2} = \text{GFP}.F$

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

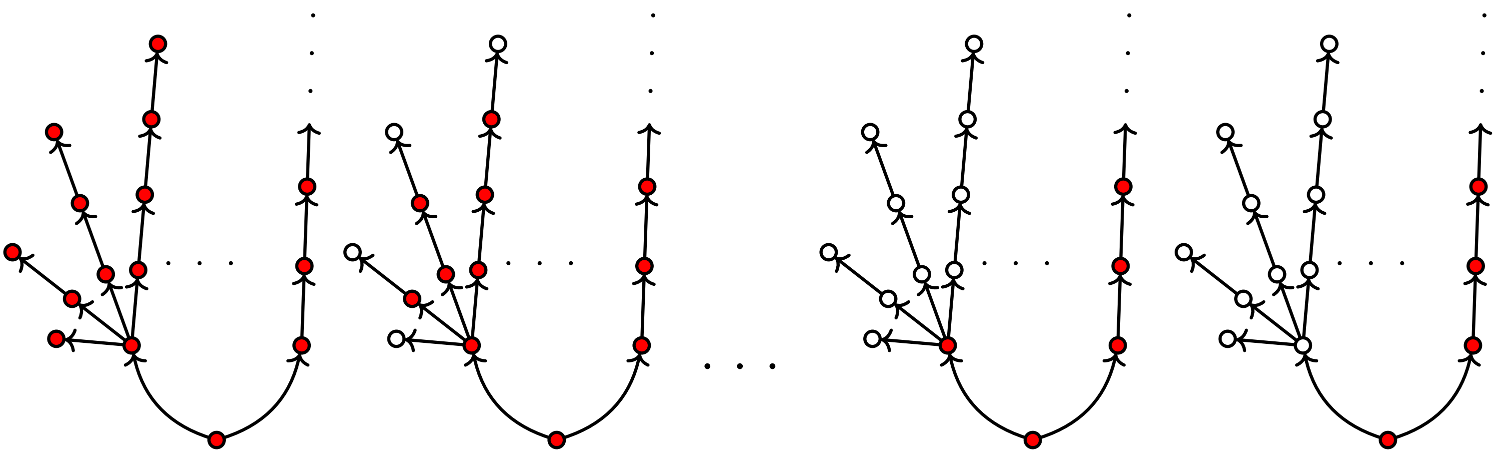


$S_0 = M$ $S_1 = F(S_0) =$ "$\exists$ a child" $S_\omega =$ "arbitrarily long paths" $S_{\omega+1} = S_{\omega+2} = \mathrm{GFP}.F$

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

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

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



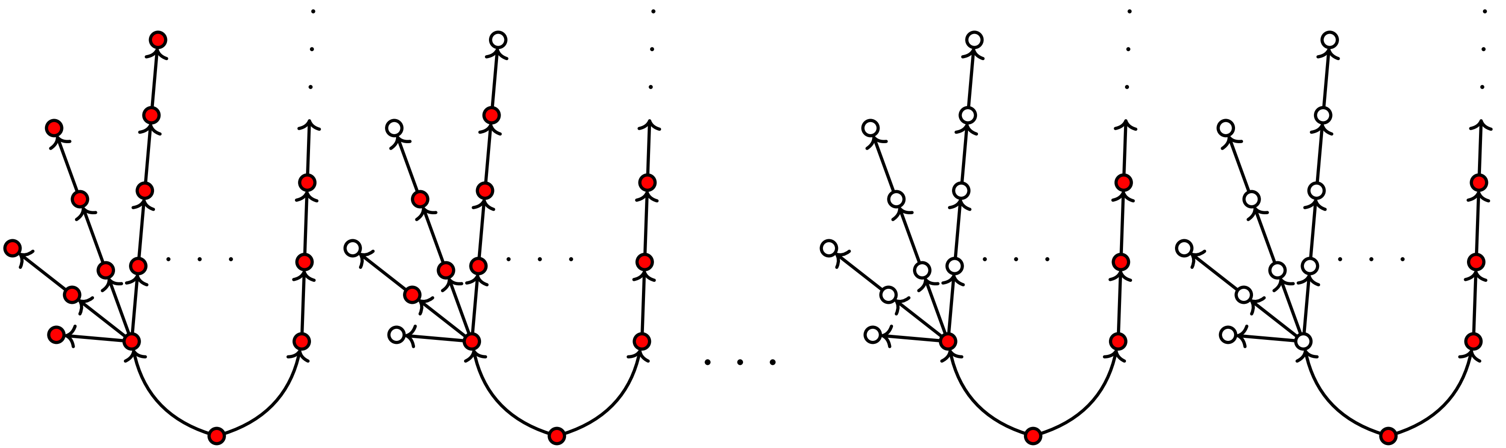
$S_0 = M$ $\quad$ $S_1 = F(S_0) =$ "$\exists$ a child" $\quad$ $S_\omega =$ "arbitrarily long paths" $\quad$ $S_{\omega+1} = S_{\omega+2} = \mathrm{GFP}.F$

$$\|$$

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

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

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



$S_0 = M$ $S_1 = F(S_0) =$ "$\exists$ a child" ... $S_\omega =$ "arbitrarily long paths" $S_{\omega+1} = S_{\omega+2} = \mathrm{GFP}.F$

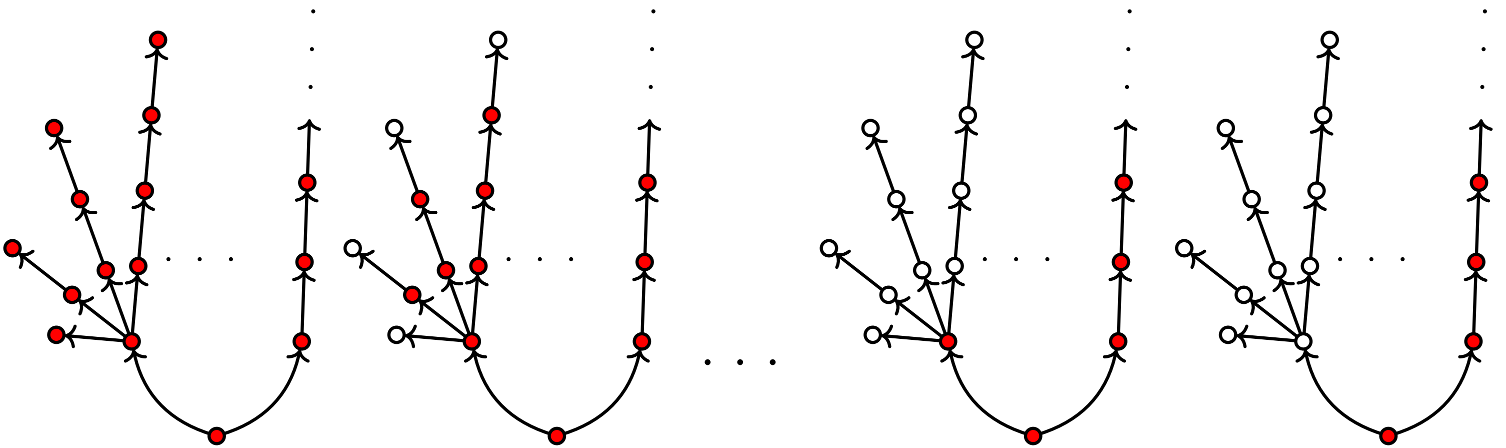
$S_\omega = [\![\nu^{\omega} x.\Diamond x]\!]$

$S_{\omega+1} = [\![\nu x.\Diamond x]\!]$

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

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

countdown calculus = μ-calculus + ν^{ω}



$S_0 = M$ $\quad$ $S_1 = F(S_0) =$ "$\exists$ a child" $\quad$ $S_\omega =$ "arbitrarily long paths" $\quad$ $S_{\omega+1} = S_{\omega+2} = \mathrm{GFP}.F$

$S_\omega = [\![\nu^{\omega} x.\Diamond x]\!]$ $\quad$ $S_{\omega+1} = [\![\nu x.\Diamond x]\!]$

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

$$\mathcal{G} = (V, E, \text{rank} : V \rightarrow \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

$$\mathcal{G} = (V, E, \text{rank} : V \to \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

nonstandard ranks

$$\mathcal{G} = (V, E, \mathsf{rank} : V \to \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \to \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

nonstandard ranks

$$\mathcal{G} = (V, E, \mathrm{rank} : V \rightarrow \{0,\ldots,\mathrm{d}\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,\mathrm{d}\}$

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

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \to \{0,\dots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\dots,d\}$

- counter $C_r \in \{0, 1, \dots\omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{C_r})$:
  - first, owner of v chooses vEw,

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \rightarrow \{0,\dots,\text{d}\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\dots,\text{d}\}$

- counter $\text{C}_r \in \{0, 1, \dots \omega\}$ for each $r \in \mathcal{D}$
- initially all C_r equal ω
- from $(v, \overline{\text{C}_r})$:
  - first, owner of v chooses vEw,
  - then, counters are updated depending on $\text{rank}(w)$:

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \to \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \rightarrow \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \rightarrow \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

nonstandard ranks

$$\mathcal{G} = (V, E, \mathrm{rank} : V \to \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

$$C'_{\mathrm{rank}(w)} < C_{\mathrm{rank}(w)}$$ [decremented]

nonstandard ranks

$$\mathcal{G} = (V, E, \text{rank} : V \to \{0,\ldots,d\}, \mathcal{D})$$

countdown game = parity game + subset $\mathcal{D} \subseteq \{0,\ldots,d\}$

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

      $$C'_{\text{rank}(w)} < C_{\text{rank}(w)}$$ [decremented]
  - and the game moves to $(w, \overline{C'_r})$.

nonstandard ranks

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

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

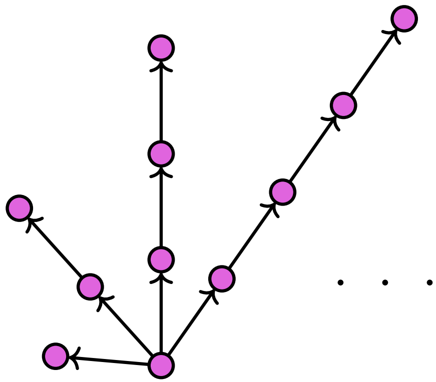
- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex

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

- ∃ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- one rank r, belonging to ∀dam (meaning: he looses all infinite plays)

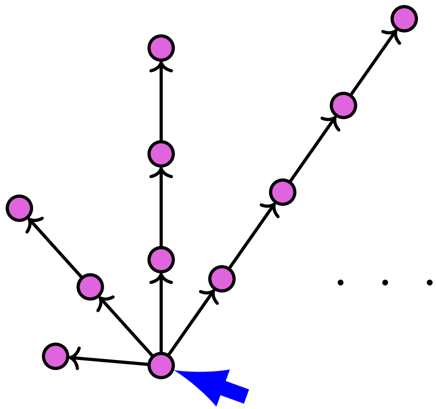
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



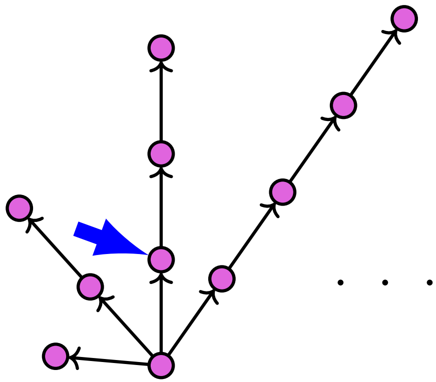
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



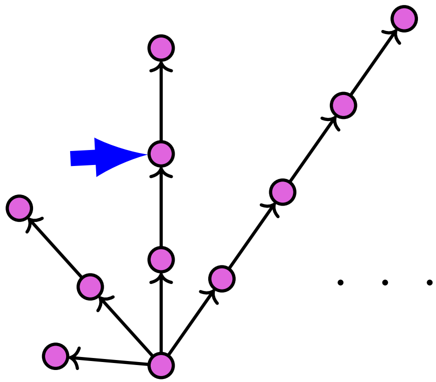
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- one rank r, belonging to ∀dam (meaning: he looses all infinite plays)



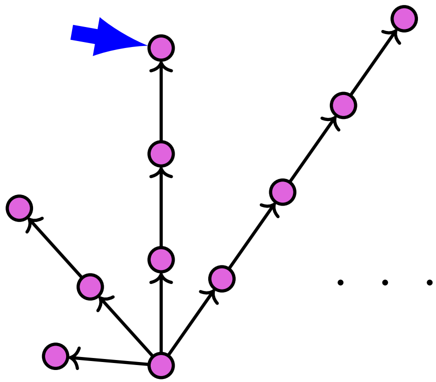
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



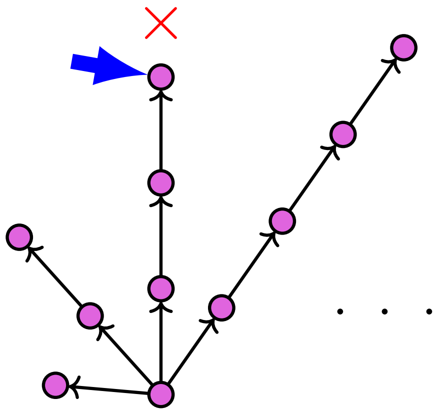
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



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

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- one rank r, belonging to $\forall$dam (meaning: he looses all infinite plays)



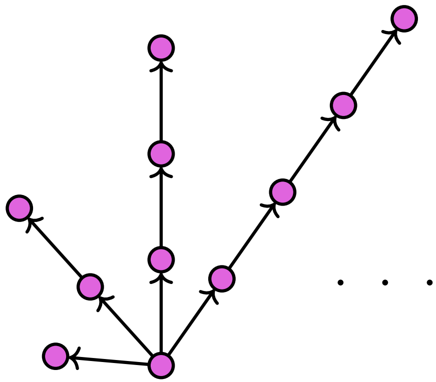
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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



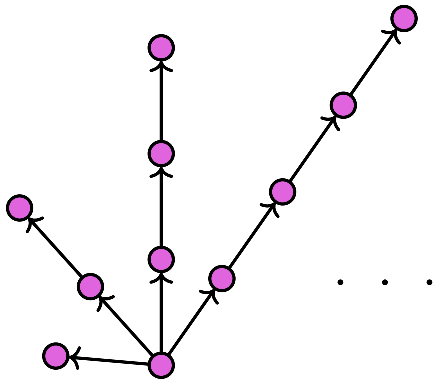
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \rightarrow v_2 \rightarrow \ldots$ vertex by vertex
- 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 $\nu^{\omega} x.\Diamond x$:

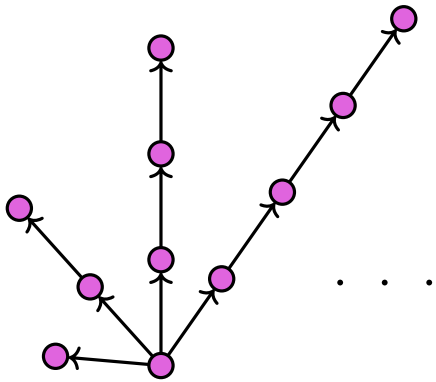
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard

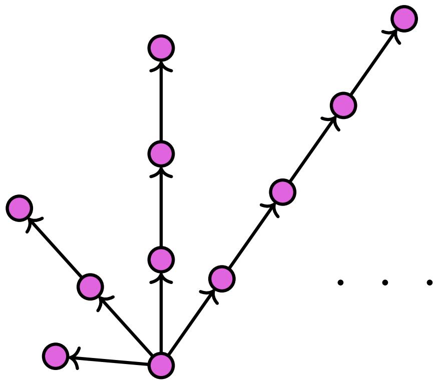
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

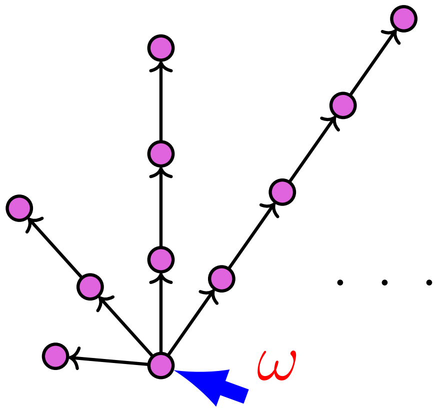
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, ∀dam picks counter value $C_r = \alpha < \omega$. Then, after each ∃ve's move he decrements C_r.

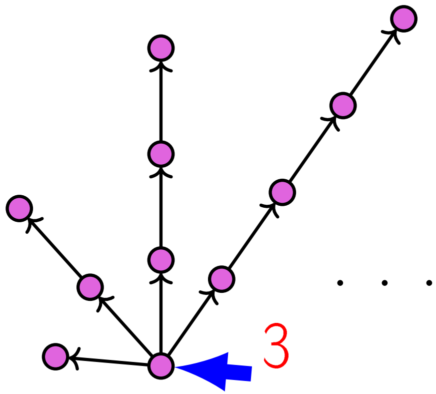
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

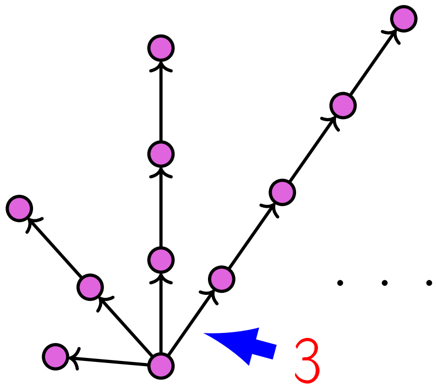
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, ∀dam picks counter value $C_r = \alpha < \omega$. Then, after each ∃ve's move he decrements C_r.

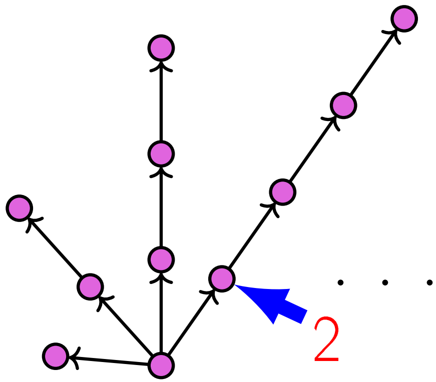
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

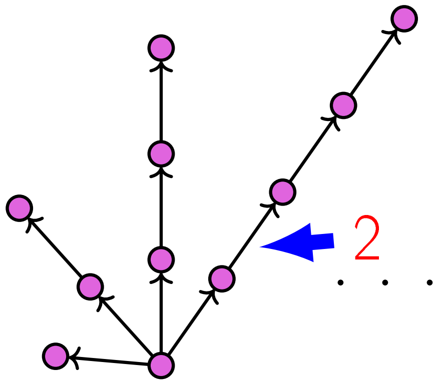
Game for $\nu x.\Diamond x$:

- ▶ $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- ▶ 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- ▶ first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

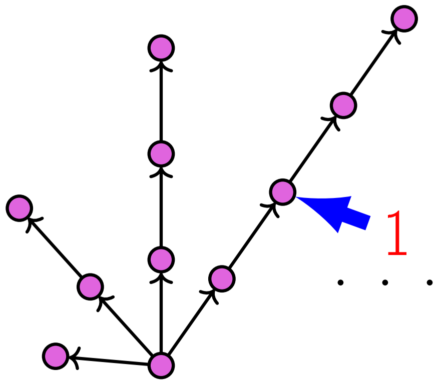
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

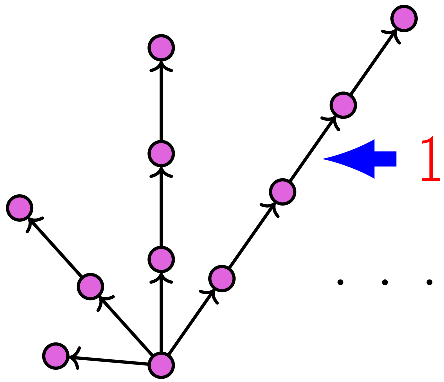
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, ∀dam picks counter value $C_r = \alpha < \omega$. Then, after each ∃ve's move he decrements C_r.

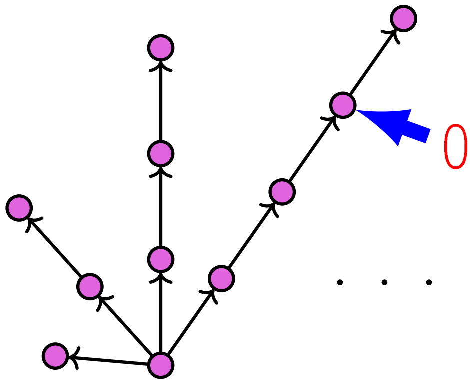
Game for $\nu x.\Diamond x$:

- $\exists$ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

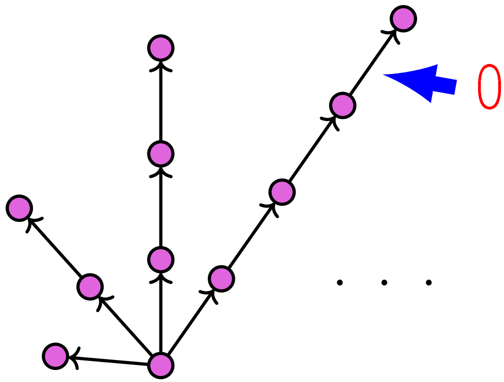
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \to v_2 \to \ldots$ vertex by vertex
- 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 $\nu x.\Diamond x$ except that rank r is nonstandard
- first, ∀dam picks counter value $C_r = \alpha < \omega$. Then, after each ∃ve's move he decrements C_r.

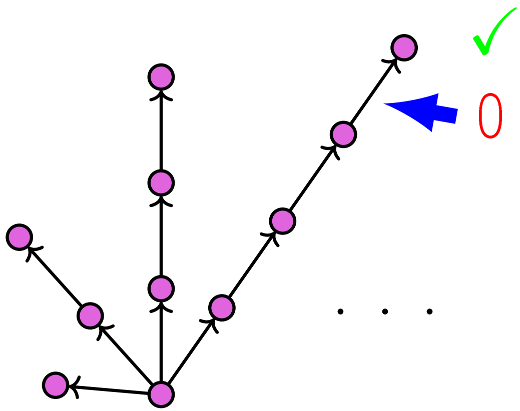
Game for $\nu x.\Diamond x$:

- ∃ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- one rank r, belonging to ∀dam (meaning: he looses all infinite plays)

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

$\iff$

∃ infinite path



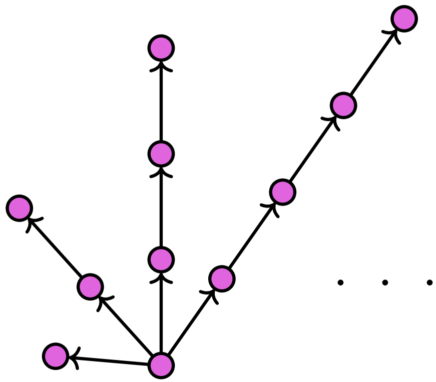
Game for $\nu^{\omega} x.\Diamond x$:

- same as for $\nu x.\Diamond x$ except that rank r is nonstandard
- first, ∀dam picks counter value $C_r = \alpha < \omega$. Then, after each ∃ve's move he decrements C_r.

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

- $\exists$ve picks a path $v_1 \to v_2 \to \dots$ vertex by vertex
- 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



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

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

- same as for $\nu x.\Diamond x$ except that rank r is nonstandard
- first, $\forall$dam picks counter value $C_r = \alpha < \omega$. Then, after each $\exists$ve's move he decrements C_r.

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!

vectorial, i.e. multiple variables bound simultaneously

wining regions definable in logic

equivalent to alternating ~~parity~~ automata
countdown!

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!

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

countdown!

vectorial, i.e. multiple variables bound simultaneously

equivalent for μ-ML

but not for μ^{ω}-ML!!!

wining regions definable in logic

Some facts and results

Some facts and results

- (finite) model checking is decidable

Some facts and results

- (finite) model checking is decidable
- satisfiability decidable for fragments (conjecture: full logic decidable)

Some facts and results

- (finite) model checking is decidable
- satisfiability decidable for fragments (conjecture: full logic decidable)
- low topological complexity, so provably not closed under projections

Some facts and results

- (finite) model checking is decidable
- satisfiability decidable for fragments (conjecture: full logic decidable)
- low topological complexity, so provably not closed under projections
- fragment without nesting of countdown operators = certain multi-valued μ-ML

Some facts and results

- (finite) model checking is decidable
- satisfiability decidable for fragments (conjecture: full logic decidable)
- low topological complexity, so provably not closed under projections
- 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 $t \mapsto \frac{1}{2}t$ as an extra unary connective

Some facts and results

- (finite) model checking is decidable
- satisfiability decidable for fragments (conjecture: full logic decidable)
- low topological complexity, so provably not closed under projections
- fragment without nesting of countdown operators = certain multi-valued μ-ML
- more nesting of countdown operators $\Longrightarrow$ more power

μ-ML, but with logical values from $[0,1]$ instead of just $\{0,1\}$

and the function $t \mapsto \frac{1}{2}t$ as an extra unary connective

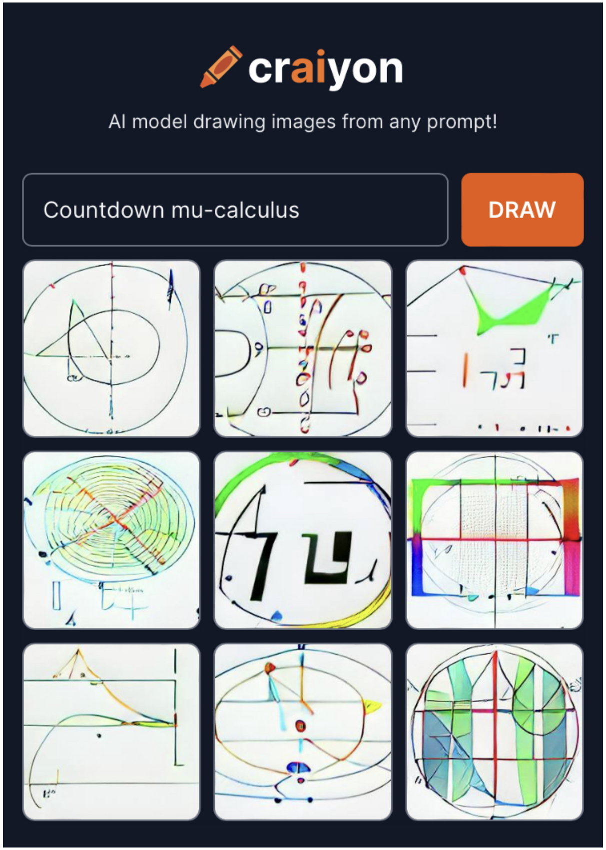
Some facts and results

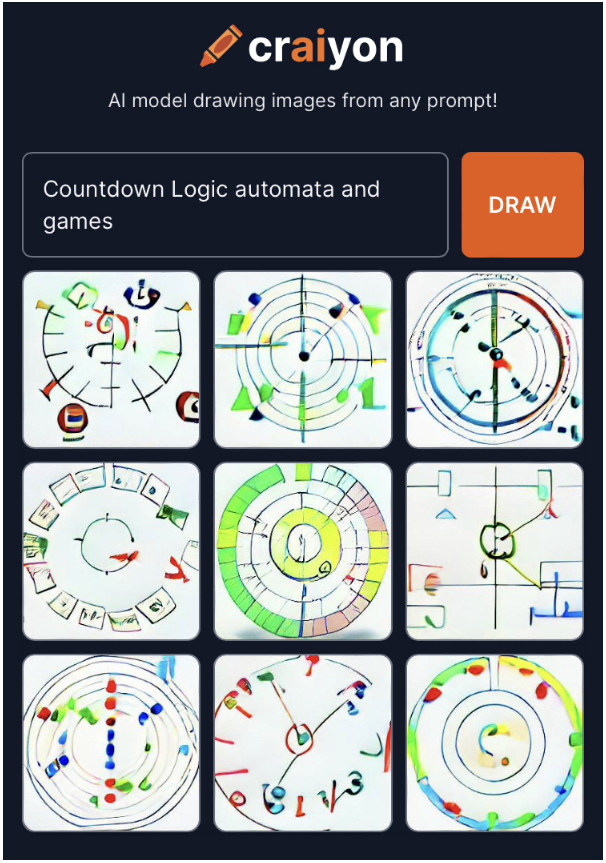
- (finite) model checking is decidable
- satisfiability decidable for fragments (conjecture: full logic decidable)
- low topological complexity, so provably not closed under projections
- fragment without nesting of countdown operators = certain multi-valued μ-ML
- more nesting of countdown operators $\implies$ more power
- nothing special about ω, take your favourite ordinal instead!

μ-ML, but with logical values from $[0,1]$ instead of just $\{0,1\}$

and the function $t \mapsto \frac{1}{2}t$ as an extra unary connective

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