Proof Systems for the Modal μ-Calculus Obtained by Determinizing Automata

Maurice Dekker, Johannes Kloibhofer, Johannes Marti, Yde Venema

September 21, 2023

Institute for Logic, Language and Computation
University of Amsterdam, Netherlands

Outline

- Define infinitary proof system for the modal μ-calculus based on traces
- Show how traces can be checked by automaton
- Integrate automaton in proof system - first determinize it
- Different determinization methods give different proof systems

Outline

- Define infinitary proof system for the modal μ-calculus based on traces
- Show how traces can be checked by automaton
- Integrate automaton in proof system - first determinize it
- Different determinization methods give different proof systems

(• Define specific determinization method for parity automata)

- Obtain proof system from it

Modal μ-calculus

Formulas in the modal μ-calculus are generated by the grammar

$$\varphi \ ::= \ p \mid \overline{p} \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu x\, \varphi \mid \nu x\, \varphi$$

Modal μ-calculus

Formulas in the modal μ-calculus are generated by the grammar

$$\varphi ::= p \mid \overline{p} \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu x\, \varphi \mid \nu x\, \varphi$$

- Formulas of the form $\mu x\ \varphi$ and $\nu x\ \varphi$ are called fixpoint formulas: $\mu x\ \varphi \equiv \varphi[\mu x\ \varphi/x]$ and $\nu x\ \varphi \equiv \varphi[\nu x\ \varphi/x]$
- In $\mu x\ \varphi$ and $\nu x\ \varphi$ there are no occurrences of $\overline{x}$ in φ
- A fixpoint formula φ is *more important* than a fixpoint formula ψ if φ is a subformula of ψ

Proof theory of the modal μ-calculus

- [Kozen '83] introduced *finitary Hilbert system*
- *Completeness* proven by [Walukiewicz '96]
- [Niwiński, Walukiewicz '95] introduced infinitary *tableaux games* in which one player has winning strategy iff formula is valid

NW derivations

An NW *derivation* is a, possibly infinite, tree defined from the following rules:

Ax: $\dfrac{}{p, \bar{p}, \Gamma}$

$\mathsf{R}_\vee$: $\dfrac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma}$

$\mathsf{R}_\wedge$: $\dfrac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$

$\mathsf{R}_\Box$: $\dfrac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma, \Delta}$

R_μ: $\dfrac{\varphi[\mu x\, \varphi/x], \Gamma}{\mu x\, \varphi, \Gamma}$

R_ν: $\dfrac{\varphi[\nu x\, \varphi/x], \Gamma}{\nu x\, \varphi, \Gamma}$

NW derivations

An NW *derivation* is a, possibly infinite, tree defined from the following rules:

$$\text{Ax: } \frac{}{p, \bar{p}, \Gamma} \qquad \text{R}_\vee\text{: } \frac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma} \qquad \text{R}_\wedge\text{: } \frac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$$

$$\text{R}_\Box\text{: } \frac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma, \Delta} \qquad \text{R}_\mu\text{: } \frac{\varphi[\mu x\, \varphi/x], \Gamma}{\mu x\, \varphi, \Gamma} \qquad \text{R}_\nu\text{: } \frac{\varphi[\nu x\, \varphi/x], \Gamma}{\nu x\, \varphi, \Gamma}$$

- There are infinite branches
- But only finitely many sequents

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\vdots}{\mu x\Box x, \nu y\Diamond y}}{\Box(\mu x\Box x), \Diamond(\nu y\Diamond y)}\,\mathsf{R}_\Box}{\Box(\mu x\Box x), \nu y\Diamond y}\,\mathsf{R}_\nu}{\mu x\Box x, \nu y\Diamond y}\,\mathsf{R}_\mu}{\mu x\Box x \lor \nu y\Diamond y}\,\mathsf{R}_\lor$$

Figure 1: NW derivation of $\mu x\Box x \lor \nu y\Diamond y$

NW proofs

- A trace $(\varphi_j)_{j\in\omega}$ on an infinite branch is an infinite sequence of formulas s.t. φ_j is an immediate ancestor of φ_{j+1} for $j \in \omega$.
- A trace is called ν-trace if the most important fixpoint formula unfolded infinitely often is a ν-formula.

NW proofs

- A trace $(\varphi_j)_{j\in\omega}$ on an infinite branch is an infinite sequence of formulas s.t. φ_j is an immediate ancestor of φ_{j+1} for $j \in \omega$.
- A trace is called ν-trace if the most important fixpoint formula unfolded infinitely often is a ν-formula.

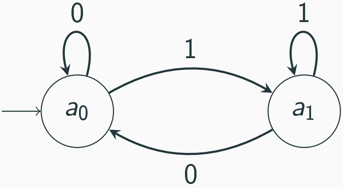
Definition
An NW *proof* is an NW derivation, where on every infinite branch there is a ν-trace.

Question
How to recognize those branches?

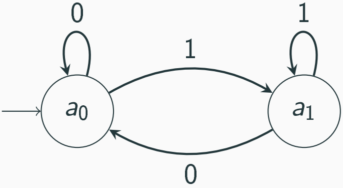
$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\mu x \Box x, \nu y \Diamond y}}{\Box(\mu x \Box x), \Diamond(\nu y \Diamond y)}\ \mathsf{R}_{\Box}}{\Box(\mu x \Box x), \nu y \Diamond y}\ \mathsf{R}_{\nu}}{\mu x \Box x, \nu y \Diamond y}\ \mathsf{R}_{\mu}}{\mu x \Box x \lor \nu y \Diamond y}\ \mathsf{R}_{\lor}
$$

Figure 2: NW proof of $\mu x \Box x \lor \nu y \Diamond y$

ω-automata are a variation of finite state automata with infinite strings as inputs:



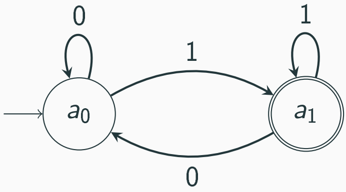
ω-automata are a variation of finite state automata with infinite strings as inputs:



The acceptance condition can be given in different ways:

Büchi automata

ω-automata are a variation of finite state automata with infinite strings as inputs:

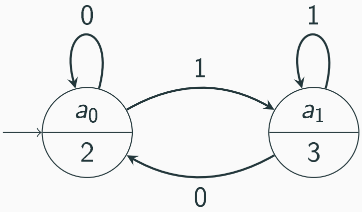


The acceptance condition can be given in different ways:

- A *Büchi* condition is given as a subset $F \subseteq A$
  - $\star$ Runs accepted, that contain infinitely many states in F

Parity automata

ω-automata are a variation of finite state automata with infinite strings as inputs:



The acceptance condition can be given in different ways:

- A *parity* condition is given as a map $\Omega : A \to \mathbb{N}$
  $\star$ Runs α accepted, s.t.
  $\max\{\Omega(a) \mid a \text{ occurs infinitely often in } \alpha\}$ is even

Recap NW proofs

An NW *derivation* is a, possibly infinite, tree defined from the following rules:

$$\text{Ax1: } \frac{}{p, \bar{p}, \Gamma} \quad \text{Ax2: } \frac{}{\top, \Gamma} \qquad \text{R}_\vee\text{: } \frac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma} \qquad \text{R}_\wedge\text{: } \frac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$$

$$\text{R}_\Box\text{: } \frac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma, \Delta} \qquad \text{R}_\mu\text{: } \frac{\varphi[\mu x\ \varphi/x], \Gamma}{\mu x\ \varphi, \Gamma} \qquad \text{R}_\nu\text{: } \frac{\varphi[\nu x\ \varphi/x], \Gamma}{\nu x\ \varphi, \Gamma}$$

Definition
An NW *proof* is an NW derivation, where on every infinite branch there is a ν-trace.

Tracking automaton

Define nondeterministic parity automaton $\mathbb{A}$ s.t.

$$\mathbb{A} \text{ accepts } \alpha \Leftrightarrow \text{ there is a } \nu\text{-trace on } \alpha$$

for all infinite branches α in an NW derivation

Tracking automaton

Define nondeterministic parity automaton $\mathbb{A}$ s.t.

$$\mathbb{A} \text{ accepts } \alpha \Leftrightarrow \text{ there is a } \nu\text{-trace on } \alpha$$

for all infinite branches α in an NW derivation

Idea:

- States are formulas
- Transitions given by ancestor relation
- Parity of fixpoint formulas:
    - ν-formulas get even parity
    - μ-formulas get odd parity
    - More important fixpoint formulas get higher parity

Obtaining new proof system

Idea: build automaton into proof system

- Sequents of form $a \vdash \Gamma$, where a state of tracking automaton $\mathbb{A}$

Obtaining new proof system

Idea: build automaton into proof system

- Sequents of form $a \vdash \Gamma$, where a state of tracking automaton $\mathbb{A}$

Need automaton to be deterministic!

Obtaining new proof system

Idea: build automaton into proof system

- Sequents of form $a \vdash \Gamma$, where a state of tracking automaton $\mathbb{A}$

Need automaton to be deterministic!

Let $\mathbb{A}^D$ be deterministic automaton accepting same language as $\mathbb{A}$

- Sequents of form $a \vdash \Gamma$, where a state of $\mathbb{A}^D$

Advantage: Global validity condition based on branches instead of traces

Challenge
Integrate states into sequents

Explicit determinization

- Most known determinization method is *Safra construction*
- Inspired by it [Jungteerapanich '10] and [Stirling '14] introduced annotated proof system
  - Sequents have form $\theta \vdash \varphi_1^{\rho_1}, \ldots, \varphi_n^{\rho_n}$

Explicit determinization

- Most known determinization method is *Safra construction*
- Inspired by it [Jungteerapanich '10] and [Stirling '14] introduced annotated proof system
    - Sequents have form $\theta \vdash \varphi_1^{\rho_1}, \ldots, \varphi_n^{\rho_n}$

- We develop determinization method for nondeterministic automata using binary trees
- States of deterministic automaton $\mathbb{B}$ consists of
    - Sets of states of $\mathbb{A}$
    - Every state annotated by tuple of binary strings

Explicit determinization

- Most known determinization method is *Safra construction*
- Inspired by it [Jungteerapanich '10] and [Stirling '14] introduced annotated proof system
    - Sequents have form $\theta \vdash \varphi_1^{\rho_1}, \ldots, \varphi_n^{\rho_n}$
- We develop determinization method for nondeterministic automata using binary trees
- States of deterministic automaton $\mathbb{B}$ consists of
    - Sets of states of $\mathbb{A}$
    - Every state annotated by tuple of binary strings
- Using this method we get different annotated proof system
    - Sequents have form $\vdash \varphi_1^{\sigma_1}, \ldots, \varphi_n^{\sigma_n}$
    - No extra information needed!

BT proof rules

Ax: $\dfrac{}{p^\sigma, \bar{p}^\tau, \Gamma}$ R$_\vee$: $\dfrac{\varphi^\sigma, \psi^\sigma, \Gamma}{(\varphi \vee \psi)^\sigma, \Gamma}$ R$_\wedge$: $\dfrac{\varphi^\sigma, \Gamma \quad \psi^\sigma, \Gamma}{(\varphi \wedge \psi)^\sigma, \Gamma}$

R$_\Box$: $\dfrac{\varphi^\sigma, \Gamma}{\Box\varphi^\sigma, \Diamond\Gamma, \Delta}$ R$_\nu$: $\dfrac{\varphi[x \backslash \nu x\ \varphi]^{\sigma\restriction k \cdot 1_k}, \Gamma^{\cdot 0_k}}{\nu x\ \varphi^\sigma, \Gamma}$ where $k = \Omega_\Phi(\nu x\ \varphi)$

R$_\mu$: $\dfrac{\varphi[x \backslash \mu x\ \varphi]^{\sigma\restriction \Omega_\Phi(\mu x\ \varphi)}, \Gamma}{\mu x\ \varphi^\sigma, \Gamma}$ Resolve: $\dfrac{\varphi^\sigma, \Gamma}{\varphi^\sigma, \varphi^\tau, \Gamma}$ where $\sigma > \tau$

Compress$_k^{s0}$: $\dfrac{\varphi_1^{(\ldots, st_1, \ldots)}, \ldots, \varphi_n^{(\ldots, st_n, \ldots)}, \Gamma}{\varphi_1^{(\ldots, s0t_1, \ldots)}, \ldots, \varphi_n^{(\ldots, s0t_n, \ldots)}, \Gamma}$ where $s \notin \Gamma_k^A$

Compress$_k^{s1}$: $\dfrac{\varphi_1^{(\ldots, st_1, \ldots)}, \ldots, \varphi_n^{(\ldots, st_n, \ldots)}, \Gamma}{\varphi_1^{(\ldots, s1t_1, \ldots)}, \ldots, \varphi_n^{(\ldots, s1t_n, \ldots)}, \Gamma}$ where $s \notin \Gamma_k^A$ and $s \neq 0 \cdots 0$

BT^∞ proofs

Definition
A BT^∞ proof is a BT derivation, where on every infinite branch there is a successful string.

- Completeness and Soundness of BT^∞ proved by using determinization method
- Advantage: Global validity condition on branches instead of traces

cyclic BT proofs

- Only finitely many sequents on infinite branch
- Get cyclic proof tree
- Infinite branches correspond to strongly connected subgraphs

cyclic BT proofs

- Only finitely many sequents on infinite branch
- Get cyclic proof tree
- Infinite branches correspond to strongly connected subgraphs

Definition
A BT proof is a finite BT derivation, where for every strongly connected subgraph there is a successful string.

- Comparing to Jungteerapanich system: Trade-off between extra information and more local validity condition

Other logics

Same method could be applied to other proof systems:

Other logics

Same method could be applied to other proof systems:

- Alternation-free mu-calculus:
    - Weak co-Büchi automaton
    - Determinization corresponds to Focus system [Marti, Venema '21]

Same method could be applied to other proof systems:

- Alternation-free mu-calculus:
  - Weak co-Büchi automaton
  - Determinization corresponds to Focus system [Marti, Venema '21]
- $\mathrm{FOL}_{\mathrm{ID}}$, Cyclic PA, etc...
  - Büchi automaton
  - Binary strings as annotations

Conclusion

- Showed how determinization methods give rise to proof systems
- Introduced determinization method for nondeterministic parity automata
- Explicitly used this method to obtain proof system for the modal μ-calculus

Future work

- Compare BT and Jungteerapanich system
  - $\star$ Path-based condition without control?
- Show completeness of Kozen's axiomatization:
  - $\star$ Idea: translate BT to Kozen's system
- Apply method to other proof systems

Thank you !