Using automata theory to obtain a new proof system for the modal μ-calculus

Johannes Kloibhofer
(j.w.w. Maurice Dekker, Johannes Marti, Yde Venema)

April 21, 2023

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

Outline

- Preliminaries:
  - Modal μ-calculus
  - Proof system for the μ-calculus
  - Automata theory

Outline

- Preliminaries:
  - Modal μ-calculus
  - Proof system for the μ-calculus
  - Automata theory
- Show connections of automata theory and proof system

Outline

- Preliminaries:
    - Modal μ-calculus
    - Proof system for the μ-calculus
    - Automata theory
- Show connections of automata theory and proof system
- Introduce determinisation method for parity automata

Outline

- Preliminaries:
  - Modal μ-calculus
  - Proof system for the μ-calculus
  - Automata theory
- Show connections of automata theory and proof system
- Introduce determinisation method for parity automata
- Define proof system using automata

Outline

- Preliminaries:
    - Modal μ-calculus
    - Proof system for the μ-calculus
    - Automata theory
- Show connections of automata theory and proof system
- Introduce determinisation method for parity automata
- Define proof system using automata
- Discuss benefits of this system

Modal μ-calculus

The *formulas* in the modal μ-calculus are generated by the grammar

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

Modal μ-calculus

The *formulas* in the modal μ-calculus are generated by the grammar

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

- Formulas of the form $\mu x \ \varphi$ and $\nu x \ \varphi$ are called *fixpoint formulas* and interpreted as the least and greatest fixpoint of φ

Modal μ-calculus

The *formulas* in the modal μ-calculus are generated by the grammar

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

- Formulas of the form $\mu x \ \varphi$ and $\nu x \ \varphi$ are called *fixpoint formulas* and interpreted as the least and greatest fixpoint of φ
- In $\mu x \ \varphi$ and $\nu x \ \varphi$ there are no occurrences of $\overline{x}$ in φ

Modal μ-calculus

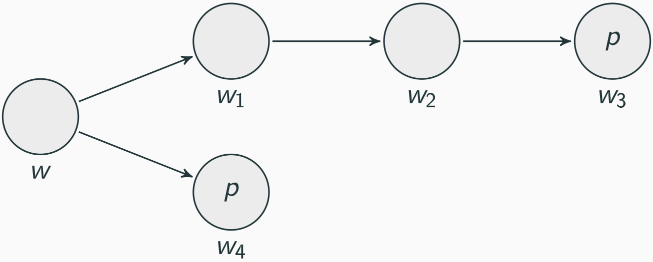
The *formulas* in the modal μ-calculus are generated by the grammar

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

- Formulas of the form $\mu x \; \varphi$ and $\nu x \; \varphi$ are called *fixpoint formulas* and interpreted as the least and greatest fixpoint of φ
- 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

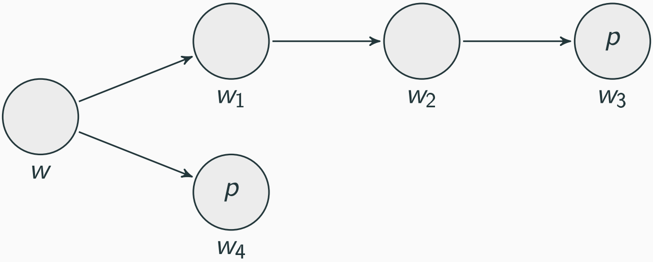
Example

Let $\mathcal{M} = (W, R, V)$ be the following Kripke model



Example

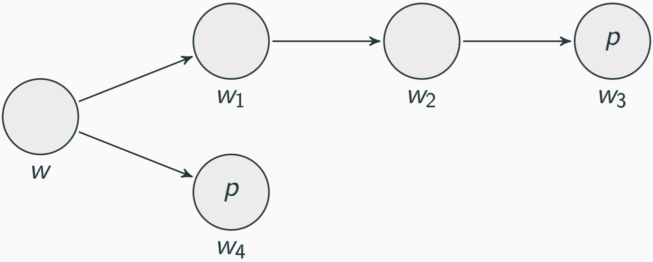
Let $\mathcal{M} = (W, R, V)$ be the following Kripke model



- $\mathcal{M}, w \models \Diamond p$

Example

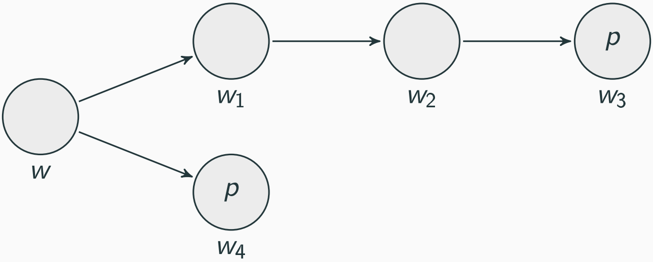
Let $\mathcal{M} = (W, R, V)$ be the following Kripke model



- $\mathcal{M}, w \models \Diamond p$
- $\mathcal{M}, w \not\models \Box p$

Example

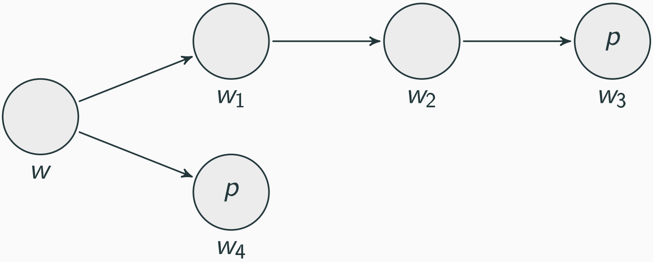
Let $\mathcal{M} = (W, R, V)$ be the following Kripke model



- $\mathcal{M}, w \models \Diamond p$
- $\mathcal{M}, w \not\models \Box p$
- $\mathcal{M}, w \models \mu x \, (p \vee \Box x)$

Example

Let $\mathcal{M} = (W, R, V)$ be the following Kripke model



- $\mathcal{M}, w \models \Diamond p$
- $\mathcal{M}, w \not\models \Box p$
- $\mathcal{M}, w \models \mu x\ (p \vee \Box x)$
- $\mathcal{M}, w \not\models \nu x\ \Diamond x$

Proof theory of the modal μ-calculus

- [Kozen '83] introduced finitary proof system with explicit induction rules

Proof theory of the modal μ-calculus

- [Kozen '83] introduced finitary proof system with explicit induction rules
- Completeness proven by [Walukiewicz '00]

Proof theory of the modal μ-calculus

- [Kozen '83] introduced finitary proof system with explicit induction rules
- Completeness proven by [Walukiewicz '00]
- [Niwiński, Walukiewicz '96] introduced infinitary tableaux games in which one player has winning strategy iff formula is valid

NW **pre-proofs**

An NW *pre-proof* is a, possibly infinite, tree defined from the following rules:

Ax1: $\dfrac{}{p, \bar{p}, \Gamma}$ Ax2: $\dfrac{}{\top, \Gamma}$ $\mathsf{R}_\vee$: $\dfrac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma}$ $\mathsf{R}_\wedge$: $\dfrac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$

$\mathsf{R}_\Box$: $\dfrac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma, \Delta}$ R_μ: $\dfrac{\varphi[\mu x.\varphi/x], \Gamma}{\mu x.\varphi, \Gamma}$ R_ν: $\dfrac{\varphi[\nu x.\varphi/x], \Gamma}{\nu x.\varphi, \Gamma}$

An NW *pre-proof* is a, possibly infinite, tree defined from the following rules:

Ax1: $\dfrac{}{p, \bar{p}, \Gamma}$ Ax2: $\dfrac{}{\top, \Gamma}$ $\mathsf{R}_\vee$: $\dfrac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma}$ $\mathsf{R}_\wedge$: $\dfrac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$

$\mathsf{R}_\Box$: $\dfrac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma, \Delta}$ R_μ: $\dfrac{\varphi[\mu x.\varphi/x], \Gamma}{\mu x.\varphi, \Gamma}$ R_ν: $\dfrac{\varphi[\nu x.\varphi/x], \Gamma}{\nu x.\varphi, \Gamma}$

- There are infinite branches
- But only finitely many sequents

Example NW pre-proof

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

Figure 1: NW pre-proof 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 such that φ_j is an immediate ancestor of φ_{j+1} for $j \in \omega$.

NW proofs

- A *trace* $(\varphi_j)_{j\in\omega}$ on an infinite branch is an infinite sequence of formulas such that 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 such that 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 pre-proof, where on every infinite branch there is a ν-trace.

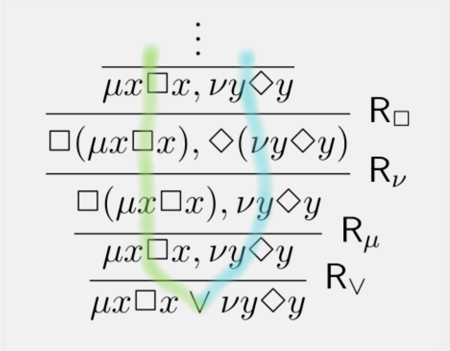


Figure 2: NW proof of $\mu x \Box x \vee \nu y \Diamond y$

ω-automata

Variation of finite state automaton which has infinite strings as inputs

ω-automata

Variation of finite state automaton which has infinite strings as inputs

Definition

Let Σ be a finite set, called an *alphabet*. A *non-deterministic automaton* over Σ is a quadruple $\mathbb{A} = \langle A, \Delta, a_I, \mathrm{Acc} \rangle$, where A is a finite set, $\Delta : A \times \Sigma \to \mathcal{P}(A)$ is the transition function of $\mathbb{A}$, $a_I \in A$ its initial state and $\mathrm{Acc} \subseteq A^\omega$ its acceptance condition.

Variation of finite state automaton which has infinite strings as inputs

Definition

Let Σ be a finite set, called an *alphabet*. A *non-deterministic automaton* over Σ is a quadruple $\mathbb{A} = \langle A, \Delta, a_I, \mathrm{Acc} \rangle$, where A is a finite set, $\Delta : A \times \Sigma \to \mathcal{P}(A)$ is the transition function of $\mathbb{A}$, $a_I \in A$ its initial state and $\mathrm{Acc} \subseteq A^\omega$ its acceptance condition.

- An automaton is called *deterministic* if for all pairs $(a, y) \in A \times \Sigma$ it holds $|\Delta(a, y)| = 1$.

Variation of finite state automaton which has infinite strings as inputs

Definition

Let Σ be a finite set, called an *alphabet*. A *non-deterministic automaton* over Σ is a quadruple $\mathbb{A} = \langle A, \Delta, a_I, \mathrm{Acc} \rangle$, where A is a finite set, $\Delta : A \times \Sigma \to \mathcal{P}(A)$ is the transition function of $\mathbb{A}$, $a_I \in A$ its initial state and $\mathrm{Acc} \subseteq A^\omega$ its acceptance condition.

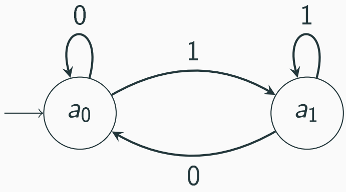
- An automaton is called *deterministic* if for all pairs $(a, y) \in A \times \Sigma$ it holds $|\Delta(a, y)| = 1$.
- A *run* of an automaton on a word $w = y_0 y_1 y_2 \ldots \in \Sigma^\omega$ is an infinite sequence $a_0 a_1 a_2 \ldots \in A^\omega$ such that $a_0 = a_I$ and $a_{i+1} \in \Delta(a_i, y_i)$ for all $i \in \omega$.

Variation of finite state automaton which has infinite strings as inputs

Definition
Let Σ be a finite set, called an *alphabet*. A *non-deterministic automaton* over Σ is a quadruple $\mathbb{A} = \langle A, \Delta, a_I, \mathrm{Acc} \rangle$, where A is a finite set, $\Delta : A \times \Sigma \to \mathcal{P}(A)$ is the transition function of $\mathbb{A}$, $a_I \in A$ its initial state and $\mathrm{Acc} \subseteq A^\omega$ its acceptance condition.

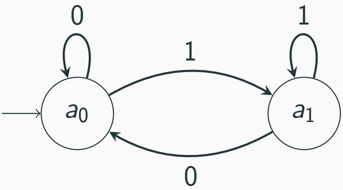
- An automaton is called *deterministic* if for all pairs $(a, y) \in A \times \Sigma$ it holds $|\Delta(a, y)| = 1$.
- A *run* of an automaton on a word $w = y_0 y_1 y_2 ... \in \Sigma^\omega$ is an infinite sequence $a_0 a_1 a_2 ... \in A^\omega$ such that $a_0 = a_I$ and $a_{i+1} \in \Delta(a_i, y_i)$ for all $i \in \omega$.
- A word w is *accepted* by $\mathbb{A}$ if there is a run of $\mathbb{A}$ on w in Acc.

Let $\Sigma = \{0, 1\}$ and $\mathbb{A} = \langle A, \Delta, a_I, \mathrm{Acc}\rangle$ be given as



Example ω-automata

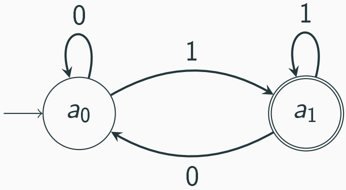
Let $\Sigma = \{0, 1\}$ and $\mathbb{A} = \langle A, \Delta, a_I, \mathrm{Acc}\rangle$ be given as



The acceptance condition can be given in different ways:

Büchi automata

Let $\Sigma = \{0, 1\}$ and $\mathbb{A} = \langle A, \Delta, a_0, \mathrm{Acc} \rangle$ be given as

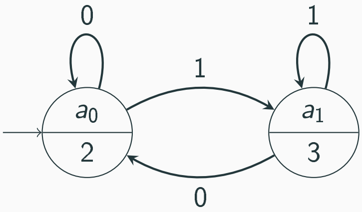


The acceptance condition can be given in different ways:

- A *Büchi* condition is given as a subset $F \subseteq A$. The corresponding acceptance condition is the set of runs, which contain infinitely many states in F.

Parity automata

Let $\Sigma = \{0, 1\}$ and $\mathbb{A} = \langle A, \Delta, a_0, \mathrm{Acc}\rangle$ be given as



The acceptance condition can be given in different ways:

- A *parity* condition is given as a map $\Omega : A \to \omega$. The corresponding acceptance condition is the set of runs α such that $\max\{\Omega(a) \mid a \text{ occurs infinitely often in } \alpha\}$ is even.

Recap NW proofs

An NW *pre-proof* is a, possibly infinite, tree defined from the following rules:

Ax1: $\dfrac{}{p, \bar{p}, \Gamma}$ Ax2: $\dfrac{}{\top, \Gamma}$ $\mathsf{R}_{\vee}$: $\dfrac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma}$ $\mathsf{R}_{\wedge}$: $\dfrac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma}$

$\mathsf{R}_{\Box}$: $\dfrac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma, \Delta}$ R_{μ}: $\dfrac{\varphi[\mu x.\varphi/x], \Gamma}{\mu x.\varphi, \Gamma}$ R_{ν}: $\dfrac{\varphi[\nu x.\varphi/x], \Gamma}{\nu x.\varphi, \Gamma}$

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

Tracking automaton

We can define nondeterministic parity automaton $\mathbb{A}$ s.t. for all infinite branches α in an NW pre-proof:

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

Tracking automaton

We can define nondeterministic parity automaton $\mathbb{A}$ s.t. for all infinite branches α in an NW pre-proof:

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

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 $\mathbb{A}^D$

Main advantage: Soundness condition based on branches instead of traces

Explicit determinisation

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

Explicit determinisation

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

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

Explicit determinisation

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

BT proof rules

Ax1: $\dfrac{}{p^\sigma, \bar{p}^\tau, \Gamma}$ Ax2: $\dfrac{}{\top^\sigma, \Gamma}$ 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 pre-proof, where on every infinite branch there is a successful string.

- Completeness and Soundness of BT^∞ proven by using determinisation method
- Advantage: Soundness condition on branches instead of traces

Example BT$^\infty$ proof

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\mu x \Box x^{0}, \nu y \Diamond y^{1}}}{\Box(\mu x \Box x)^{0}, \Diamond(\nu y \Diamond y)^{1}}\ \mathsf{R}_{\Box}}{\Box(\mu x \Box x)^{0}, \Diamond(\nu y \Diamond y)^{11}}\ \mathsf{Compress}^{11}}{\Box(\mu x \Box x)^{0}, \nu y \Diamond y^{1}}\ \mathsf{R}_{\nu}}{\mu x \Box x^{0}, \nu y \Diamond y^{1}}\ \mathsf{R}_{\mu}}{\Box(\mu x \Box x)^{0}, \Diamond(\nu y \Diamond y)^{1}}\ \mathsf{R}_{\Box}}{\Box(\mu x \Box x)^{\epsilon}, \nu y \Diamond y^{\epsilon}}\ \mathsf{R}_{\nu}}{\mu x \Box x^{\epsilon}, \nu y \Diamond y^{\epsilon}}\ \mathsf{R}_{\mu}}{\mu x \Box x \vee \nu y \Diamond y^{\epsilon}}\ \mathsf{R}_{\vee}
$$

BT proofs

- Only finitely many sequents on infinite branch
- Add discharge rule:

$$D^x: \dfrac{\begin{matrix}[\Gamma]^x \\ \vdots \\ \Gamma\end{matrix}}{\Gamma}$$

BT proofs

- Only finitely many sequents on infinite branch
- Add discharge rule:

$$D^x: \dfrac{\begin{matrix}[\Gamma]^x \\ \vdots \\ \Gamma\end{matrix}}{\Gamma}$$

- Get cyclic proof tree
- Infinite branches correspond to strongly connected components

BT proofs

- Only finitely many sequents on infinite branch
- Add discharge rule:

$$\begin{array}{c} [\Gamma]^x \\ \vdots \\ \mathsf{D}^x\colon \dfrac{\Gamma}{\Gamma} \end{array}$$

- Get cyclic proof tree
- Infinite branches correspond to strongly connected components

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

- Comparing to Jungteerapanich system: Trade-off between extra information and stronger soundness condition

Example BT proof

$$
\begin{array}{c l}
[\mu x \Box x^{0}, \nu y \Diamond y^{1}]^{\mathsf{x}} & \\
\hline
\Box(\mu x \Box x)^{0}, \Diamond(\nu y \Diamond y)^{1} & \mathsf{R}_{\Box} \\
\hline
\Box(\mu x \Box x)^{0}, \Diamond(\nu y \Diamond y)^{11} & \mathsf{Compress}^{11} \\
\hline
\Box(\mu x \Box x)^{0}, \nu y \Diamond y^{1} & \mathsf{R}_{\nu} \\
\hline
\mu x \Box x^{0}, \nu y \Diamond y^{1} & \mathsf{R}_{\mu} \\
\hline
\mu x \Box x^{0}, \nu y \Diamond y^{1} & \mathsf{D}^{\mathsf{x}} \\
\hline
\Box(\mu x \Box x)^{0}, \Diamond(\nu y \Diamond y)^{1} & \mathsf{R}_{\Box} \\
\hline
\Box(\mu x \Box x)^{\epsilon}, \nu y \Diamond y^{\epsilon} & \mathsf{R}_{\nu} \\
\hline
\mu x \Box x^{\epsilon}, \nu y \Diamond y^{\epsilon} & \mathsf{R}_{\mu} \\
\hline
\mu x \Box x \vee \nu y \Diamond y^{\epsilon} & \mathsf{R}_{\vee}
\end{array}
$$

Other logics

Same method could be applied to other logics:

Other logics

Same method could be applied to other logics:

- Alternation-free mu-calculus:
    - Weak co-Büchi automaton
    - Determinisation corresponds to Focus system

Other logics

Same method could be applied to other logics:

- Alternation-free mu-calculus:
  - Weak co-Büchi automaton
  - Determinisation corresponds to Focus system
- $\mathrm{FOL}_{\mathrm{ID}}$, Cyclic PA, etc...
  - Büchi automaton
  - Binary strings as annotations

Conclusion

- Introduced determinisation method for nondeterministic parity automata

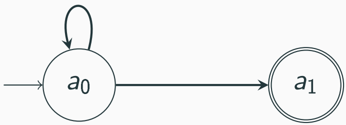
Conclusion

- Introduced determinisation method for nondeterministic parity automata
- Explicitly used this method to obtain proof system for the modal mu-calculus

Coffee !

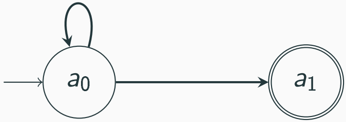
Example 1

Let $\mathbb{B}$ be the following nondeterministic Büchi automaton:

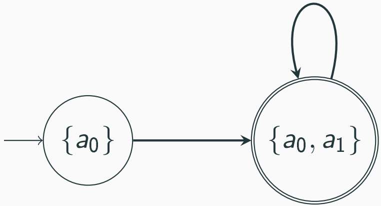


Example 1

Let $\mathbb{B}$ be the following nondeterministic Büchi automaton:

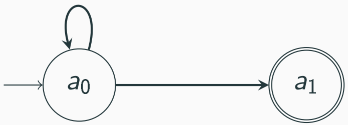


The subset construction yields the deterministic automaton $\mathbb{B}^S$

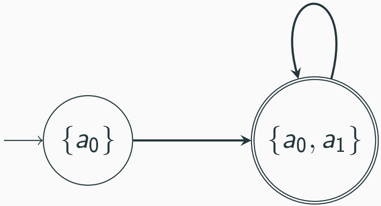


Example 1

Let $\mathbb{B}$ be the following nondeterministic Büchi automaton:



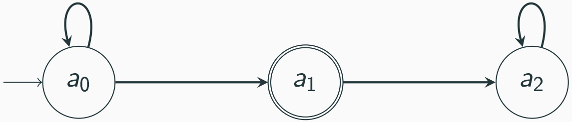
The subset construction yields the deterministic automaton $\mathbb{B}^S$



- Yet $\mathbb{B}^S$ is accepting and $\mathbb{B}$ is not!

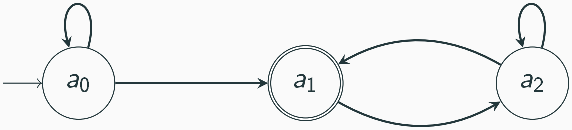
Example 2

Let $\mathbb{B}$ be the following nondeterministic Büchi automaton:



Example 3

Let $\mathbb{B}$ be the following nondeterministic Büchi automaton:



Thank you !