Modal Separation of Fixpoint Formulae

Jean Christoph Jung & Jędrzej Kołodziejski

Technical University of Dortmund

6 III 2025

Jena

Separators

Separators

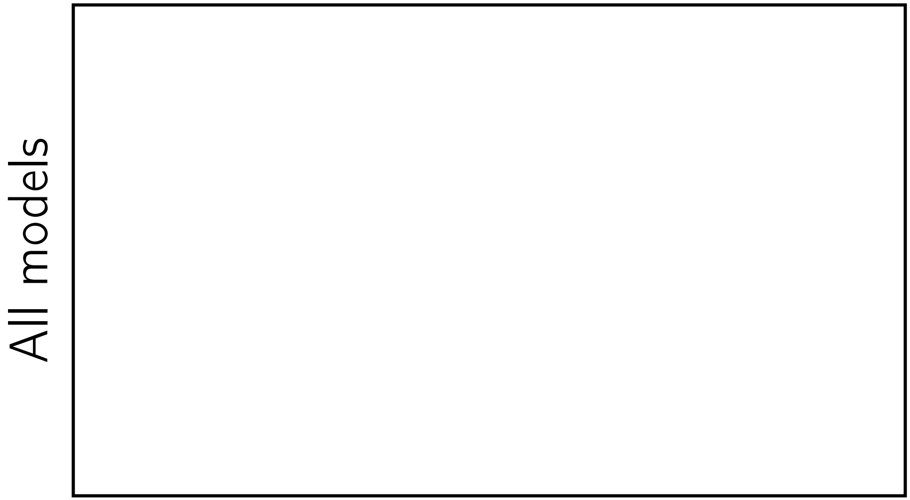
given mutually inconsistent $\varphi \models \neg\varphi'$

Separators

given mutually inconsistent $\varphi \models \neg\varphi'$

a **separator** is a formula ψ s.t. $\varphi \models \psi$ and $\psi \models \neg\varphi'$

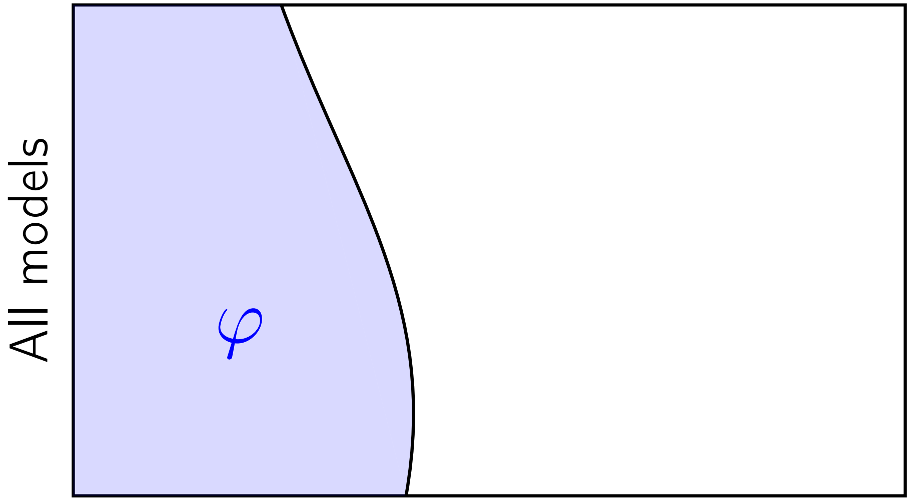
Separators



given mutually inconsistent $\varphi \models \neg\varphi'$

a **separator** is a formula ψ s.t. $\varphi \models \psi$ and $\psi \models \neg\varphi'$

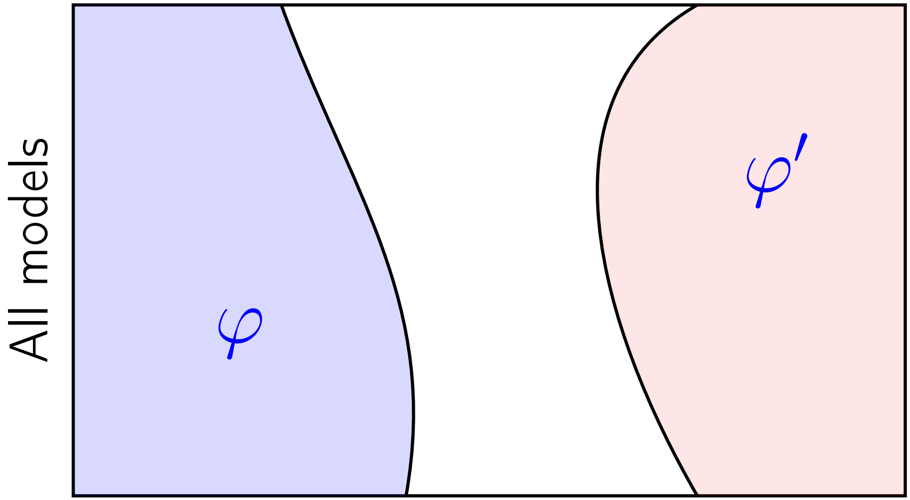
Separators



given mutually inconsistent $\varphi \models \neg\varphi'$

a **separator** is a formula ψ s.t. $\varphi \models \psi$ and $\psi \models \neg\varphi'$

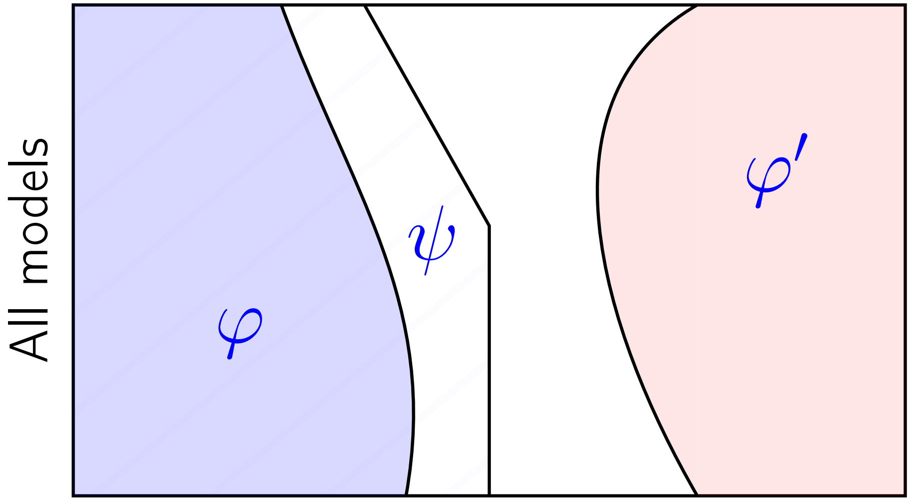
Separators



given mutually inconsistent $\varphi \models \neg\varphi'$

a **separator** is a formula ψ s.t. $\varphi \models \psi$ and $\psi \models \neg\varphi'$

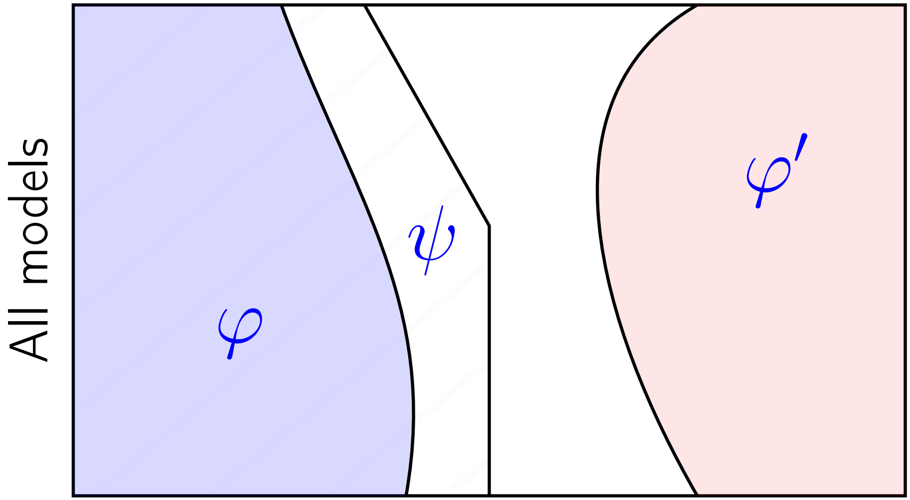
Separators



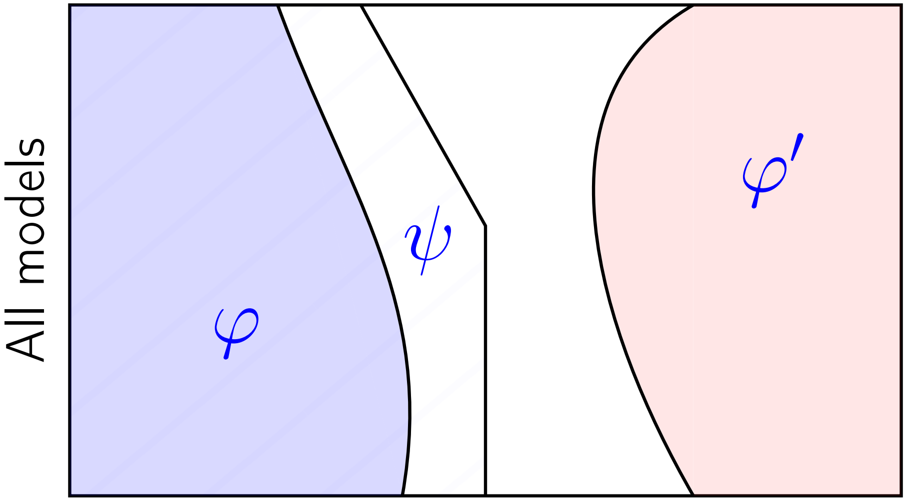
given mutually inconsistent $\varphi \models \neg\varphi'$

a **separator** is a formula ψ s.t. $\varphi \models \psi$ and $\psi \models \neg\varphi'$

Separators



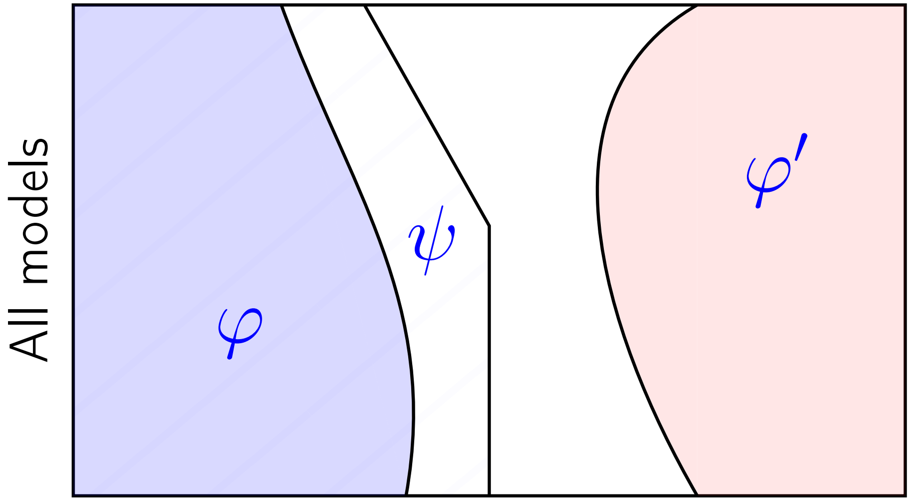
Separators



complicated

formulae $\varphi \models \neg\varphi'$

Separators

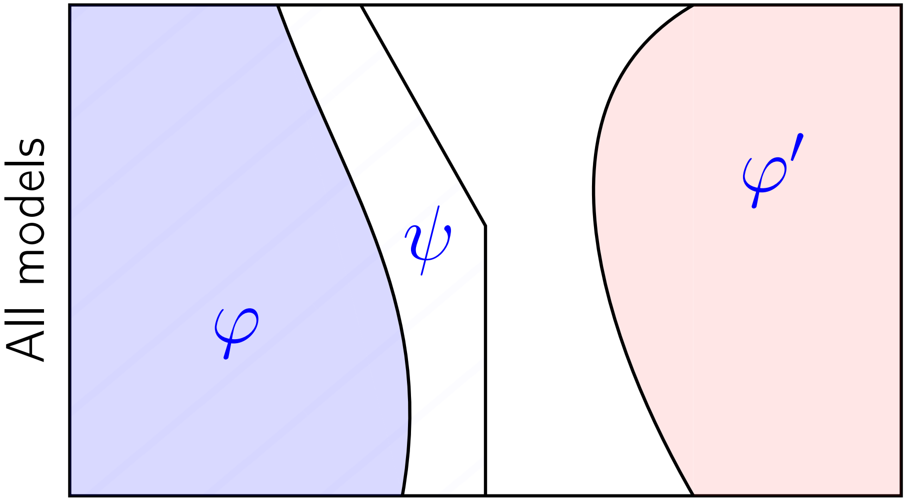


in expressive logic $\mathcal{L}^+$

complicated

formulae $\varphi \models \neg\varphi'$

Separators

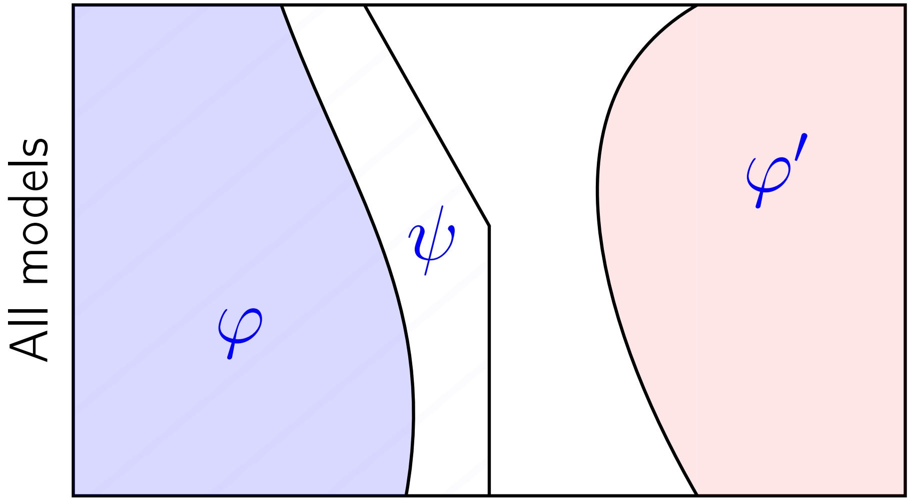


in expressive logic $\mathcal{L}^+$

complicated
formulae $\varphi \models \neg\varphi'$

simple ψ s.t.
$\varphi \models \psi \models \neg\varphi'$?

Separators



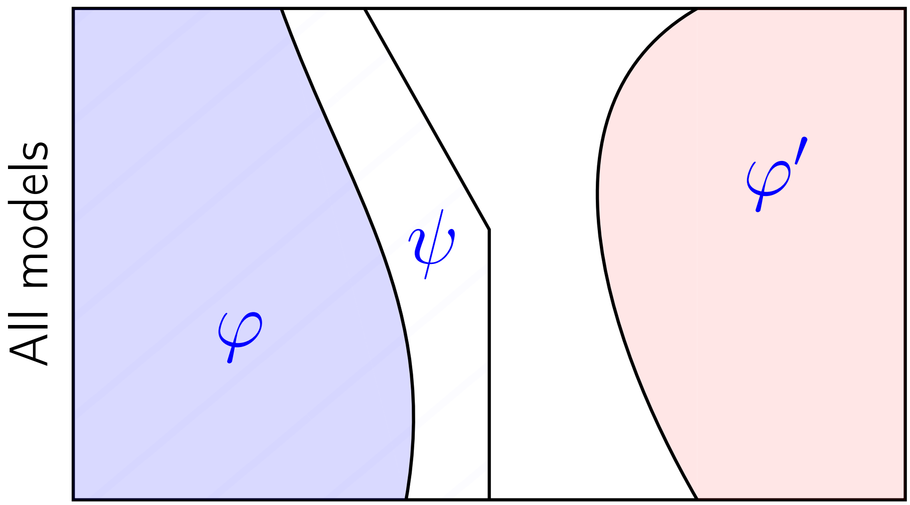
in expressive logic $\mathcal{L}^+$

complicated
formulae $\varphi \models \neg\varphi'$

in tamed logic $\mathcal{L} \subseteq \mathcal{L}^+$

simple ψ s.t.
$\varphi \models \psi \models \neg\varphi'$?

Separators



in expressive logic $\mathcal{L}^+$

complicated

formulae $\varphi \models \neg\varphi'$

simple explanation of contradiction

in tamed logic $\mathcal{L} \subseteq \mathcal{L}^+$

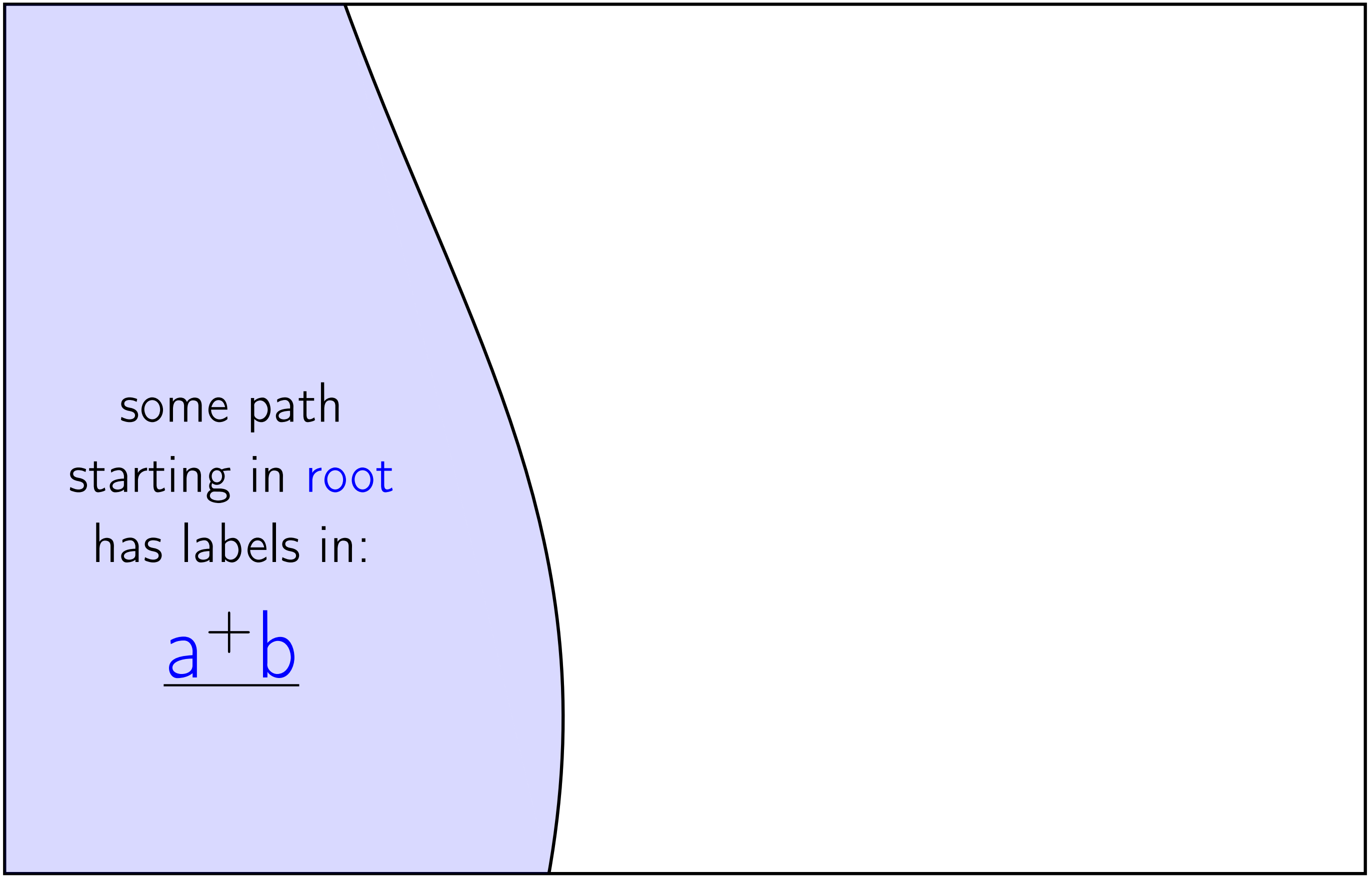
simple ψ s.t.

$\varphi \models \psi \models \neg\varphi'$?

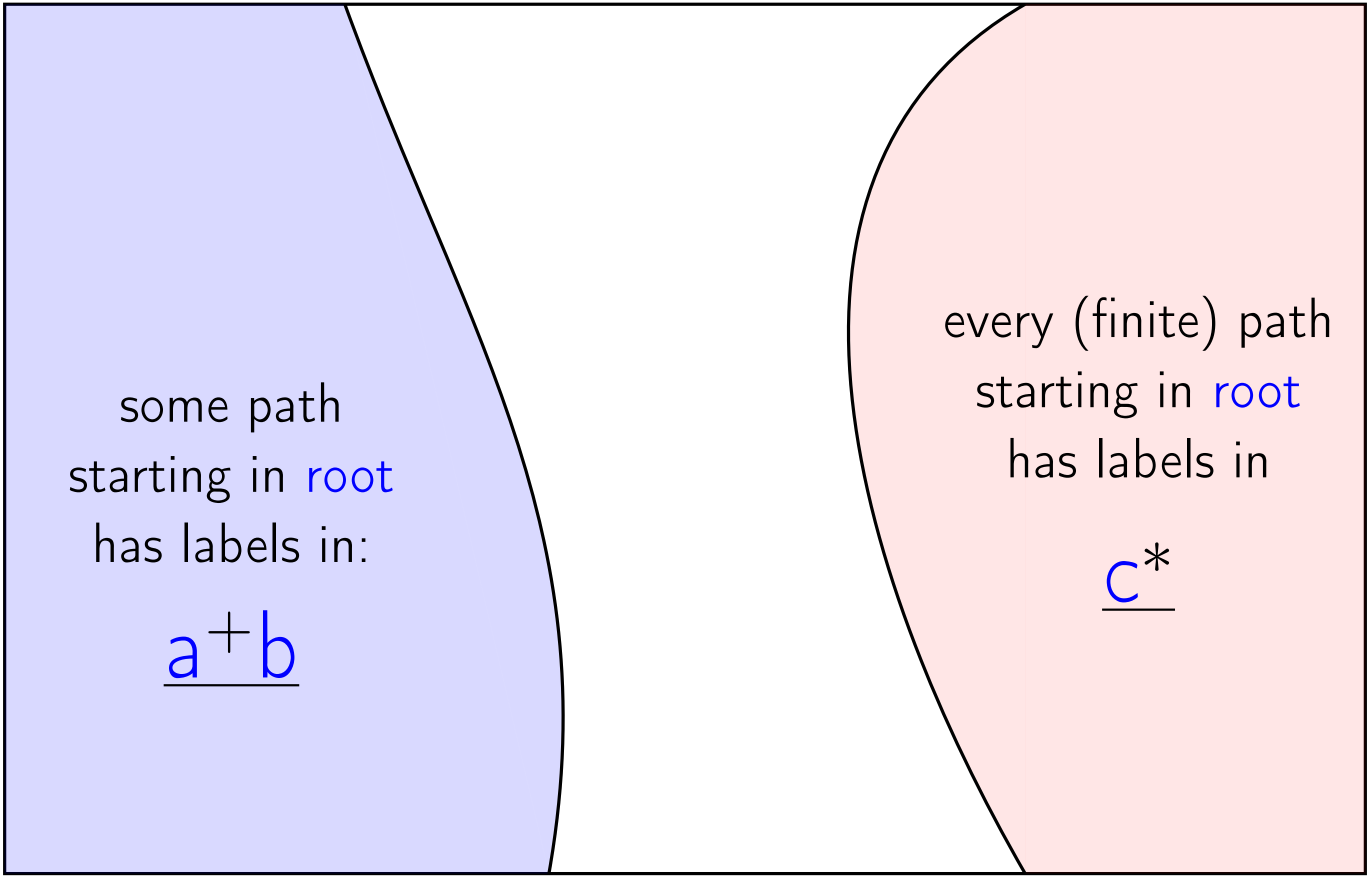
Example: labelled trees

Example: labelled trees

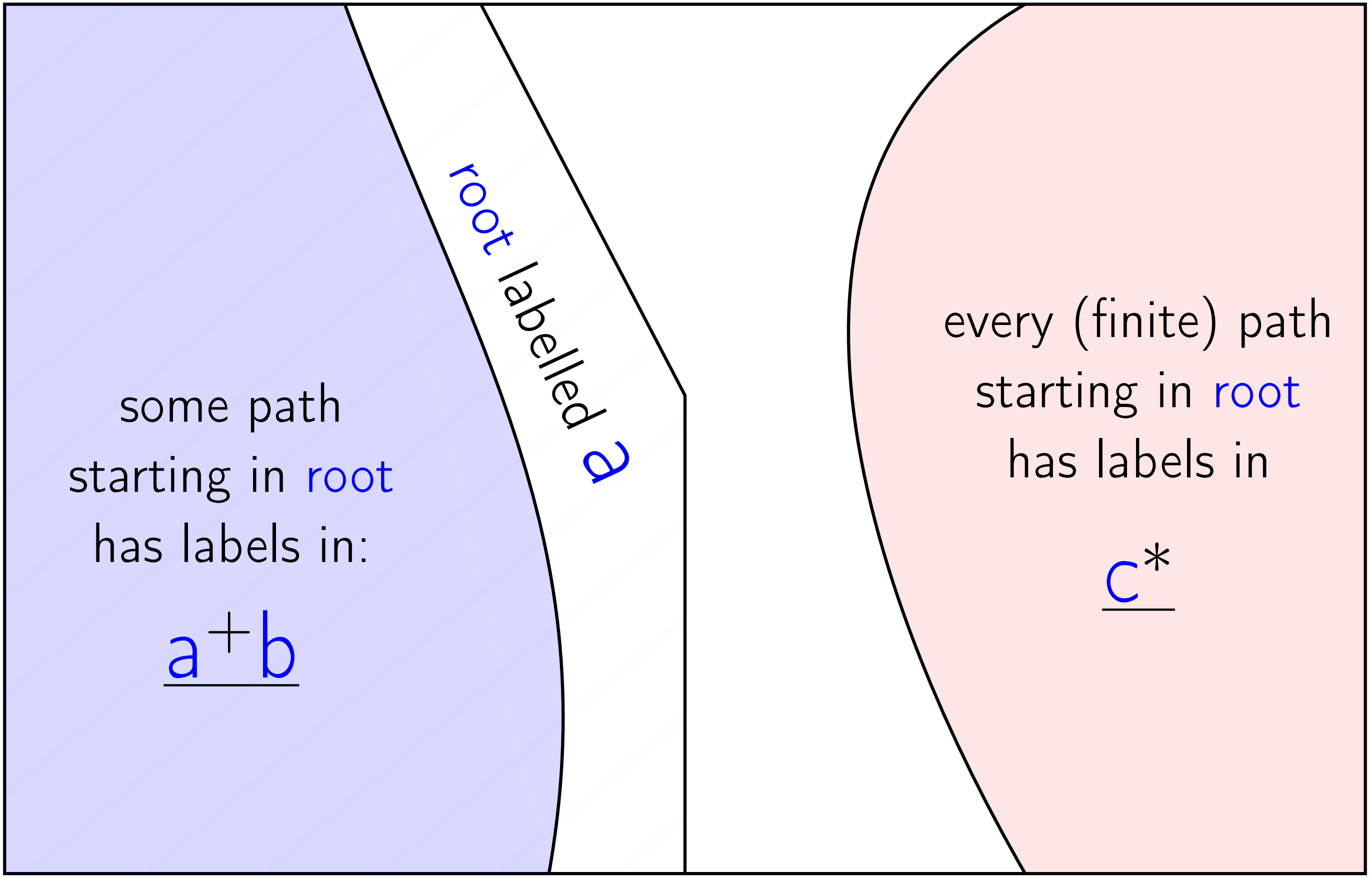
Example: labelled trees



Example: labelled trees



Example: labelled trees



Decision problem: $\mathcal{L}$-separability

Decision problem: $\mathcal{L}$-separability

given: $\varphi, \varphi' \in \boldsymbol{\mathcal{L}}^{+}$

Decision problem: $\mathcal{L}$-separability

given: $\varphi, \varphi' \in \boldsymbol{\mathcal{L}}^+$

is there a separator $\psi \in \boldsymbol{\mathcal{L}}$?

Separability generalizes definability

Separability generalizes definability

- For every formulae φ and ψ:

Separability generalizes definability

- For every formulae φ and ψ:

Separability generalizes definability

- For every formulae φ and ψ:

$$\psi \text{ separates } \varphi \text{ from } \neg\varphi$$

Separability generalizes definability

- For every formulae φ and ψ:

$$\psi \text{ separates } \varphi \text{ from } \neg\varphi$$

$$\Longleftrightarrow$$

$$\varphi \text{ and } \psi \text{ are equivalent.}$$

Separability generalizes definability

- For every formulae φ and ψ:

$$\psi \text{ separates } \varphi \text{ from } \neg\varphi$$

$$\Longleftrightarrow$$

$$\varphi \text{ and } \psi \text{ are equivalent.}$$

- Hence, $\mathcal{L}$-definability: "is given φ expressible in $\mathcal{L}$?"

Separability generalizes definability

- For every formulae φ and ψ:

ψ separates φ from $\neg\varphi$

$$\Longleftrightarrow$$

φ and ψ are equivalent.

- Hence, $\mathcal{L}$-definability: "is given φ expressible in $\mathcal{L}$?"
- is a special case of $\mathcal{L}$-separability.

The logics $\mathcal{L}$ and $\mathcal{L}^+$

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$$\mathcal{L} = \text{modal logic ML}$$

syntax:

$$a \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi$$

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

$$a \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi$$

semantics

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

$$\boldsymbol{a} \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi$$

semantics

formulae interpreted in
points of labelled
directed graphs

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

$$\boldsymbol{a} \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi$$

atomic
propositions a $\in$ At

semantics

formulae interpreted in
points of labelled
directed graphs

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

$$a \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi$$

atomic propositions a $\in$ At

φ true in some child

semantics

formulae interpreted in points of labelled directed graphs

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

$$a \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi$$

atomic propositions $a \in$ At

φ true in some child

semantics

formulae interpreted in points of labelled directed graphs

$\mathcal{L}^+ = \mu$-ML = ML + **fixpoints**

The logics $\mathcal{L}$ and $\mathcal{L}^+$

$\mathcal{L}$ = modal logic ML

syntax:

$$a \mid \neg\varphi \mid \varphi \vee \psi \mid \Diamond\varphi \mid x \mid \mu x.\varphi$$

atomic propositions $a \in At$

φ true in some child

semantics

formulae interpreted in points of labelled directed graphs

$\mathcal{L}^+ = \mu$-ML = ML + **fixpoints**

The semantics of μ-ML = ML + fixpoints

The semantics of μ-ML = ML + fixpoints

$$\mu\text{-ML}$$

$$=$$

Automata

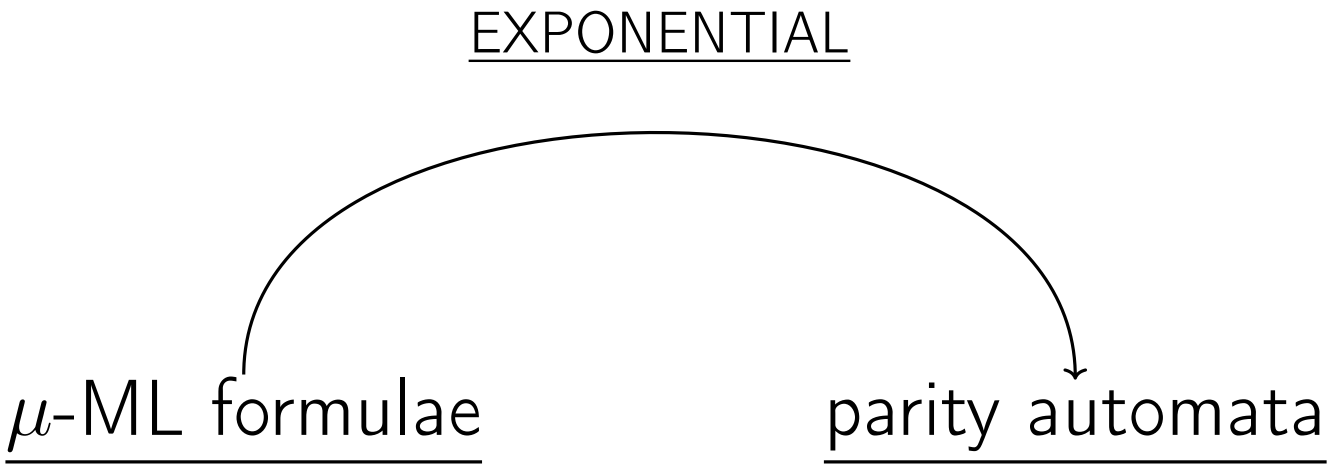
Translations

Translations

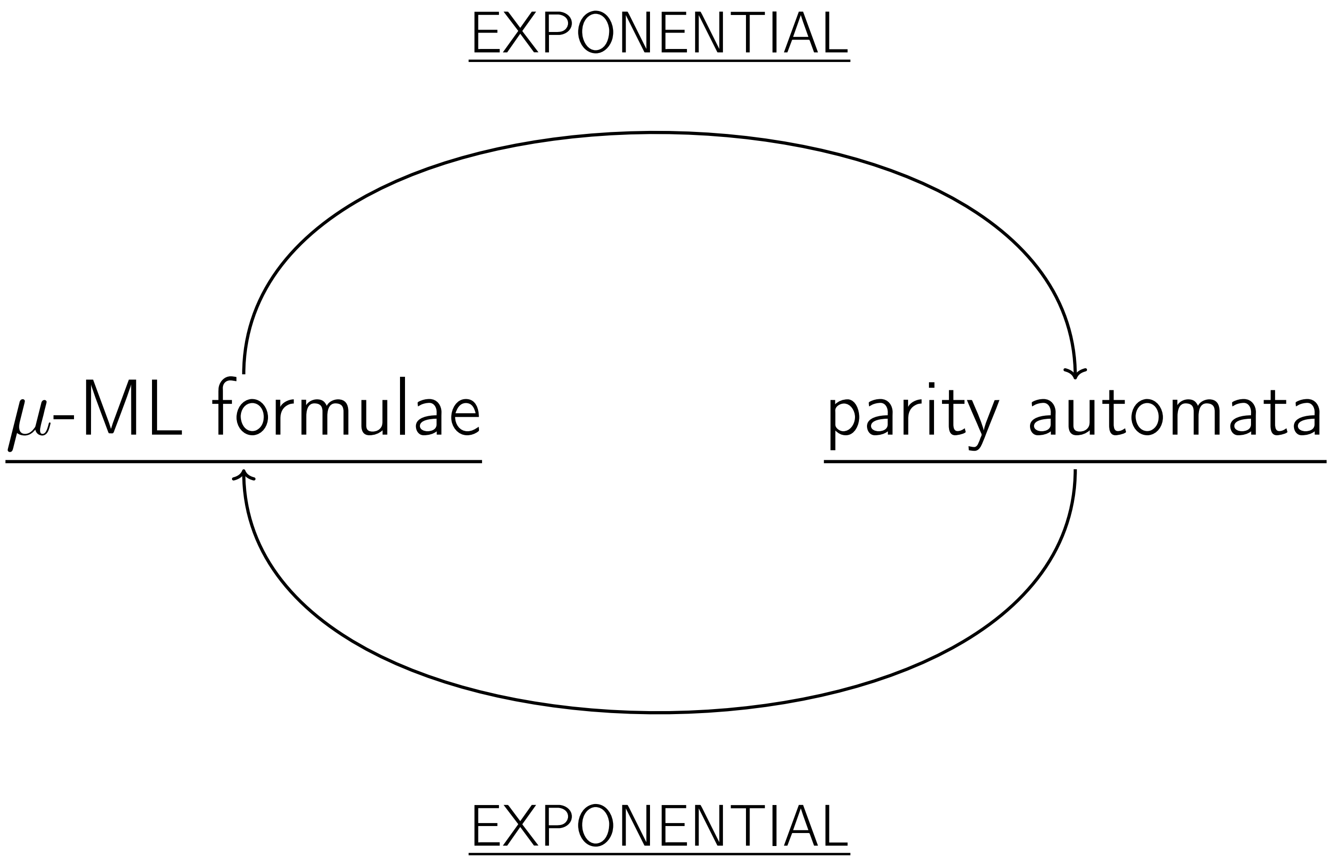
μ-ML formulae

parity automata

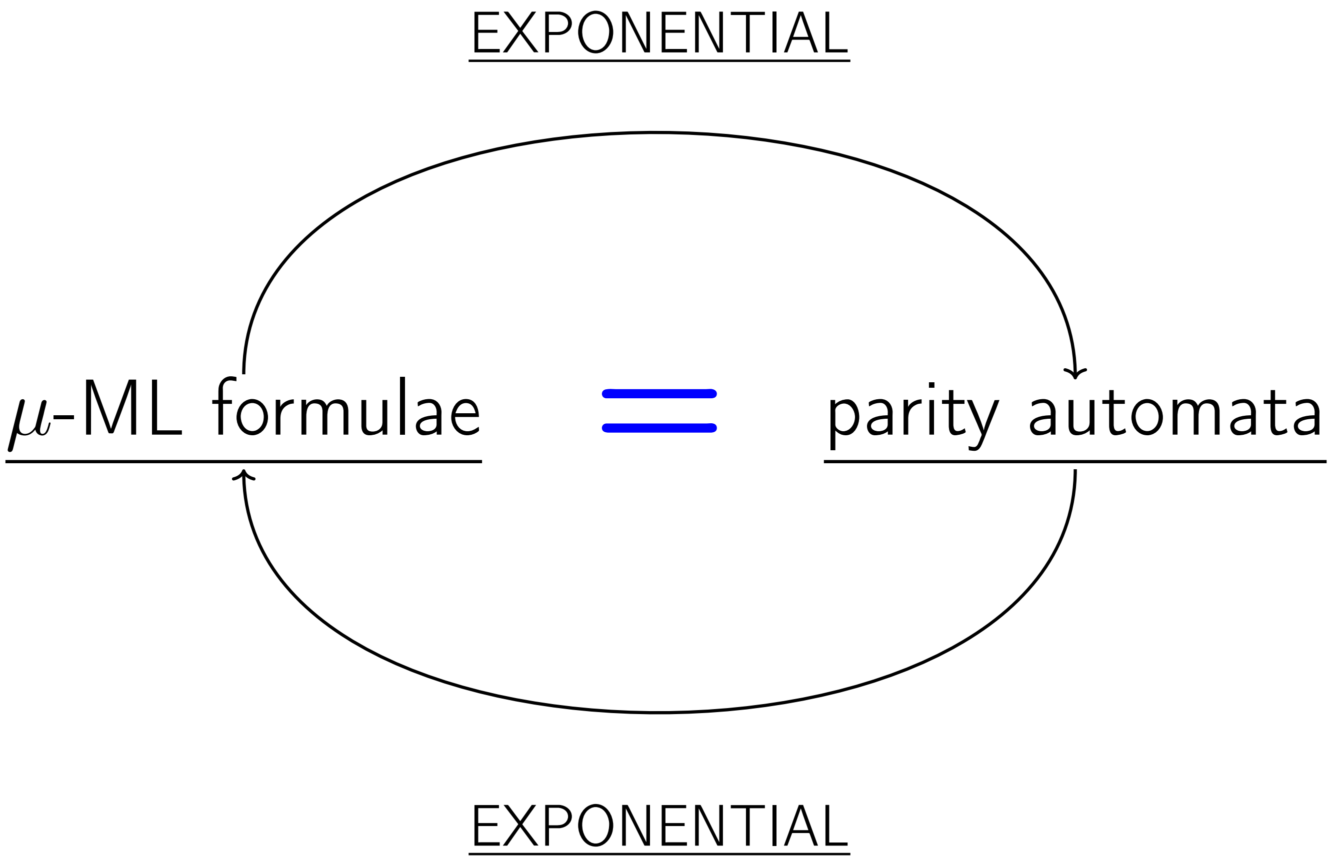
Translations



Translations



Translations



The question: modal separability

The question: modal separability

- Given contradictory φ and φ' in μ-ML...

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

The question: modal separability

- Given contradictory φ and φ' in μ-ML…
- … **is there** a separator ψ in ML? Can it be **computed**?

Example

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.a \wedge \Diamond(b \vee x)$$

"some path has labels from a^+b"

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.\mathsf{a} \wedge \Diamond(\mathsf{b} \vee x)$$

"some path has labels from $\mathsf{a}^+\mathsf{b}$"

$$\varphi' = \nu y.\mathsf{c} \wedge \Box y$$

"all (finite) paths belong to c^*"

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.\mathsf{a} \wedge \Diamond(\mathsf{b} \vee x)$$

"some path has labels from $\mathsf{a}^+\mathsf{b}$"

$$\psi = \mathsf{a}$$

"root satisfies a"

$$\varphi' = \nu y.\mathsf{c} \wedge \Box y$$

"all (finite) paths belong to c^*"

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.\mathsf{a} \wedge \Diamond(\mathsf{b} \vee x)$$

"some path has labels from $\mathsf{a}^+\mathsf{b}$"

$$\psi = \mathsf{a}$$

"root satisfies a"

$$\varphi' = \nu y.\mathsf{c} \wedge \Box y$$

"all (finite) paths belong to c^*"

Non-example

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.a \wedge \Diamond(b \vee x)$$

"some path has labels from a^+b"

$$\psi = a$$

"root satisfies a"

$$\varphi' = \nu y.c \wedge \Box y$$

"all (finite) paths belong to c^*"

Non-example

$$\varphi = \varphi_{WF} = \mu x.\Box x$$

"no infinite paths"

The question: modal separability

- Given contradictory φ and φ' in μ-ML…
- … **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.\mathsf{a} \wedge \Diamond(\mathsf{b} \vee x)$$

"some path has labels from $\mathsf{a}^{+}\mathsf{b}$"

$$\psi = \mathsf{a}$$

"root satisfies a"

$$\varphi' = \nu y.\mathsf{c} \wedge \Box y$$

"all (finite) paths belong to c^{*}"

Non-example

$$\varphi = \varphi_{\mathsf{WF}} = \mu x.\Box x$$

"no infinite paths"

$$\varphi' = \neg\varphi_{\mathsf{WF}}$$

"there is an infinite path"

The question: modal separability

- Given contradictory φ and φ' in μ-ML...
- ... **is there** a separator ψ in ML? Can it be **computed**?

Example

$$\varphi = \mu x.\mathsf{a} \wedge \Diamond(\mathsf{b} \vee x)$$

"some path has labels from $\mathsf{a}^+\mathsf{b}$"

$$\psi = \mathsf{a}$$

"root satisfies a"

$$\varphi' = \nu y.\mathsf{c} \wedge \Box y$$

"all (finite) paths belong to c^*"

Non-example

$$\varphi = \varphi_{\mathsf{WF}} = \mu x.\Box x$$

"no infinite paths"

φ entails no modal formulae!

$$\varphi' = \neg\varphi_{\mathsf{WF}}$$

"there is an infinite path"

The answer:

The answer:

	all models	words	binary trees	d-ary trees for $d \geq 3$
ML-definability	ExpTime	PSpace	ExpTime	ExpTime
ML-separability	ExpTime	PSpace	ExpTime	2-ExpTime
separator construction	double exp.	single exp.	double exp.	triple exp.
interpolant existence for modal logic	always	always	always	coNExpTime

The answer:

	all models	words	binary trees	d-ary trees for $d \geq 3$
ML-definability	ExpTime	PSpace	ExpTime	ExpTime
ML-separability	ExpTime	PSpace	ExpTime	2-ExpTime
separator construction	double exp.	single exp.	double exp.	triple exp.
interpolant existence for modal logic	always	always	always	coNExpTime

- all the complexity results are *completeness* results.

The answer:

	all models	words	binary trees	d-ary trees for $d \geq 3$
ML-definability	ExpTime	PSpace	ExpTime	ExpTime
ML-separability	ExpTime	PSpace	ExpTime	2-ExpTime
separator construction	double exp.	single exp.	double exp.	triple exp.
interpolant existence for modal logic	always	always	always	coNExpTime

- all the complexity results are *completeness* results.
- *words* mean *unary trees*: words with successor relation, no order.

The answer:

	all models	words	binary trees	d-ary trees for $d \geq 3$
ML-definability	ExpTime	PSpace	ExpTime	ExpTime
ML-separability	ExpTime	PSpace	ExpTime	2-ExpTime
separator construction	double exp.	single exp.	double exp.	triple exp.
interpolant existence for modal logic	always	always	always	coNExpTime

- all the complexity results are *completeness* results.
- *words* mean *unary trees*: words with successor relation, no order.
- in all cases trees are unordered.

What's hot:

What's hot:

- ML-separability is 2-ExpTime-complete over ternary trees...

What's hot:

- ML-separability is 2-ExpTime-complete over ternary trees...
- ...but only ExpTime-complete over binary trees.

What's hot:

- ML-separability is 2-ExpTime-complete over ternary trees...
- ...but only ExpTime-complete over binary trees.
- Craig interpolants (type of separators) for ML always exist over binary trees...

What's hot:

- ML-separability is 2-ExpTime-complete over ternary trees...
- ...but only ExpTime-complete over binary trees.
- Craig interpolants (type of separators) for ML always exist over binary trees...
- ...but over ternary trees deciding its existence is coNExpTime-complete.

What's hot:

- ML-separability is 2-ExpTime-complete over ternary trees...
- ...but only ExpTime-complete over binary trees.
- Craig interpolants (type of separators) for ML always exist over binary trees...
- ...but over ternary trees deciding its existence is coNExpTime-complete.

Ternary (and higher arity) trees

are harder than the binary ones!

What's hot:

- ML-separability is 2-ExpTime-complete over ternary trees...
- ...but only ExpTime-complete over binary trees.
- Craig interpolants (type of separators) for ML always exist over binary trees...
- ...but over ternary trees deciding its existence is coNExpTime-complete.

!!! Ternary (and higher arity) trees are harder than the binary ones! !!!

Behind separability

Behind separability

no modal separator for φ and φ'

Behind separability

no modal separator for φ and φ'

$$\Longleftrightarrow$$

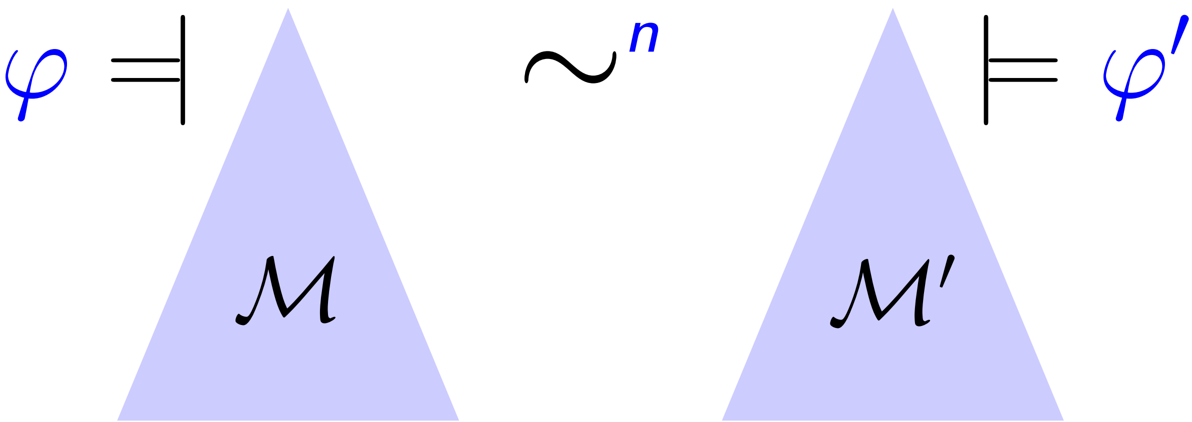
for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^{n} \mathcal{M}' \models \varphi'$

Behind separability

no modal separator for φ and φ'

$$\iff$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

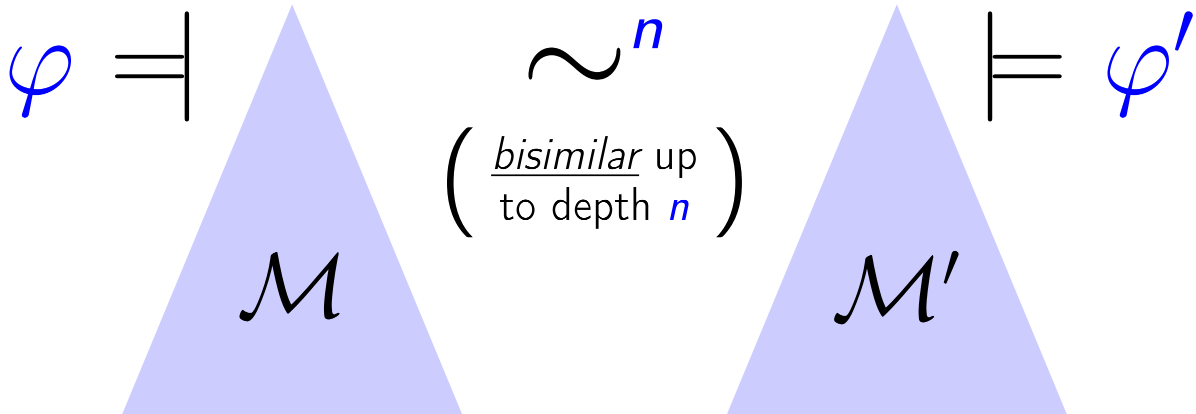


Behind separability

no modal separator for φ and φ'

$$\Longleftrightarrow$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$



Behind separability

no modal separator for φ and φ'

$$\Longleftrightarrow$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

$$\overset{?}{\Longleftrightarrow}$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \cong^n \mathcal{M}' \models \varphi'$

Behind separability

no modal separator for φ and φ'

$$\iff$$

bisimilar up to depth n

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

$$\overset{?}{\iff}$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \cong^n \mathcal{M}' \models \varphi'$

Behind separability

no modal separator for φ and φ'

$$\iff$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^{n} \mathcal{M}' \models \varphi'$

bisimilar up to depth n

$$\overset{?}{\iff}$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \cong^{n} \mathcal{M}' \models \varphi'$

isomorphic up to depth n

Behind separability

no modal separator for φ and φ'

bisimilar up to depth n

$$\Longleftrightarrow$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

✓ all models

$$\overset{?}{\Longleftrightarrow}$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \cong^n \mathcal{M}' \models \varphi'$

isomorphic up to depth n

Behind separability

no modal separator for φ and φ'

$$\Longleftrightarrow$$

bisimilar up to depth n

for every $n \in \mathbb{N}$ there are: $\varphi \Dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

✓ all models

✓ finite trees

$$\overset{?}{\Longleftrightarrow}$$

for every $n \in \mathbb{N}$ there are: $\varphi \Dashv \mathcal{M} \cong^n \mathcal{M}' \models \varphi'$

isomorphic up to depth n

Behind separability

no modal separator for φ and φ'

$$\Longleftrightarrow$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

bisimilar up to depth n

✓ all models

✓ finite trees

✓ binary trees

$$\overset{?}{\Longleftrightarrow}$$

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \cong^n \mathcal{M}' \models \varphi'$

isomorphic up to depth n

Behind separability

no modal separator for φ and φ'

$\Longleftrightarrow$

bisimilar up to depth n

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \sim^n \mathcal{M}' \models \varphi'$

✓ all models

✓ finite trees

✓ binary trees

$\overset{?}{\Longleftrightarrow}$

✗ ternary trees

for every $n \in \mathbb{N}$ there are: $\varphi \dashv \mathcal{M} \cong^n \mathcal{M}' \models \varphi'$

isomorphic up to depth n

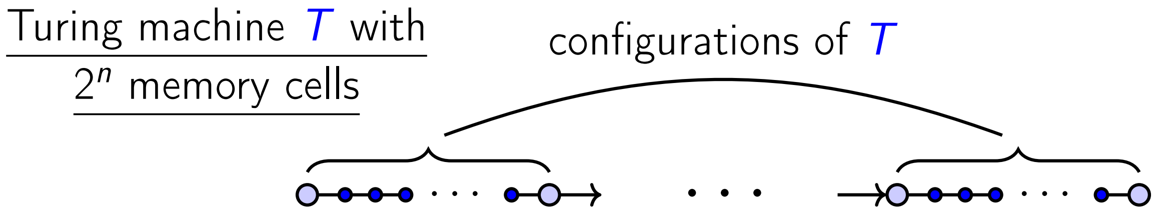
Ternary case: lower bound

Ternary case: lower bound

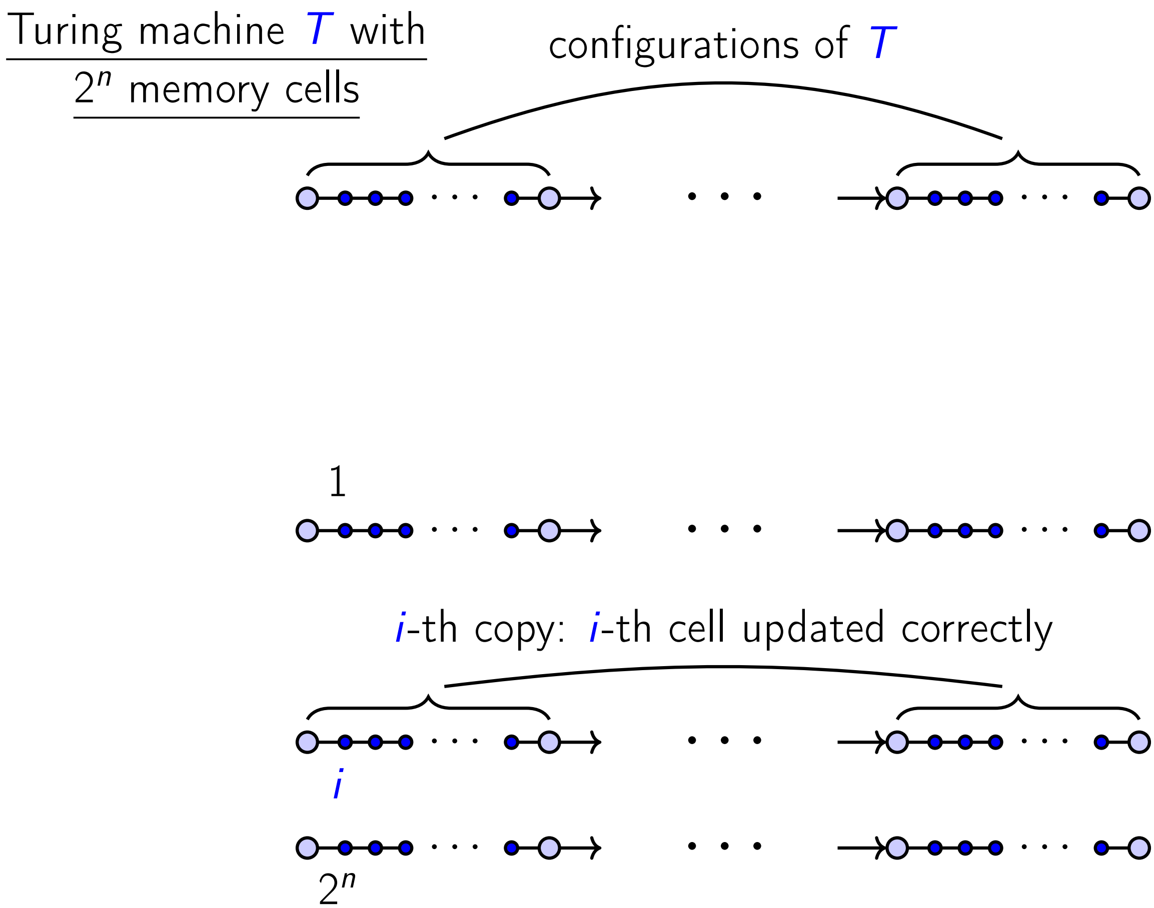
Turing machine T with

2^n memory cells

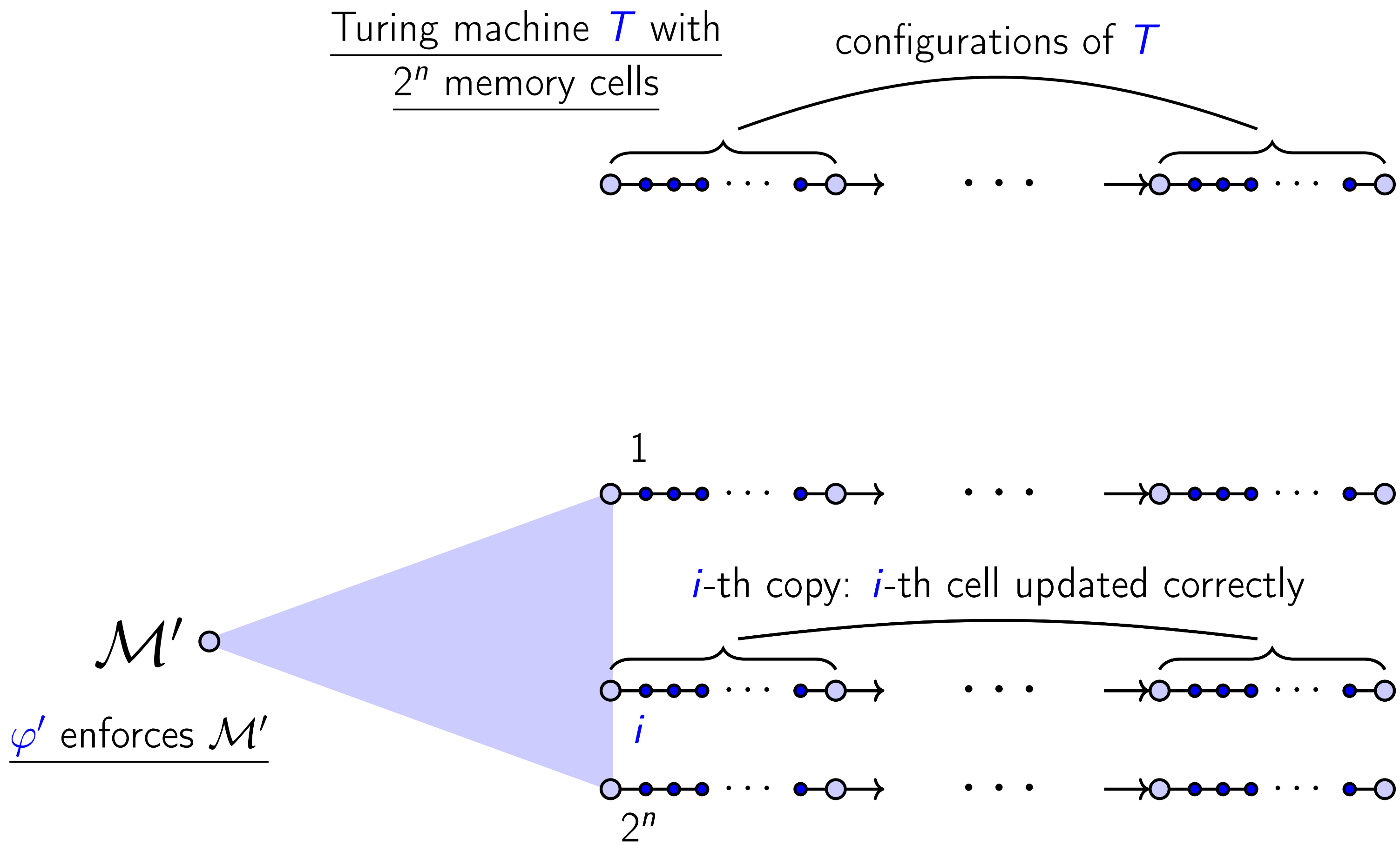
Ternary case: lower bound



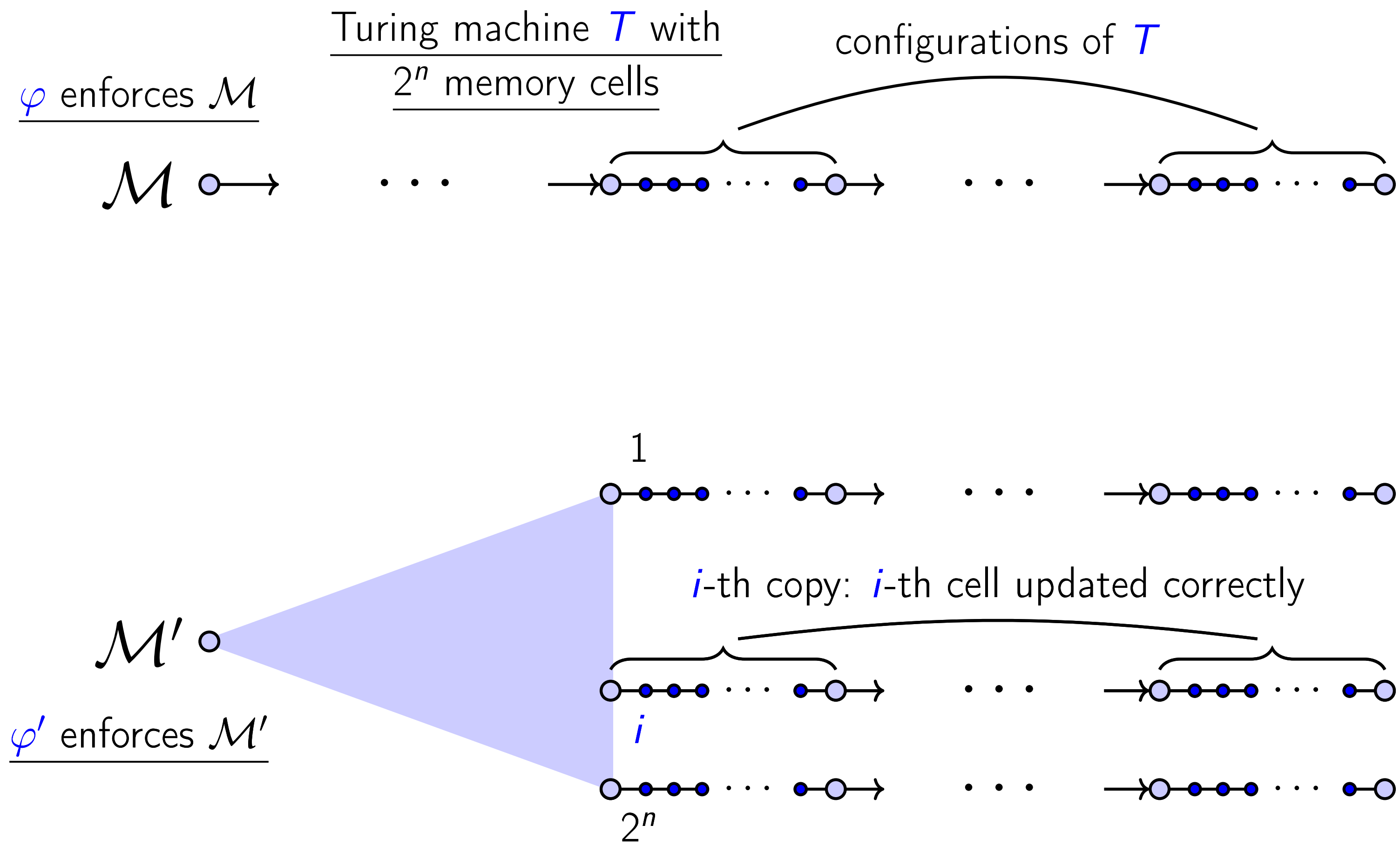
Ternary case: lower bound



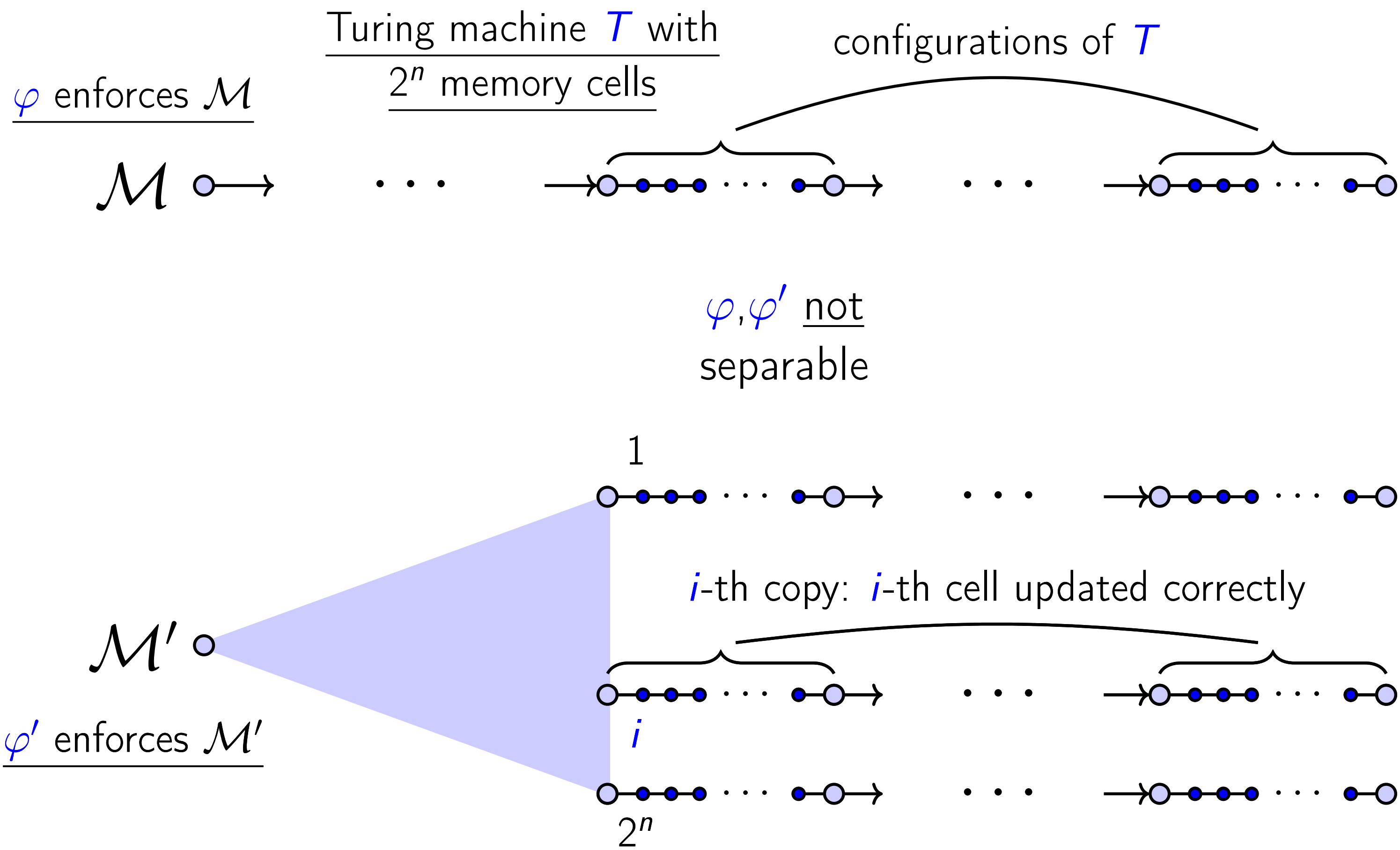
Ternary case: lower bound



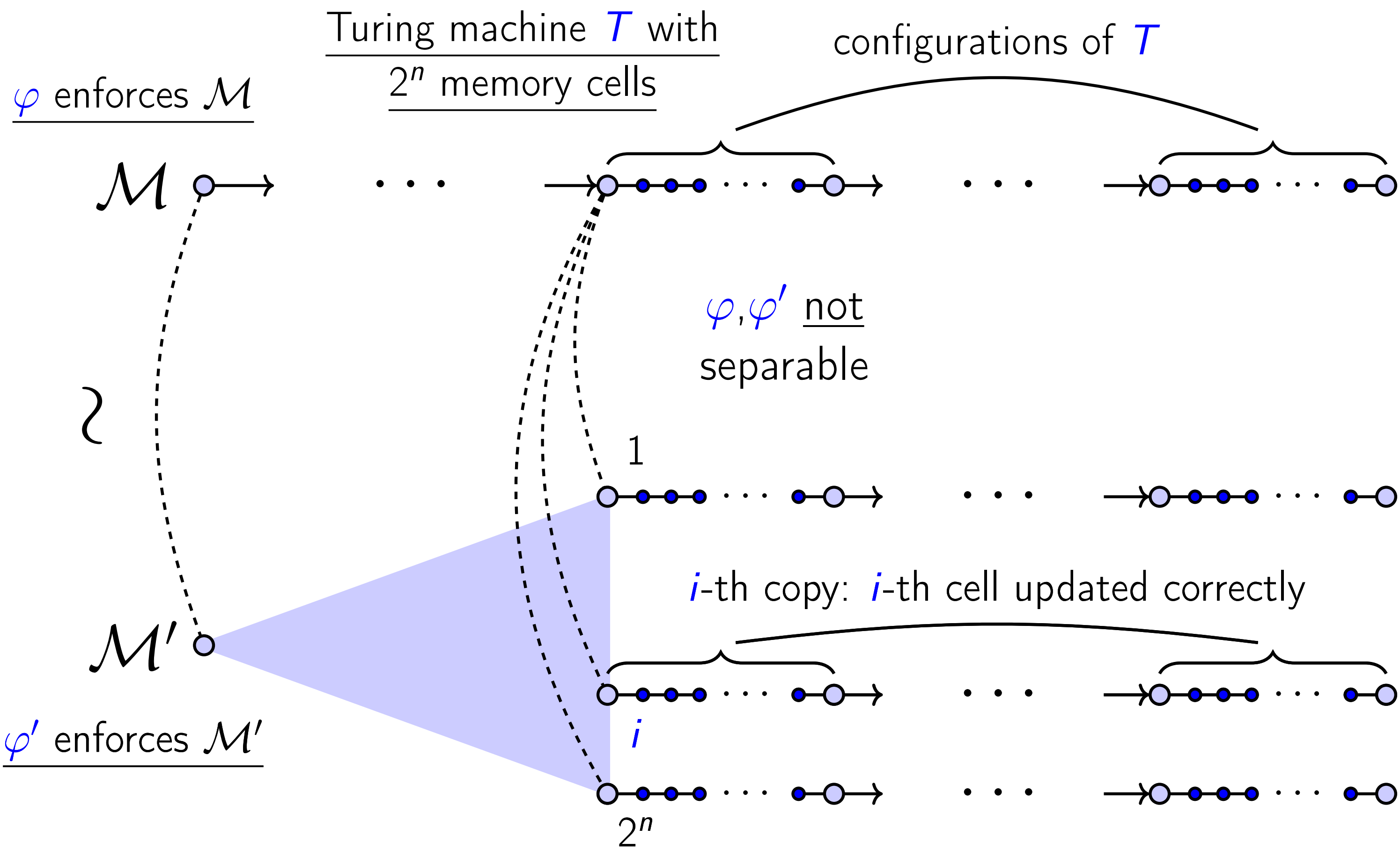
Ternary case: lower bound



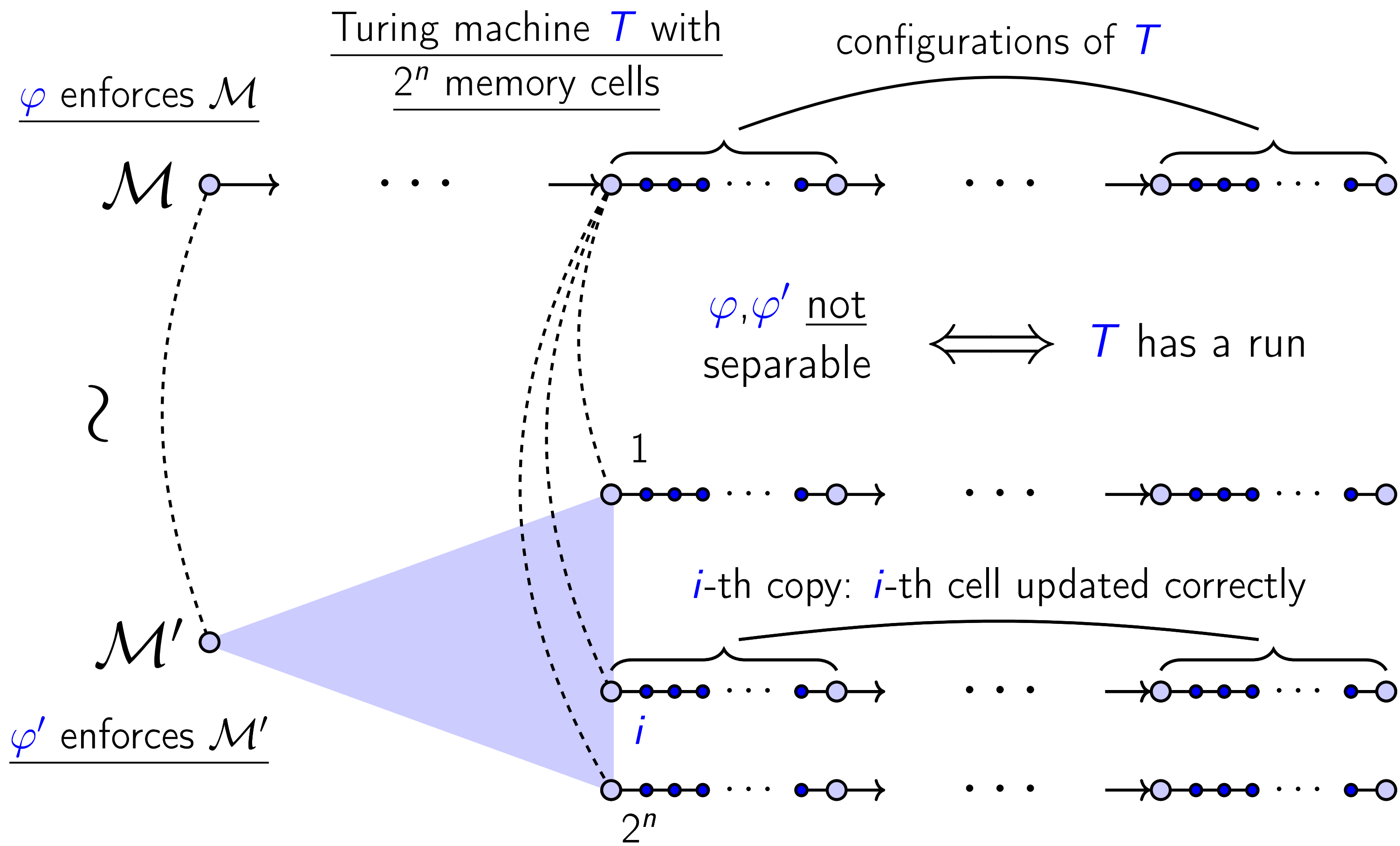
Ternary case: lower bound



Ternary case: lower bound



Ternary case: lower bound



Ternary case: lower bound

Ternary case: lower bound

- for a given ExpSpace Turing machine T we construct φ,φ' such that:

Ternary case: lower bound

- for a given ExpSpace Turing machine T we construct φ,φ' such that:

φ,φ' not separable $\iff$ T has a run

Ternary case: lower bound

- for a given ExpSpace Turing machine T we construct φ, φ' such that:

φ, φ' not separable $\iff$ T has a run

- idea implemented using gadgets possible over ternary, but not binary trees

Ternary case: lower bound

- for a given ExpSpace Turing machine T we construct φ,φ' such that:

$$\varphi,\varphi' \text{ \underline{not} separable} \iff T \text{ has a run}$$

- idea implemented using gadgets possible over ternary, but not binary trees
- with more effort: alternating ExpSpace machines

Ternary case: lower bound

- for a given ExpSpace Turing machine T we construct φ,φ' such that:

$$\varphi,\varphi' \text{ \underline{not} separable} \iff T \text{ has a run}$$

- idea implemented using gadgets possible over ternary, but not binary trees
- with more effort: alternating ExpSpace machines
- conclusion: modal separation is 2-ExpTime-hard over ternary trees!

Ternary case: upper bound

Ternary case: upper bound

- assume: tree $\mathcal{M} = (V, E)$, every node with at most ternary branching

Ternary case: upper bound

- assume: tree $\mathcal{M} = (V, E)$, every node with at most ternary branching
- and a nondeterministic parity automaton $\mathcal{A} = (Q, \delta, q_I, \text{rank})$

Ternary case: upper bound

- assume: tree $\mathcal{M} = (V, E)$, every node with at most ternary branching
- and a nondeterministic parity automaton $\mathcal{A} = (Q, \delta, q_I, \text{rank})$
- we define a game $\mathcal{G}(\mathcal{M}, \mathcal{A})$ played between ∃ve and ∀dam such that:

Ternary case: upper bound

- assume: tree $\mathcal{M} = (V, E)$, every node with at most ternary branching
- and a nondeterministic parity automaton $\mathcal{A} = (Q, \delta, q_I, \mathrm{rank})$
- we define a game $\mathcal{G}(\mathcal{M}, \mathcal{A})$ played between $\exists$ve and $\forall$dam such that:

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient* of some ternary $\mathcal{N} \models \mathcal{A}$

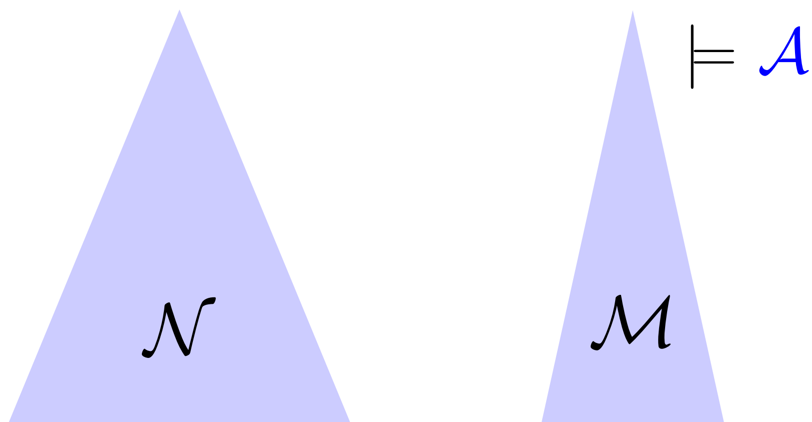
Ternary case: upper bound

- assume: tree $\mathcal{M} = (V, E)$, every node with at most ternary branching
- and a nondeterministic parity automaton $\mathcal{A} = (Q, \delta, q_I, \text{rank})$
- we define a game $\mathcal{G}(\mathcal{M}, \mathcal{A})$ played between $\exists$ve and $\forall$dam such that:

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\iff$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$



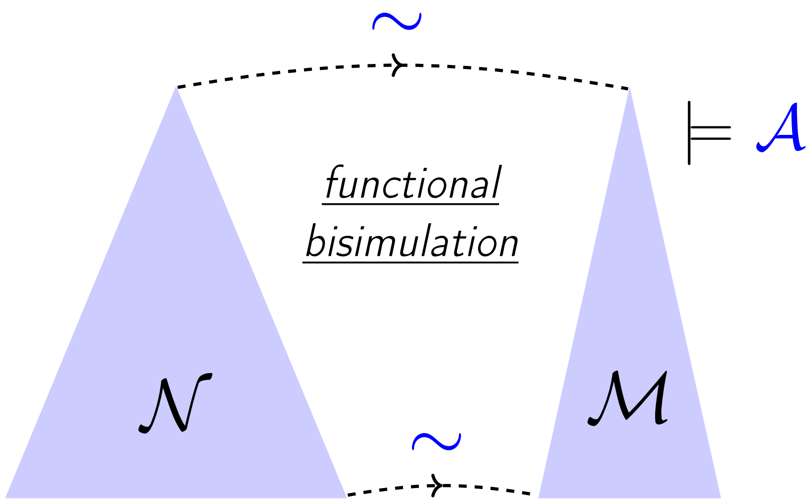
Ternary case: upper bound

- assume: tree $\mathcal{M} = (V, E)$, every node with at most ternary branching
- and a nondeterministic parity automaton $\mathcal{A} = (Q, \delta, q_I, \text{rank})$
- we define a game $\mathcal{G}(\mathcal{M}, \mathcal{A})$ played between $\exists$ve and $\forall$dam such that:

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$\iff$

$\mathcal{M}$ is a *bisimulation quotient* of some ternary $\mathcal{N} \models \mathcal{A}$



Ternary case: upper bound

Ternary case: upper bound

- for $\mathcal{M} = (V, E)$ and $\mathcal{A} = (Q, \delta, q_I, \mathrm{rank})$:

Ternary case: upper bound

- for $\mathcal{M} = (V, E)$ and $\mathcal{A} = (Q, \delta, q_I, \mathrm{rank})$:

$$\exists\text{ve wins } \mathcal{G}(\mathcal{M}, \mathcal{A})$$

$$\iff$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

Ternary case: upper bound

- for $\mathcal{M} = (V, E)$ and $\mathcal{A} = (Q, \delta, q_I, \text{rank})$:

$$\exists\text{ve wins } \mathcal{G}(\mathcal{M}, \mathcal{A})$$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- positions: $V \times Q$

Ternary case: upper bound

- for $\mathcal{M} = (V, E)$ and $\mathcal{A} = (Q, \delta, q_I, \text{rank})$:

$$\exists\text{ve wins } \mathcal{G}(\mathcal{M}, \mathcal{A})$$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- positions: $V \times Q$
- from (v,q) $\exists$ve chooses a transition $\{q_1,q_2,q_3\} = D \in \delta(q, \text{color}(v))$
- and a surjective map $h : D \to W$ where W is the set of children of v.

Ternary case: upper bound

- for $\mathcal{M} = (V, E)$ and $\mathcal{A} = (Q, \delta, q_I, \text{rank})$:

$$\exists\text{ve wins } \mathcal{G}(\mathcal{M}, \mathcal{A})$$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- positions: $V \times Q$
- from (v,q) ∃ve chooses a transition $\{q_1,q_2,q_3\} = D \in \delta(q, \text{color}(v))$
- and a surjective map $h : D \to W$ where W is the set of children of v.
- ∀dam responds with a choice of $q_i \in D$
- the next round starts in $(h(q_i),q_i)$.

Ternary case: upper bound

- for $\mathcal{M} = (V, E)$ and $\mathcal{A} = (Q, \delta, q_I, \text{rank})$:

$$\exists\text{ve wins } \mathcal{G}(\mathcal{M}, \mathcal{A})$$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- positions: $V \times Q$
- from (v, q) ∃ve chooses a transition $\{q_1, q_2, q_3\} = D \in \delta(q, \text{color}(v))$
- and a surjective map $h : D \to W$ where W is the set of children of v.
- ∀dam responds with a choice of $q_i \in D$
- the next round starts in $(h(q_i), q_i)$.
- Parity game: ranks inherited from $\mathcal{A}$.

Ternary case: upper bound

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

$\mathcal{B}$ accepts $\mathcal{M}$

$$\iff$$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\iff$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

$\mathcal{B}$ accepts $\mathcal{M}$

$$\Longleftrightarrow$$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- ...and use it to decide separation.

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

$\mathcal{B}$ accepts $\mathcal{M}$

$$\Longleftrightarrow$$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- ...and use it to decide separation.

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

$\mathcal{B}$ accepts $\mathcal{M}$

$$\iff$$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\iff$$

$\mathcal{M}$ is a *bisimulation quotient*
of some ternary $\mathcal{N} \models \mathcal{A}$

- ...and use it to decide separation.

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

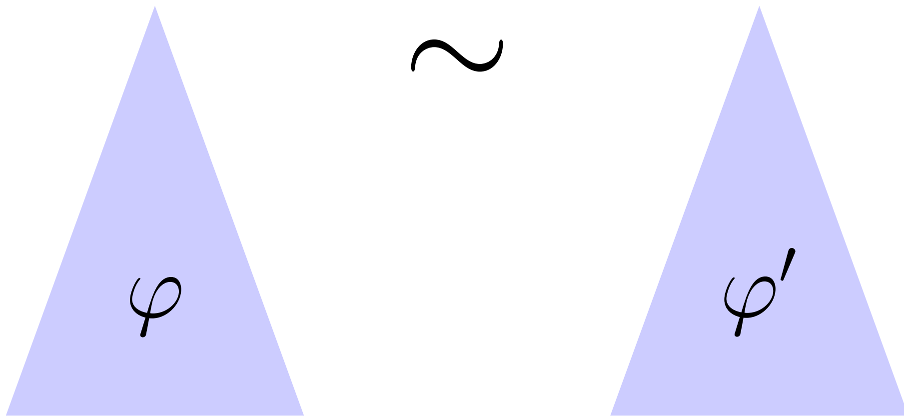
$\mathcal{B}$ accepts $\mathcal{M}$

$$\Longleftrightarrow$$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\Longleftrightarrow$$

$\mathcal{M}$ is a *bisimulation quotient* of some ternary $\mathcal{N} \models \mathcal{A}$



- ...and use it to decide separation.

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

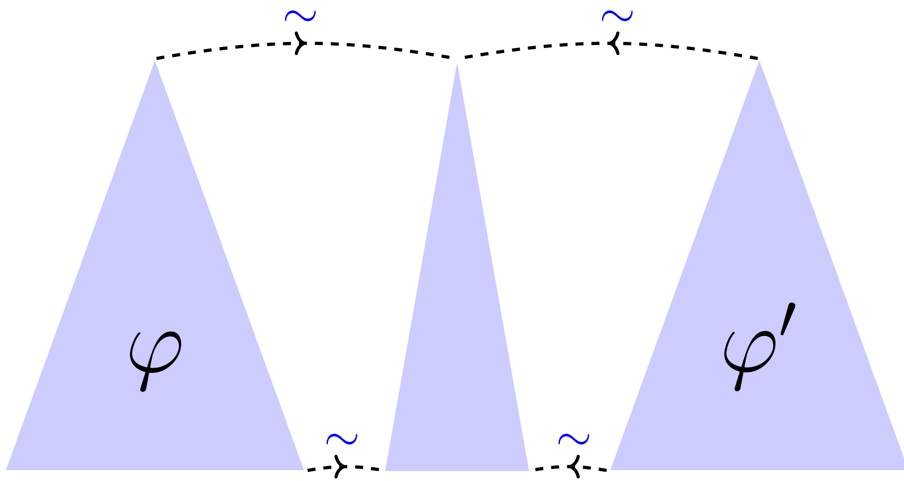
$\mathcal{B}$ accepts $\mathcal{M}$

$$\iff$$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$$\iff$$

$\mathcal{M}$ is a *bisimulation quotient* of some ternary $\mathcal{N} \models \mathcal{A}$



- ...and use it to decide separation.

Ternary case: upper bound

- for $\mathcal{A}$ we construct exponentially-sized $\mathcal{B}$ such that for all $\mathcal{M}$:

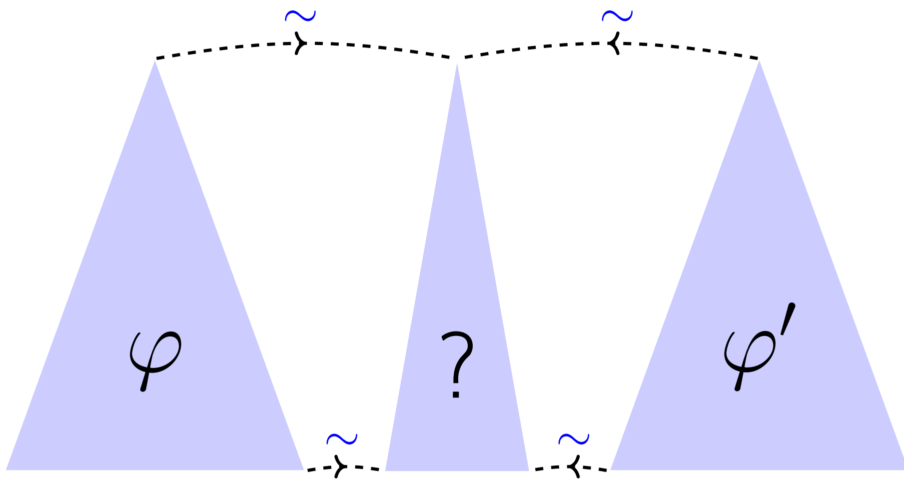
$\mathcal{B}$ accepts $\mathcal{M}$

$\iff$

$\exists$ve wins $\mathcal{G}(\mathcal{M}, \mathcal{A})$

$\iff$

$\mathcal{M}$ is a *bisimulation quotient* of some ternary $\mathcal{N} \models \mathcal{A}$



- ...and use it to decide separation.

Thank you!

Relativization

Relativization

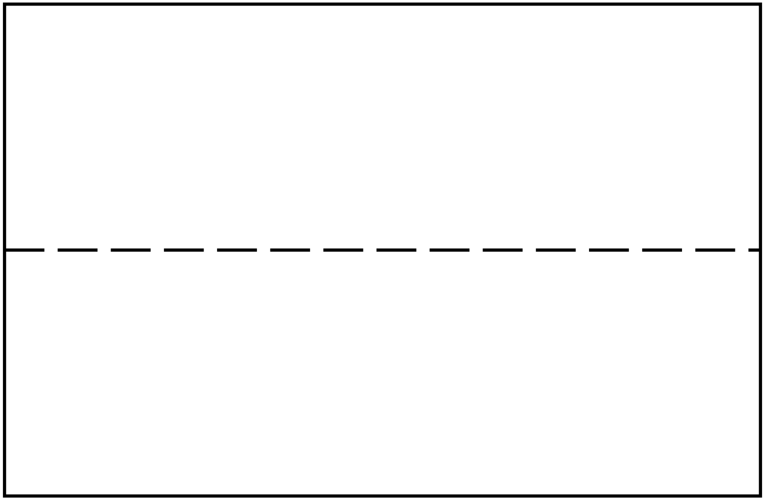
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.

Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.

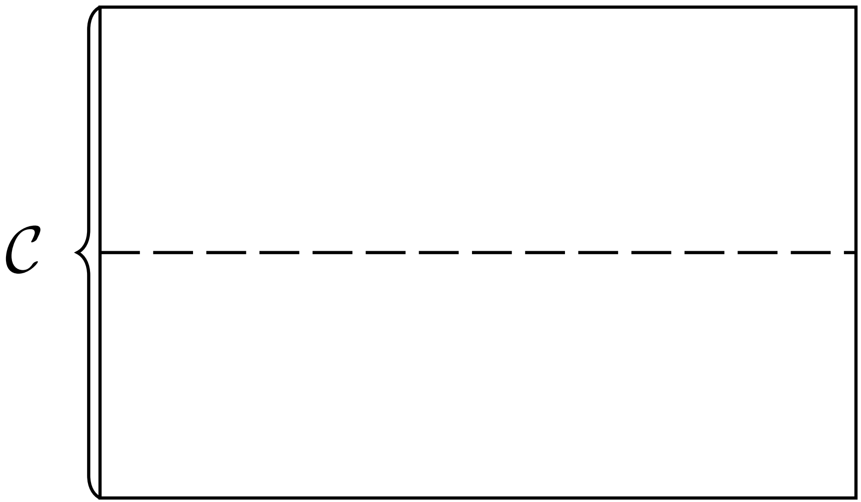
Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\quad \mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



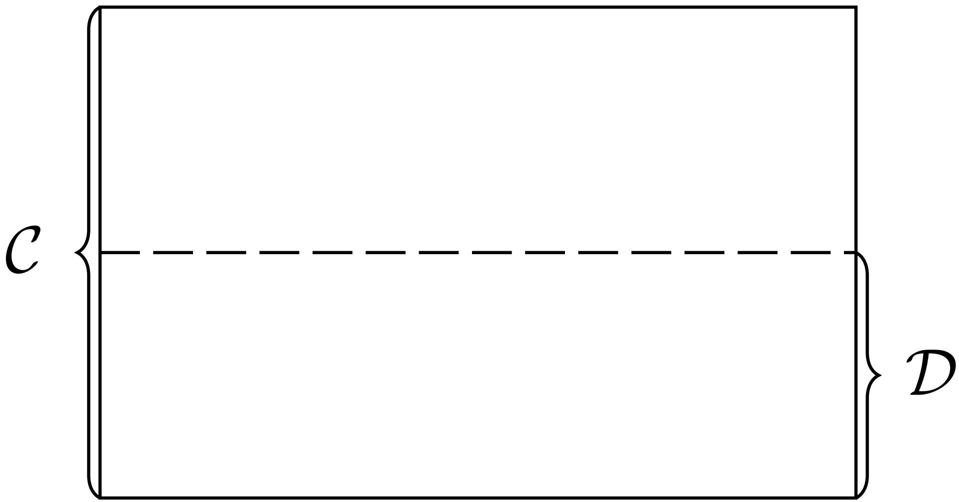
Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



Relativization

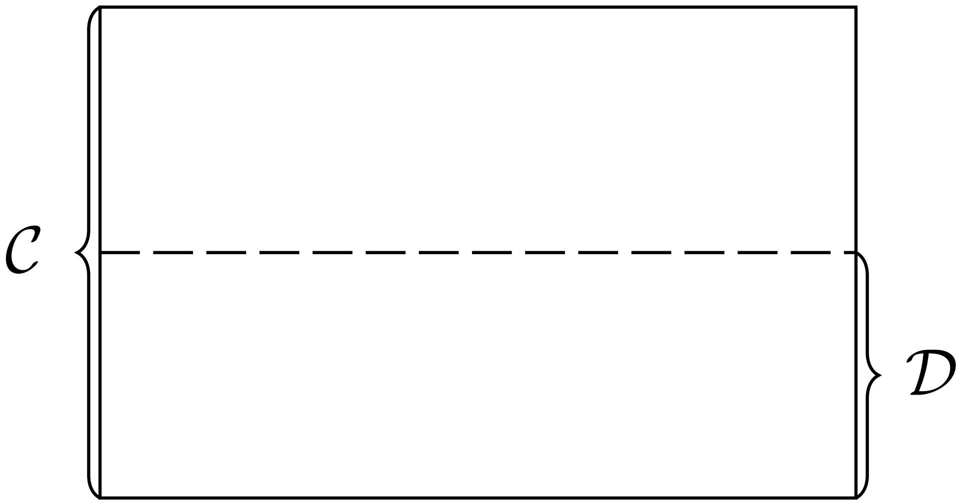
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



Relativization

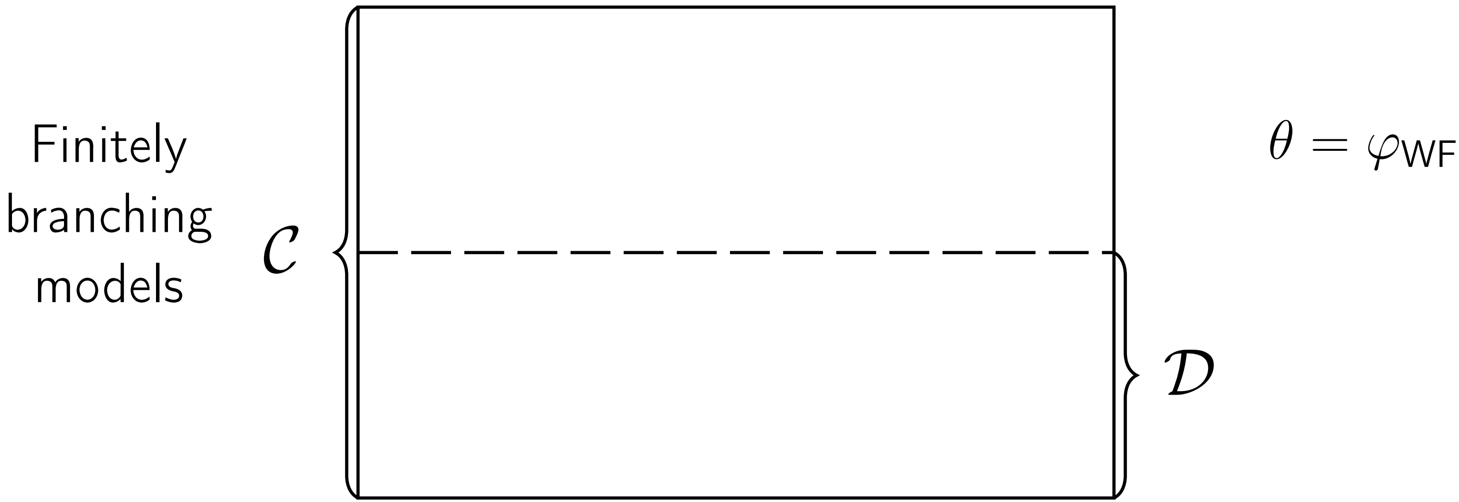
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\quad \mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.

Finitely branching models



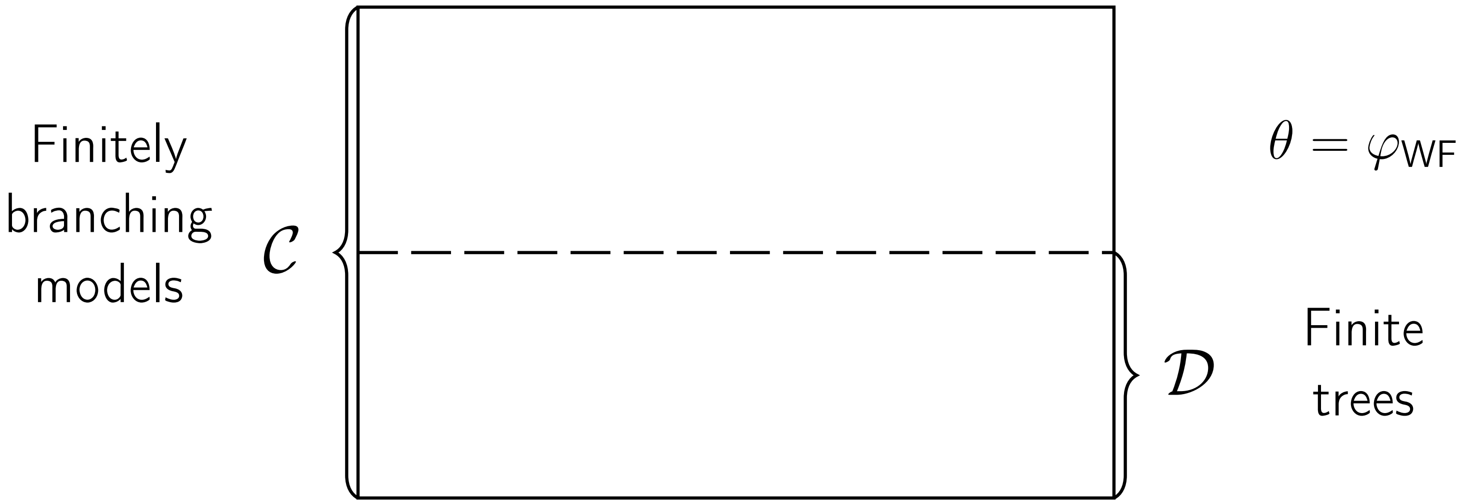
Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



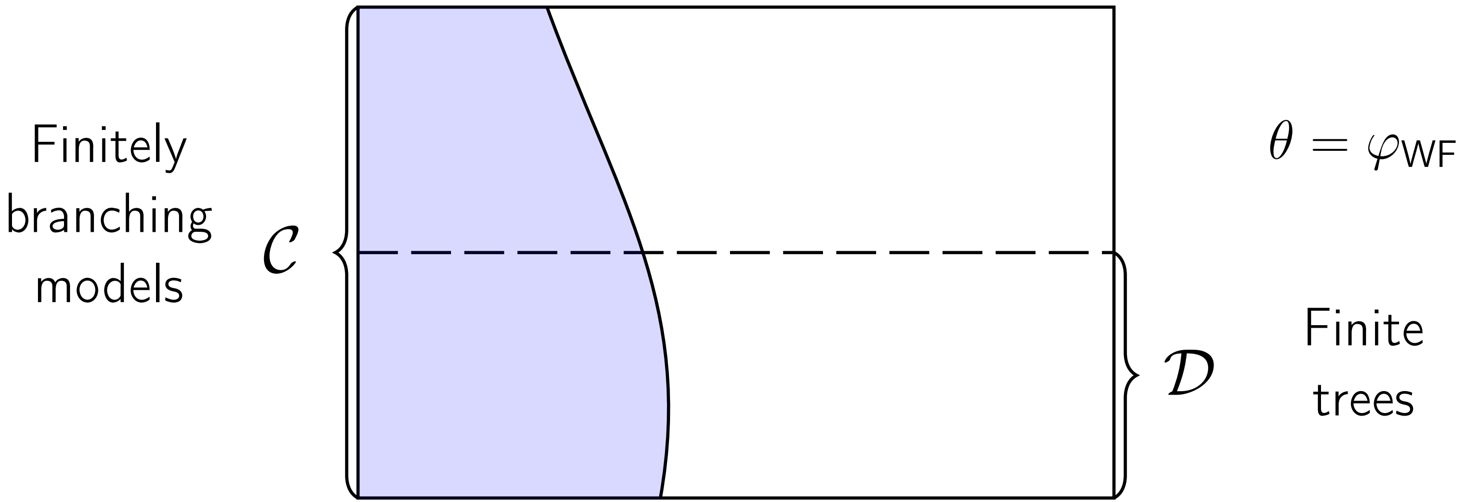
Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\quad \mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



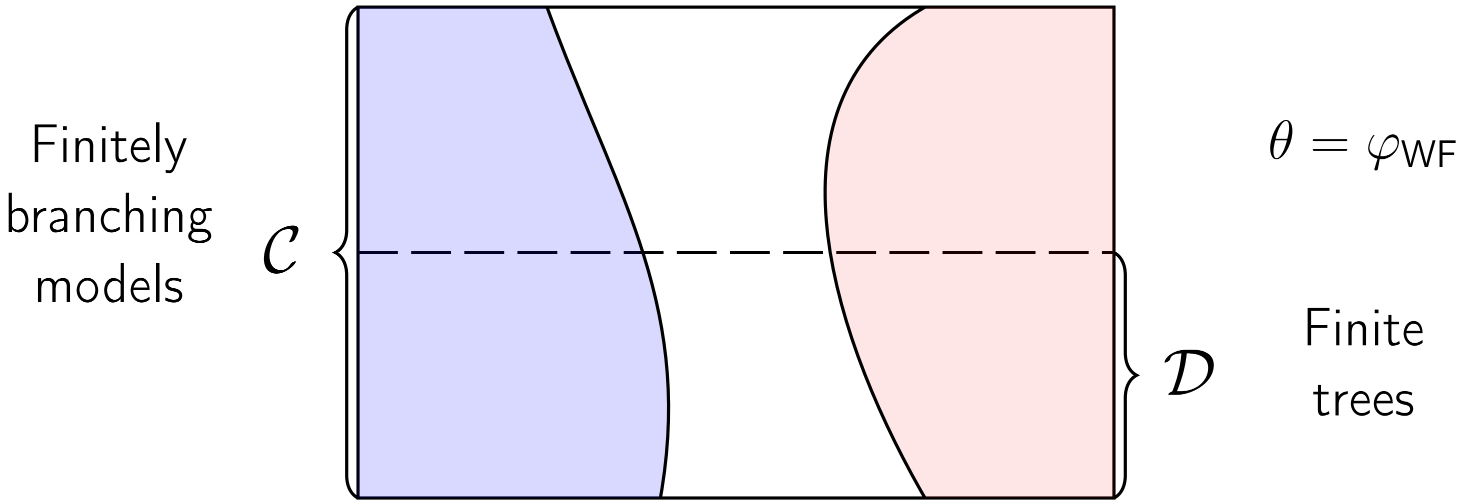
Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



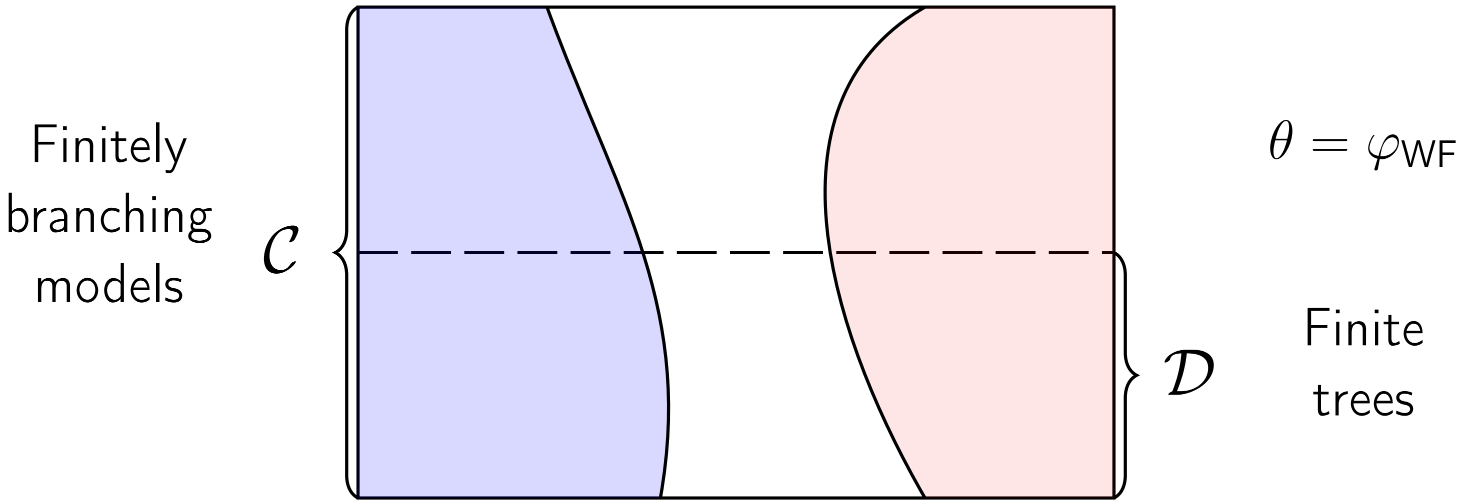
Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



Relativization

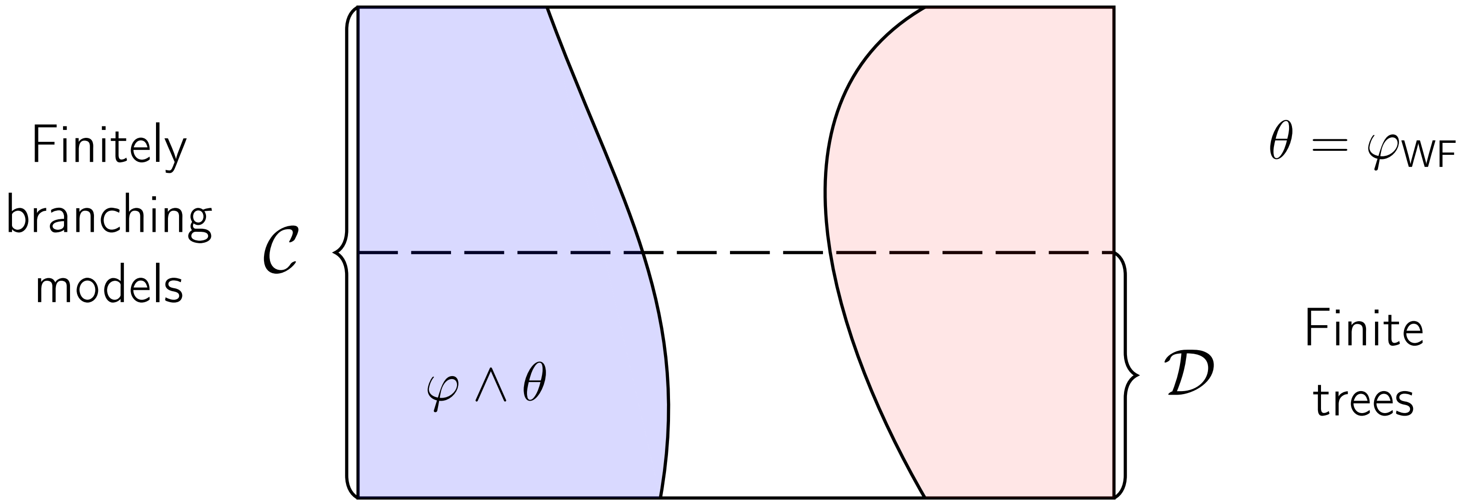
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



- Then: ψ separates φ from φ' over $\mathcal{D}$ iff it separates $\varphi \wedge \theta$ from $\varphi' \wedge \theta$ over $\mathcal{C}$.

Relativization

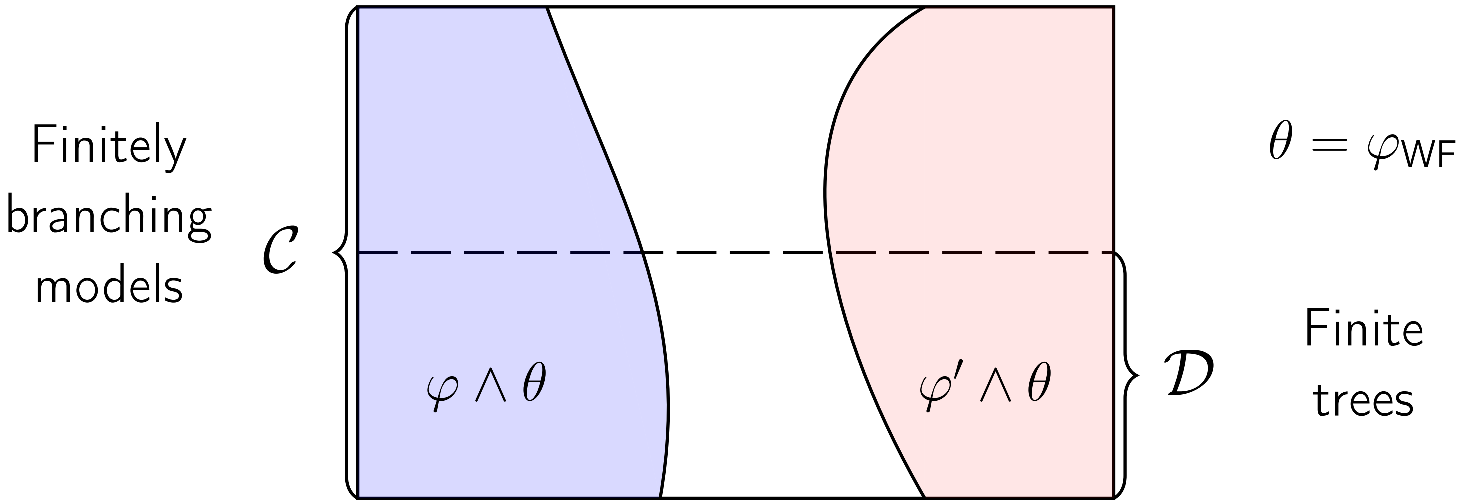
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



- Then: ψ separates φ from φ' over $\mathcal{D}$ iff it separates $\varphi \wedge \theta$ from $\varphi' \wedge \theta$ over $\mathcal{C}$.

Relativization

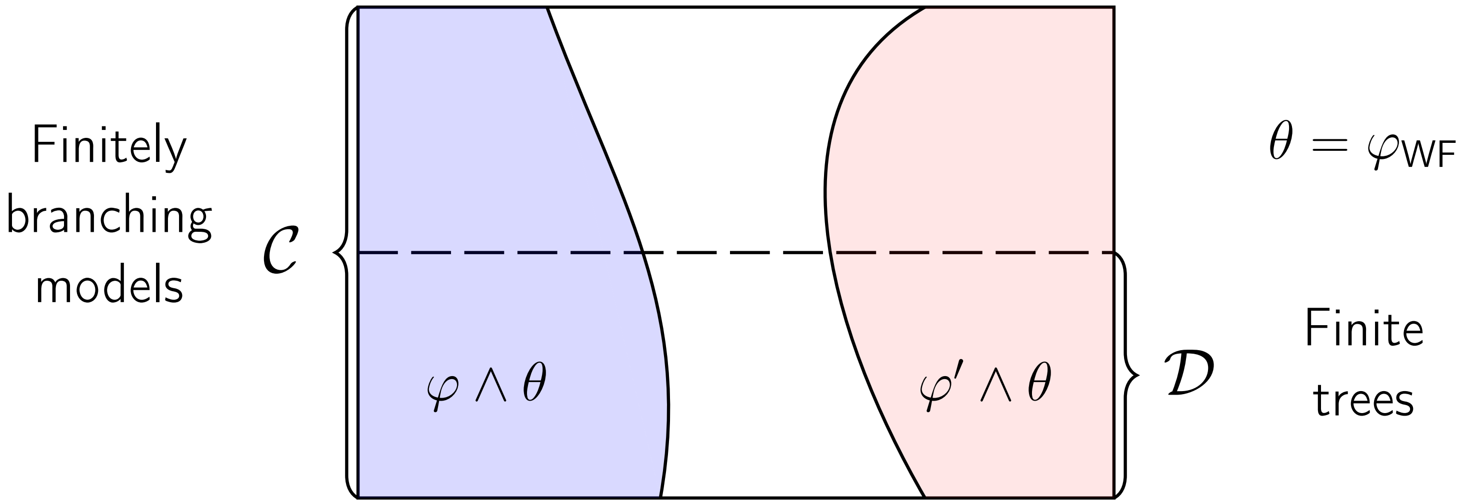
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



- Then: ψ separates φ from φ' over $\mathcal{D}$ iff it separates $\varphi \wedge \theta$ from $\varphi' \wedge \theta$ over $\mathcal{C}$.

Relativization

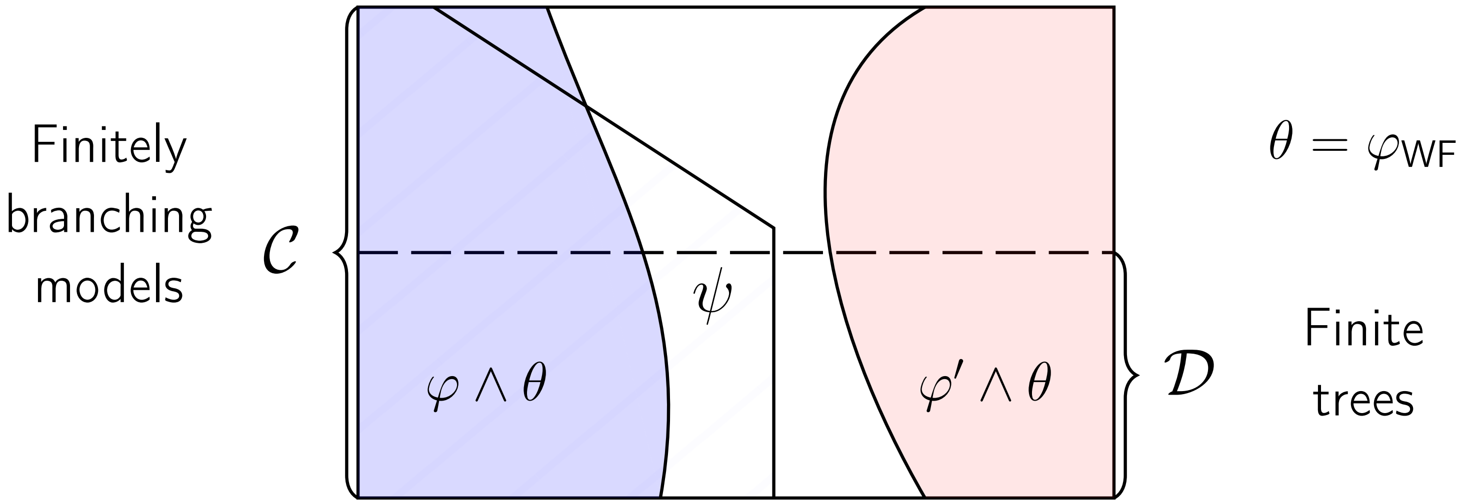
- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



- Then: ψ separates φ from φ' over $\mathcal{D}$ iff it separates $\varphi \wedge \theta$ from $\varphi' \wedge \theta$ over $\mathcal{C}$.
- Example: ψ separates φ from φ' over <u>finite</u> words
- iff it separates $\varphi \wedge \varphi_{\mathsf{WF}}$ from $\varphi' \wedge \varphi_{\mathsf{WF}}$ over (arbitrary) words

Relativization

- Assume classess of models $\mathcal{C}$ and $\mathcal{D}$ and formula θ such that
- θ defines $\mathcal{D}$ in $\mathcal{C}$: $\mathcal{M} \in \mathcal{D}$ iff $\mathcal{M} \in \mathcal{C}$ and $\mathcal{M} \models \varphi$.



- Then: ψ separates φ from φ' over $\mathcal{D}$ iff it separates $\varphi \wedge \theta$ from $\varphi' \wedge \theta$ over $\mathcal{C}$.
- Example: ψ separates φ from φ' over finite words
- iff it separates $\varphi \wedge \varphi_{\mathsf{WF}}$ from $\varphi' \wedge \varphi_{\mathsf{WF}}$ over (arbitrary) words