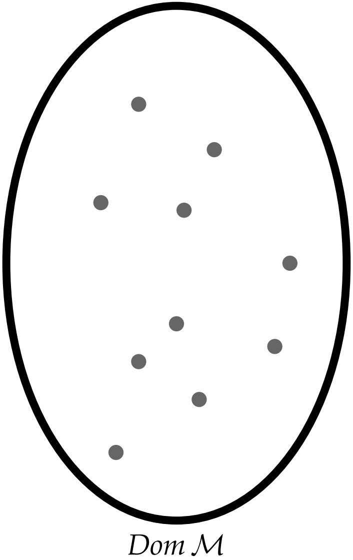
On the size of shortest modal descriptions

Santiago Figueira Daniel Gorín

CS Department
Universidad de Buenos Aires
Argentina

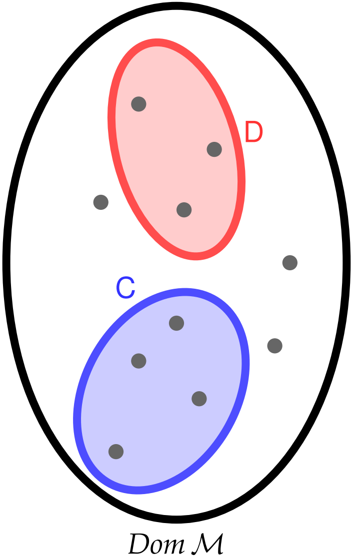
Advances in Modal Logic
Moscow, August 24th 2010

Separations and descriptions



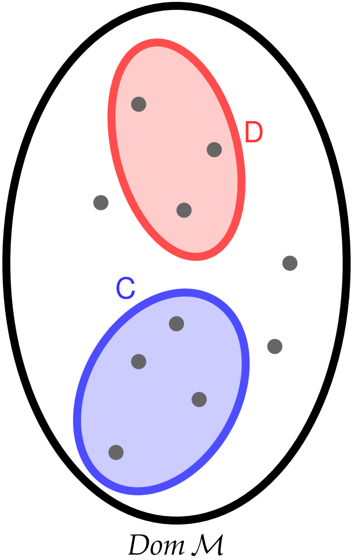
- Let $\mathcal{M}$ be a Kripke model

Separations and descriptions



- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq \mathsf{C}, \mathsf{D} \subset \mathit{Dom}\,\mathcal{M}$

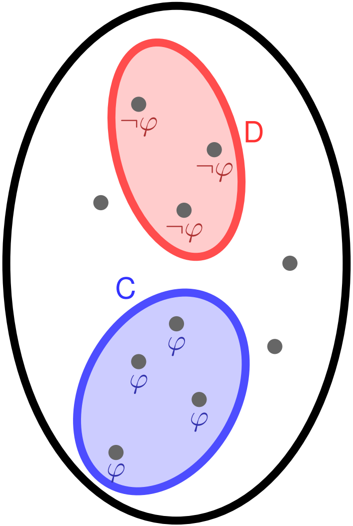
Separations and descriptions



- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq \mathsf{C}, \mathsf{D} \subset \mathit{Dom}\,\mathcal{M}$
- For any formula φ, we say that:
  - φ *separates* C and D in $\mathcal{M}$ iff

$$\mathcal{M}, \mathsf{C} \models \varphi \text{ and } \mathcal{M}, \mathsf{D} \not\models \varphi$$

Separations and descriptions

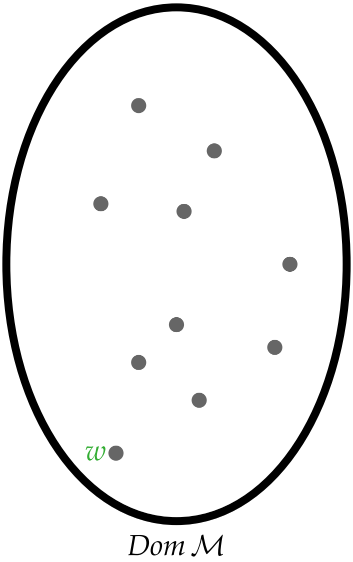


Dom $\mathcal{M}$

- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq$ C, D $\subset$ *Dom* $\mathcal{M}$
- For any formula φ, we say that:
  - φ *separates* C and D in $\mathcal{M}$ iff

$$\mathcal{M}, \mathsf{C} \models \varphi \text{ and } \mathcal{M}, \mathsf{D} \not\models \varphi$$

Separations and descriptions



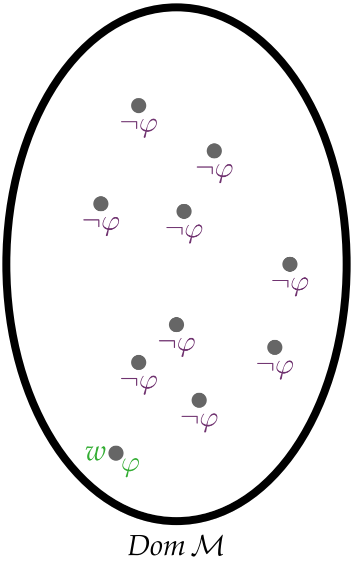
- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq \mathsf{C}, \mathsf{D} \subset \mathit{Dom}\,\mathcal{M}$
- For any formula φ, we say that:
  - φ *separates* C and D in $\mathcal{M}$ iff

$$\mathcal{M}, \mathsf{C} \models \varphi \text{ and } \mathcal{M}, \mathsf{D} \not\models \varphi$$

- Similarly, for $w \in \mathit{Dom}\,\mathcal{M}$ we say:
  - φ *describes* w in $\mathcal{M}$ iff

$$\mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, v \not\models \varphi \forall v \neq w$$

Separations and descriptions



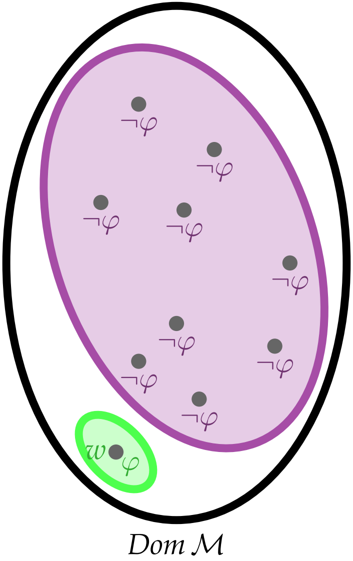
- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq \mathsf{C}, \mathsf{D} \subset Dom\,\mathcal{M}$
- For any formula φ, we say that:
  - φ *separates* C and D in $\mathcal{M}$ iff

$$\mathcal{M}, \mathsf{C} \models \varphi \text{ and } \mathcal{M}, \mathsf{D} \not\models \varphi$$

- Similarly, for $w \in Dom\,\mathcal{M}$ we say:
  - φ *describes* w in $\mathcal{M}$ iff

$$\mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, v \not\models \varphi \forall v \neq w$$

Separations and descriptions



- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq \mathsf{C}, \mathsf{D} \subset \mathit{Dom}\ \mathcal{M}$
- For any formula φ, we say that:
  - φ *separates* C and D in $\mathcal{M}$ iff

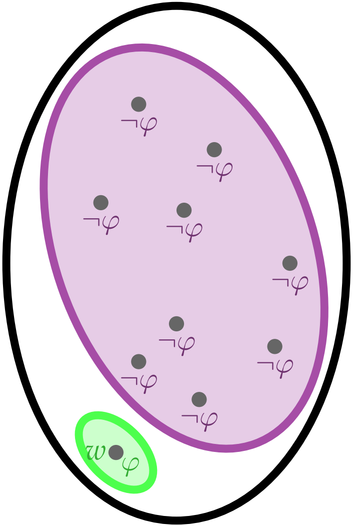
$$\mathcal{M}, \mathsf{C} \models \varphi \text{ and } \mathcal{M}, \mathsf{D} \not\models \varphi$$

- Similarly, for $w \in \mathit{Dom}\ \mathcal{M}$ we say:
  - φ *describes* w in $\mathcal{M}$ iff

$$\mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, v \not\models \varphi \forall v \neq w$$

- Description is a form of separation

Separations and descriptions



Dom $\mathcal{M}$

- Let $\mathcal{M}$ be a Kripke model
- And let $\emptyset \neq \mathsf{C}, \mathsf{D} \subset Dom\ \mathcal{M}$
- For any formula φ, we say that:
  - φ *separates* C and D in $\mathcal{M}$ iff

$$\mathcal{M}, \mathsf{C} \models \varphi \text{ and } \mathcal{M}, \mathsf{D} \not\models \varphi$$

- Similarly, for $w \in Dom\ \mathcal{M}$ we say:
  - φ *describes* w in $\mathcal{M}$ iff

$$\mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, v \not\models \varphi \forall v \neq w$$

- Description is a form of separation
- φ could be of any suitable logic

Separation and description problems

The separation problem

Given a finite model $\mathcal{M}$ and sets C, D $\subset$ *Dom* $\mathcal{M}$, find a φ that separates C and D, if possible.

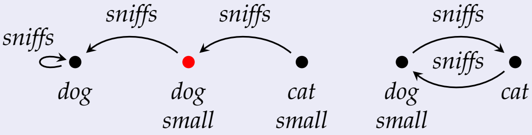
The description problem

Given a finite model $\mathcal{M}$ and a world $w \in$ *Dom* $\mathcal{M}$, find a φ that describes w, if possible.

- They can be seen as another kind of inference task
- But they didn't receive much attention so far
- We are interested in their computational properties

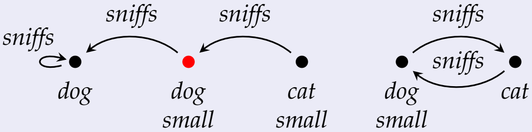
Motivation: Generation of Referring Expressions

An application of logics in Natural Language Generation



Motivation: Generation of Referring Expressions

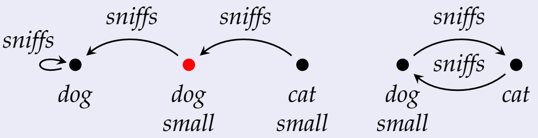
An application of logics in Natural Language Generation



- "the dog that is sniffing another dog"

Motivation: Generation of Referring Expressions

An application of logics in Natural Language Generation



content determination

(e.g., as a FOL formula)

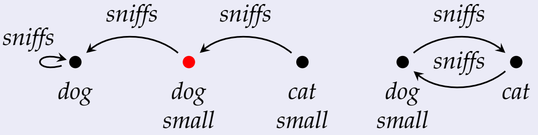
- $dog(x) \wedge \exists y.(x \not\approx y \wedge dog(y) \wedge sniffs(x, y))$

surface realization

- "the dog that is sniffing another dog"

Motivation: Generation of Referring Expressions

An application of logics in Natural Language Generation



content determination
(e.g., as a FOL formula)

- $dog(x) \wedge \exists y.(x \not\approx y \wedge dog(y) \wedge sniffs(x, y))$

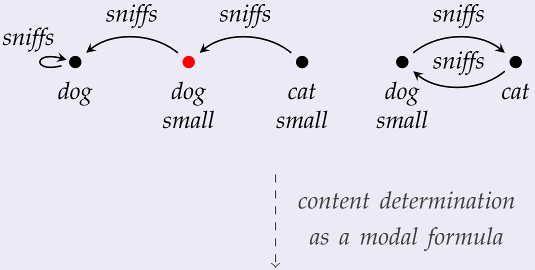
surface realization

- "the dog that is sniffing another dog"

(logical) content determination ≈ description problem

Motivation: Generation of Referring Expressions

An application of logics in Natural Language Generation

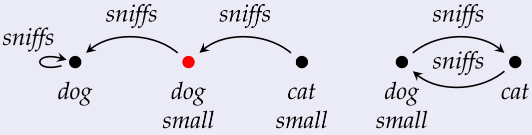


$$\bullet \quad dog \wedge small \wedge \langle sniffs \rangle dog$$

(logical) content determination ≈ description problem

Motivation: Generation of Referring Expressions

An application of logics in Natural Language Generation



content determination as a modal formula

• $dog \wedge small \wedge \langle sniffs \rangle dog$

surface realization

• "the small dog that is sniffing a dog"

(logical) content determination ≈ description problem

Motivation: Modal logics in the GRE

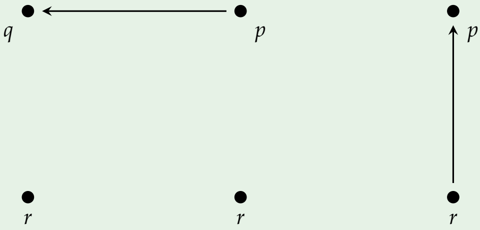
Areces, Koller & Striegnitz (2008)

- They propose modal logics for content determination:
  $\mathcal{ML}$ – the basic modal language $(\neg, \wedge, \Diamond)$
  $\mathcal{EL}$ – the existential positive fragment of $\mathcal{ML}$ $(\wedge, \Diamond)$
- *Rationale*:
  - Good expressive power
  - Simple surface realization algorithms
  - Relatively low computational complexity for inference tasks
- In particular, they show that:

  "The modal description problem needs polynomial time"

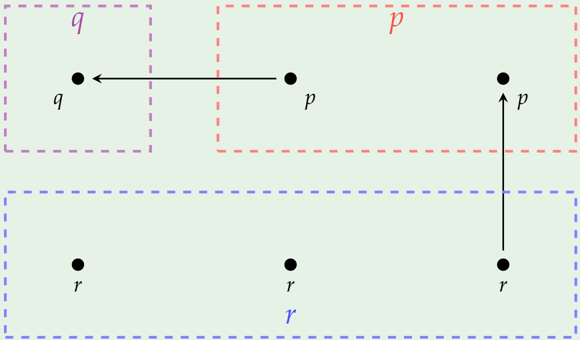
The modal description problem in polynomial time

A variation of Tarjan's bisimulation contraction algorithm



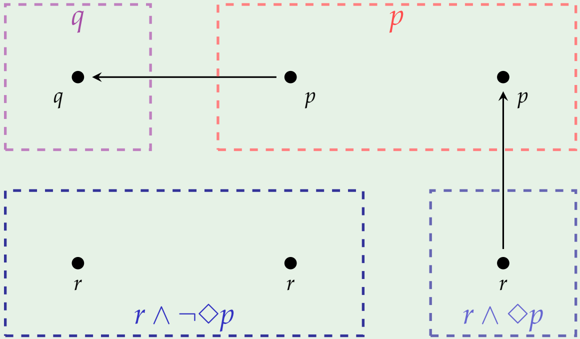
The modal description problem in polynomial time

A variation of Tarjan's bisimulation contraction algorithm



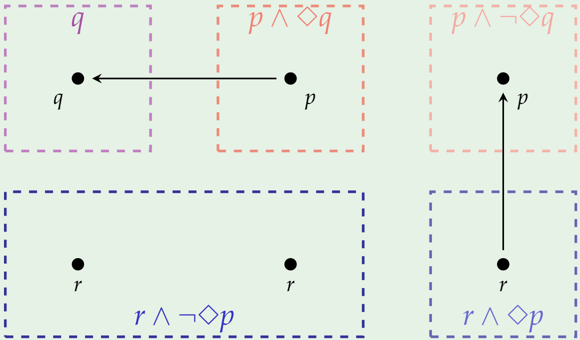
The modal description problem in polynomial time

A variation of Tarjan's bisimulation contraction algorithm



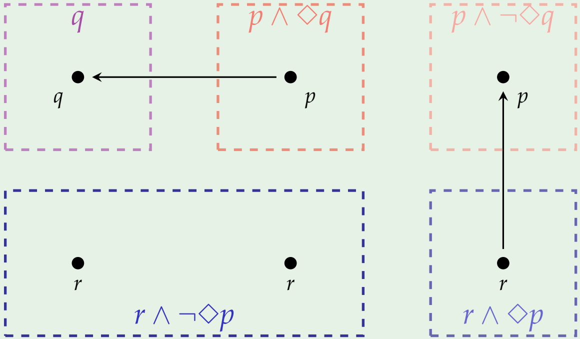
The modal description problem in polynomial time

A variation of Tarjan's bisimulation contraction algorithm



The modal description problem in polynomial time

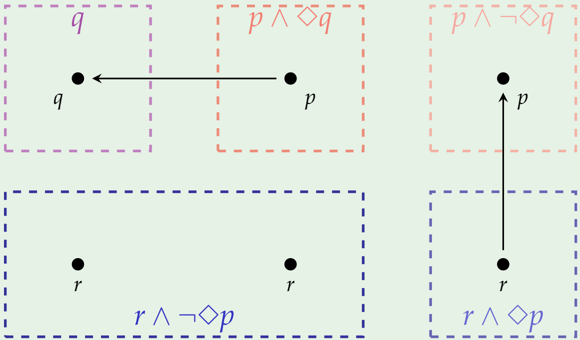
A variation of Tarjan's bisimulation contraction algorithm



- Tarjan's algorithm runs in polynomial time
- Hence, the modal description problem is polynomial

The modal description problem in polynomial time

A variation of Tarjan's bisimulation contraction algorithm



- Tarjan's algorithm runs in polynomial time
- Hence, the modal description problem is polynomial
- But this is assuming that $\wedge$ takes constant time!

The modal description problem in polynomial time

for DAG representation!

- This algorithm produces a formula represented as a DAG

The modal description problem in polynomial time

for DAG representation!

- This algorithm produces a formula represented as a DAG
- The size of the DAG is polynomial in the size of the model

The modal description problem in polynomial time

for DAG representation!

- This algorithm produces a formula represented as a DAG
- The size of the DAG is polynomial in the size of the model
- Surface realization step doesn't exploit DAG representation

The modal description problem in polynomial time

for DAG representation!

- This algorithm produces a formula represented as a DAG
- The size of the DAG is polynomial in the size of the model
- Surface realization step doesn’t exploit DAG representation
  - Most probably can’t be done anyway

The modal description problem in polynomial time

for DAG representation!

- This algorithm produces a formula represented as a DAG
- The size of the DAG is polynomial in the size of the model
- Surface realization step doesn't exploit DAG representation
  - Most probably can't be done anyway
- Is the *tree* representation of this formula also polynomial?

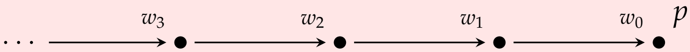
The modal description problem in polynomial time

for DAG representation!

- This algorithm produces a formula represented as a DAG
- The size of the DAG is polynomial in the size of the model
- Surface realization step doesn't exploit DAG representation
    - Most probably can't be done anyway
- Is the *tree* representation of this formula also polynomial?
- If not, "modal content determination" can't be said to take polynomial time

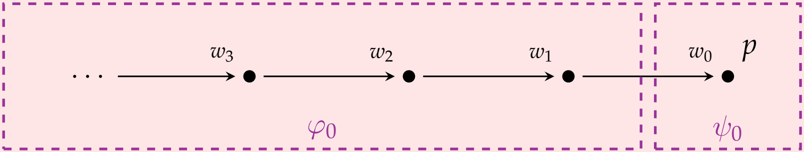
The modal description problem in polynomial time

also for tree representation?



The modal description problem in polynomial time

also for tree representation?

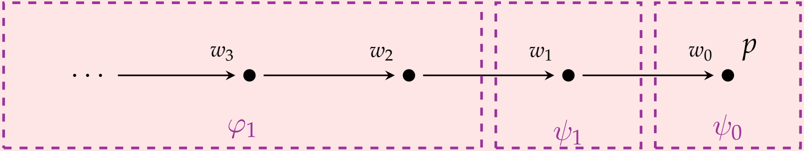


$\varphi_0 = \neg p$

$\psi_0 = p$

The modal description problem in polynomial time

also for tree representation?



$\varphi_0 = \neg p$

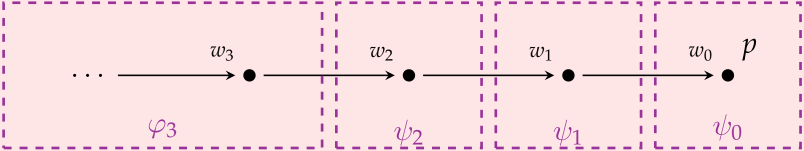
$\varphi_1 = \varphi_0 \wedge \neg\Diamond\psi_0$

$\psi_0 = p$

$\psi_1 = \varphi_0 \wedge \Diamond\psi_0$

The modal description problem in polynomial time

also for tree representation?



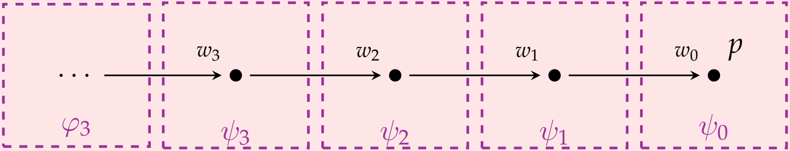
$$\varphi_0 = \neg p \qquad \psi_0 = p$$

$$\varphi_1 = \varphi_0 \wedge \neg\Diamond\psi_0 \qquad \psi_1 = \varphi_0 \wedge \Diamond\psi_0$$

$$\varphi_2 = \varphi_1 \wedge \neg\Diamond\psi_1 \qquad \psi_2 = \varphi_1 \wedge \Diamond\psi_1$$

The modal description problem in polynomial time

also for tree representation?



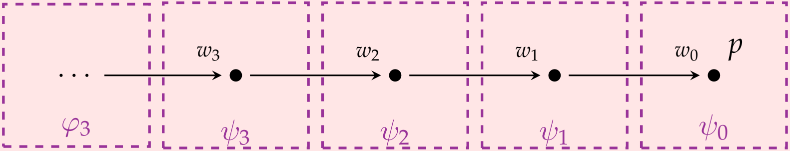
$$\begin{aligned} \varphi_0 &= \neg p \\ \varphi_1 &= \varphi_0 \wedge \neg\Diamond\psi_0 \\ \varphi_2 &= \varphi_1 \wedge \neg\Diamond\psi_1 \\ &\vdots \\ \varphi_{i+1} &= \varphi_i \wedge \neg\Diamond\psi_i \end{aligned}$$

$$\begin{aligned} \psi_0 &= p \\ \psi_1 &= \varphi_0 \wedge \Diamond\psi_0 \\ \psi_2 &= \varphi_1 \wedge \Diamond\psi_1 \\ &\vdots \\ \psi_{i+1} &= \varphi_i \wedge \Diamond\psi_i \end{aligned}$$

- Each ψ_i is description for w_i with size exponential in i

The modal description problem in polynomial time

also for tree representation?



$$
\begin{aligned}
\varphi_0 &= \neg p & \qquad \psi_0 &= p \\
\varphi_1 &= \varphi_0 \wedge \neg\Diamond\psi_0 & \qquad \psi_1 &= \varphi_0 \wedge \Diamond\psi_0 \\
\varphi_2 &= \varphi_1 \wedge \neg\Diamond\psi_1 & \qquad \psi_2 &= \varphi_1 \wedge \Diamond\psi_1 \\
&\vdots & &\vdots \\
\varphi_{i+1} &= \varphi_i \wedge \neg\Diamond\psi_i & \qquad \psi_{i+1} &= \varphi_i \wedge \Diamond\psi_i
\end{aligned}
$$

- Each ψ_i is description for w_i with size exponential in i
- Observe that w_i admits a linear description: $\underbrace{\Diamond\Diamond\ldots\Diamond}_{i \text{ times}} p$

Where do we go from here?

- The example shows that this algorithm is not polynomial
- Can we *fix* it?
- Can we find *another* one that is indeed polynomial?

Where do we go from here?

- The example shows that this algorithm is not polynomial
- Can we *fix* it?
- Can we find *another* one that is indeed polynomial?
- We show that no such algorithm exists!

Bounds for the separation / description problems

Basic modal language $\mathcal{ML}$

Theorem (Lower bound)

Any upper bound for the size of a solution for the separation or description problem for $\mathcal{ML}$ is at least exponential.

Corollary

No polynomial time algorithm exists that solves the description or separation problem returning the formula as a tree.

Bounds for the separation / description problems

Basic modal language $\mathcal{ML}$

Theorem (Lower bound)

Any upper bound for the size of a solution for the separation or description problem for $\mathcal{ML}$ is at least exponential.

Corollary

No polynomial time algorithm exists that solves the description or separation problem returning the formula as a tree.

Theorem (Upper bound)

If $\varphi \in \mathcal{ML}$ is a minimum description for v in $\mathcal{M} = \langle W, R, V \rangle$, then $|\varphi| \in O(2^{\frac{1}{2}|W|^2} \cdot |V|)$.

Proof of the lower bound for $\mathcal{ML}$

Outline

Theorem (Lower bound)

Any upper bound for the size of a solution for the separation or description problem for $\mathcal{ML}$ is at least exponential.

Proof outline.

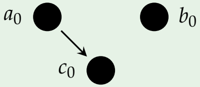
- Give a family of finite models with two distinguished points, $(\mathcal{M}_n, a_n, b_n)_{n \in \mathbb{N}}$, such that:
  i. $|\mathcal{M}_n| \in O(n)$
  ii. The shortest formula separating a_n and b_n is exponential in n
  iii. There exists a description of a_n in $\mathcal{M}_n$
- From i. and ii. the lower bound for separation follows
- This plus iii. proves the lower bound for description

□

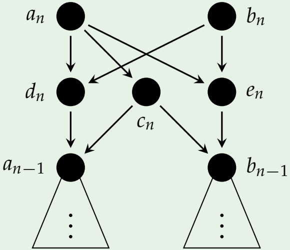
Proof of the lower bound for $\mathcal{ML}$

The (recursive) family of models

$(\mathcal{M}_0, a_0, b_0)$



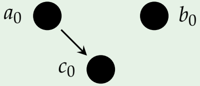
$(\mathcal{M}_n, a_n, b_n), n > 0$



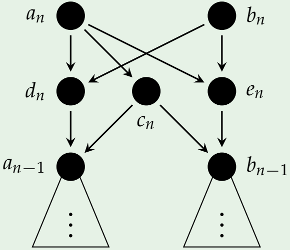
Proof of the lower bound for $\mathcal{ML}$

The (recursive) family of models

$(\mathcal{M}_0, a_0, b_0)$



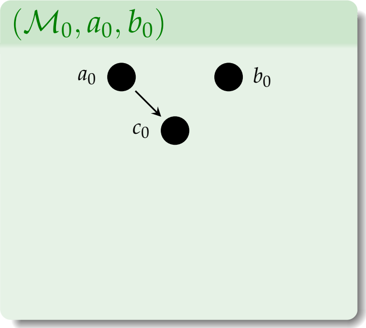
$(\mathcal{M}_n, a_n, b_n), n > 0$

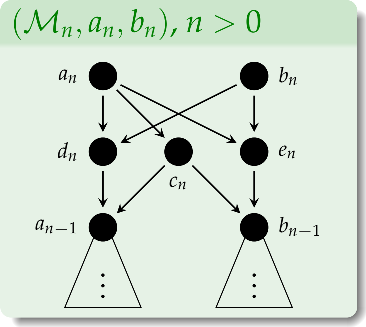


i. $|\mathcal{M}_n| = O(n)$

Proof of the lower bound for $\mathcal{ML}$

The (recursive) family of models





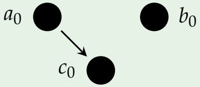
i. $|\mathcal{M}_n| = O(n)$

iii. If φ separates a_n and b_n, then $\varphi \wedge \Diamond^{2n+1}\top$ describes a_n in $\mathcal{M}_n$

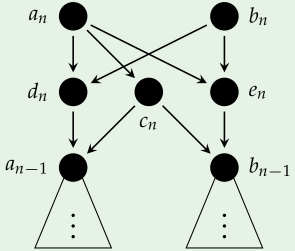
Proof of the lower bound for $\mathcal{ML}$

The (recursive) family of models

$(\mathcal{M}_0, a_0, b_0)$



$(\mathcal{M}_n, a_n, b_n), n > 0$



i. $|\mathcal{M}_n| = O(n)$

ii. We need to verify that

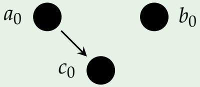
- a_n and b_n can be indeed separated by some ψ

iii. If φ separates a_n and b_n, then $\varphi \wedge \Diamond^{2n+1}\top$ describes a_n in $\mathcal{M}_n$

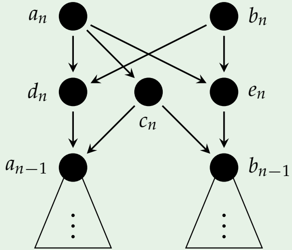
Proof of the lower bound for $\mathcal{ML}$

The (recursive) family of models

$(\mathcal{M}_0, a_0, b_0)$



$(\mathcal{M}_n, a_n, b_n), n > 0$



i. $|\mathcal{M}_n| = O(n)$

ii. We need to verify that

- a_n and b_n can be indeed separated by some ψ
- any such ψ will have size exponential in n

iii. If φ separates a_n and b_n, then $\varphi \wedge \Diamond^{2n+1}\top$ describes a_n in $\mathcal{M}_n$

Proof of the lower bound for $\mathcal{ML}$

Ehrenfeucht-Fraïssé games and strategies

- Ehrenfeucht-Fraïssé games: classical model-theoretical tool
- $\mathcal{G}_{\mathcal{M}}(a, b, n)$: n-turn game played by Spoiler and Duplicator
- Thm: Duplicator has a winning strategy for $\mathcal{G}_{\mathcal{M}}(a, b, n)$ iff no formula with n nested $\Diamond$'s separates a and b in $\mathcal{M}$

Proof of the lower bound for $\mathcal{ML}$

Ehrenfeucht-Fraïssé games and strategies

- Ehrenfeucht-Fraïssé games: classical model-theoretical tool
- $\mathcal{G}_{\mathcal{M}}(a, b, n)$: n-turn game played by Spoiler and Duplicator
- Thm: Duplicator has a winning strategy for $\mathcal{G}_{\mathcal{M}}(a, b, n)$ iff no formula with n nested $\Diamond$'s separates a and b in $\mathcal{M}$
- Dually, Spoiler has a winning strategy for $\mathcal{G}_{\mathcal{M}}(a, b, n)$ iff a formula with at most n nested $\Diamond$'s separates a and b in $\mathcal{M}$

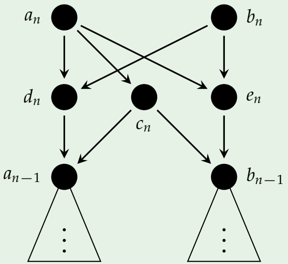
Proof of the lower bound for $\mathcal{ML}$

Ehrenfeucht-Fraïssé games and strategies

- Ehrenfeucht-Fraïssé games: classical model-theoretical tool
- $\mathcal{G}_{\mathcal{M}}(a, b, n)$: n-turn game played by Spoiler and Duplicator
- Thm: Duplicator has a winning strategy for $\mathcal{G}_{\mathcal{M}}(a, b, n)$ iff no formula with n nested $\Diamond$'s separates a and b in $\mathcal{M}$
- Dually, Spoiler has a winning strategy for $\mathcal{G}_{\mathcal{M}}(a, b, n)$ iff a formula with at most n nested $\Diamond$'s separates a and b in $\mathcal{M}$
- Moreover, we can relate the *size* of Spoiler's strategy with that of the separating formula

Proof of the lower bound for $\mathcal{ML}$

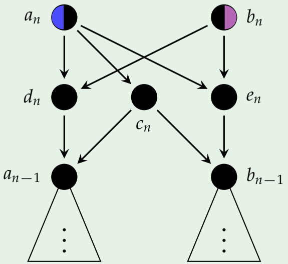
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

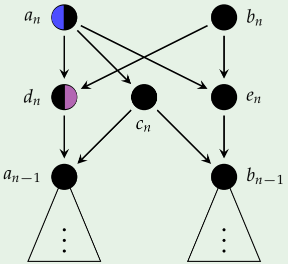
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

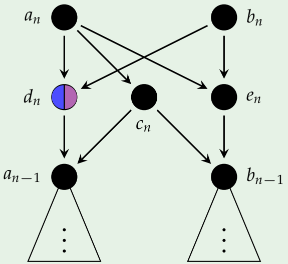
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

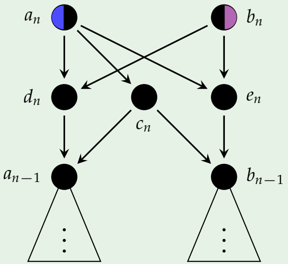
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

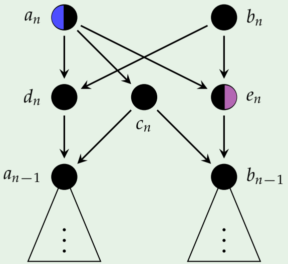
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

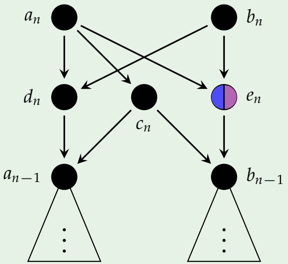
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n + 1)$

Proof of the lower bound for $\mathcal{ML}$

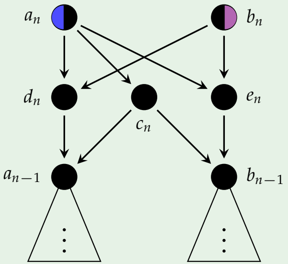
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

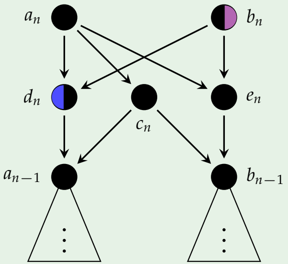
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

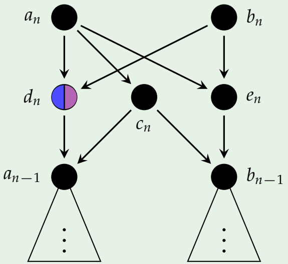
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

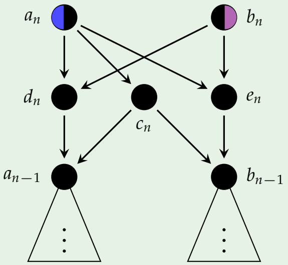
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

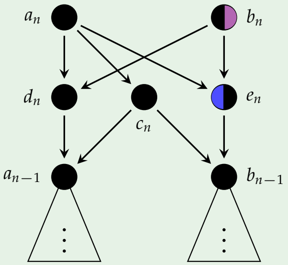
$(\mathcal{M}_n, a_n, b_n), n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

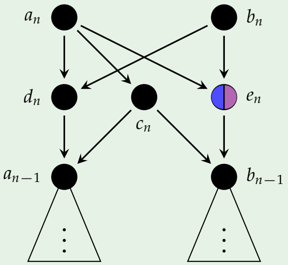
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

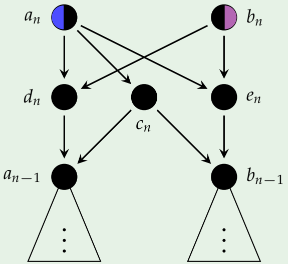
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

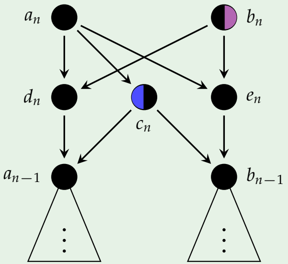
$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$

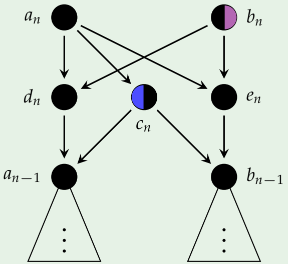


Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

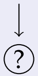
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



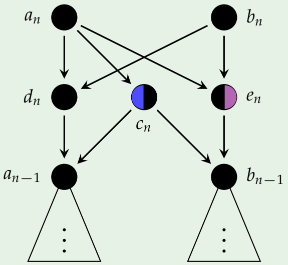
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



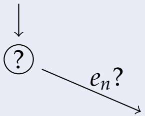
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n), n > 0$



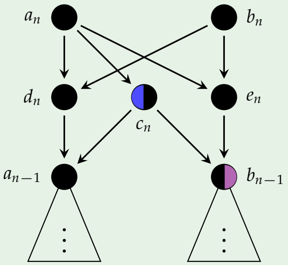
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



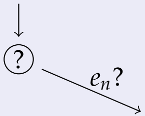
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



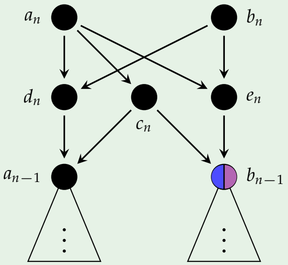
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



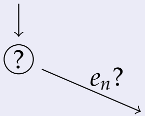
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



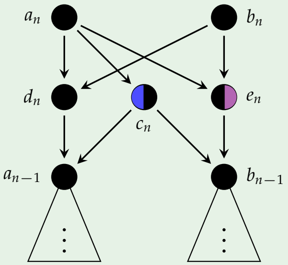
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



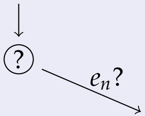
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



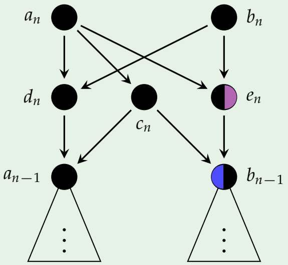
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



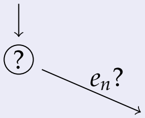
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



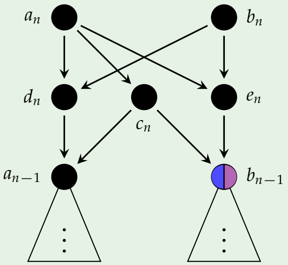
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



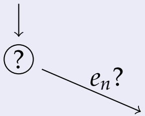
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



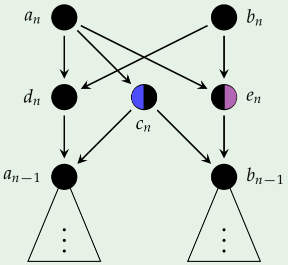
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



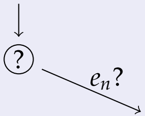
Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



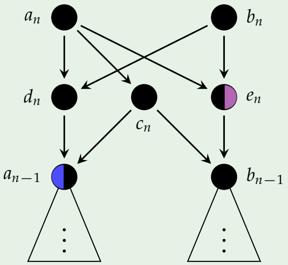
Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n



Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

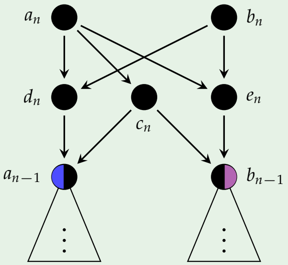
?

e_n?

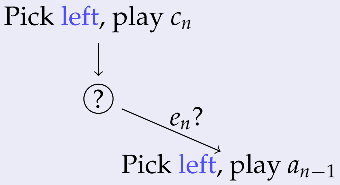
Pick left, play a_{n-1}

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$

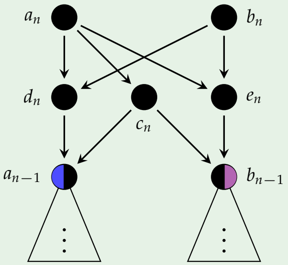


Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$



Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

?

e_n?

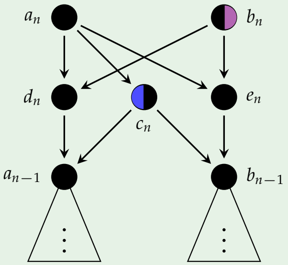
Pick left, play a_{n-1}

↓

Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

?

e_n?

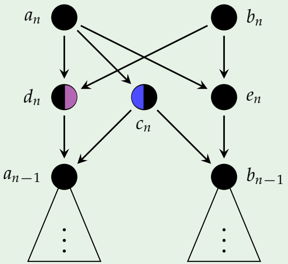
Pick left, play a_{n-1}

↓

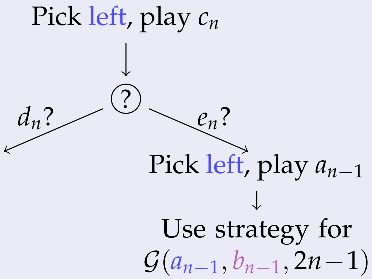
Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$

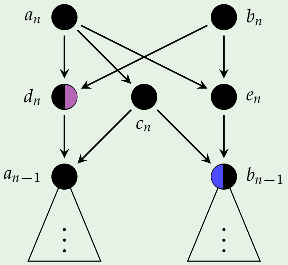


Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$



Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

?

d_n? → Pick left, play b_{n-1}

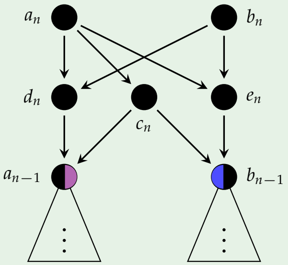
e_n? → Pick left, play a_{n-1}

↓

Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n), n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

?

d_n? → Pick left, play b_{n-1}

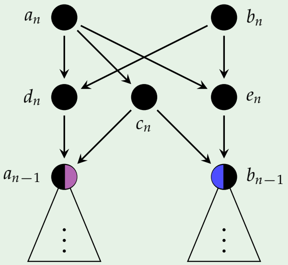
e_n? → Pick left, play a_{n-1}

↓

Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

$\downarrow$

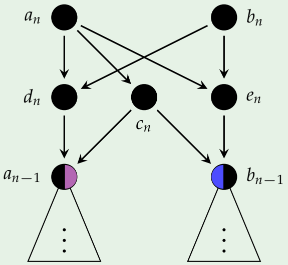
?

d_n? → Pick left, play b_{n-1} $\downarrow$ Use strategy for $\mathcal{G}(b_{n-1}, a_{n-1}, 2n-1)$

e_n? → Pick left, play a_{n-1} $\downarrow$ Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n), n > 0$



Spoiler's strategy for $\mathcal{G}(b_n, a_n, 2n+1)$

Pick right, play c_n

↓

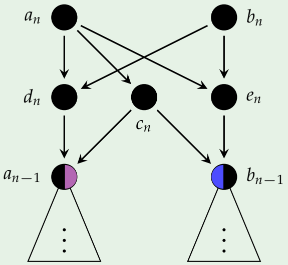
?

d_n? → Pick right, play b_{n-1} ↓ Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

e_n? → Pick right, play a_{n-1} ↓ Use strategy for $\mathcal{G}(b_{n-1}, a_{n-1}, 2n-1)$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

?

d_n? → Pick left, play b_{n-1} ↓ Use strategy for $\mathcal{G}(b_{n-1}, a_{n-1}, 2n-1)$

e_n? → Pick left, play a_{n-1} ↓ Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

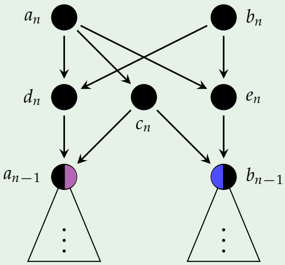
Intuitive argument

- The strategy induces φ_i, exponential in i, that separates a_i and b_i:

$$\varphi_0 = \Diamond\top \qquad \varphi_{n+1} = \Diamond(\Diamond\neg\varphi_n \wedge \Diamond\varphi_n)$$

Proof of the lower bound for $\mathcal{ML}$

$(\mathcal{M}_n, a_n, b_n)$, $n > 0$



Spoiler's strategy for $\mathcal{G}(a_n, b_n, 2n+1)$

Pick left, play c_n

↓

?

d_n? → Pick left, play b_{n-1} ↓ Use strategy for $\mathcal{G}(b_{n-1}, a_{n-1}, 2n-1)$

e_n? → Pick left, play a_{n-1} ↓ Use strategy for $\mathcal{G}(a_{n-1}, b_{n-1}, 2n-1)$

Intuitive argument

- The strategy induces φ_i, exponential in i, that separates a_i and b_i:

$$\varphi_0 = \Diamond\top \qquad \varphi_{n+1} = \Diamond(\Diamond\neg\varphi_n \wedge \Diamond\varphi_n)$$

- *A smaller separating formula would correspond to a smaller strategy.*
- But we saw that this is the only winning strategy for Spoiler!

Proof of the *upper* bound for $\mathcal{ML}$

Theorem (Upper bound)

If $\varphi \in \mathcal{ML}$ *is a minimum description for* v *in* $\mathcal{M} = \langle W, R, V \rangle$*, then* $|\varphi| \in O(2^{\frac{1}{2}|W|^2} \cdot |V|)$.

Proof idea.

Follows directly by analyzing the size of a formula generated by bisimulation contraction. □

What if we drop negation?

- AKS (2008) also investigate $\mathcal{EL}$ for content determination
  - Very weak fragment of $\mathcal{ML}$, only $\wedge$ and $\Diamond$ can be used
  - "Formulas with arbitrary negation are harder to realize"

What if we drop negation?

- AKS (2008) also investigate $\mathcal{EL}$ for content determination
  - Very weak fragment of $\mathcal{ML}$, only $\wedge$ and $\Diamond$ can be used
  - “Formulas with arbitrary negation are harder to realize”
- The proof of the lower bound for $\mathcal{ML}$ does not work for $\mathcal{EL}$
  - No $\mathcal{EL}$-formula separates a_n and b_n

What if we drop negation?

- AKS (2008) also investigate $\mathcal{EL}$ for content determination
  - Very weak fragment of $\mathcal{ML}$, only $\wedge$ and $\Diamond$ can be used
  - "Formulas with arbitrary negation are harder to realize"
- The proof of the lower bound for $\mathcal{ML}$ does not work for $\mathcal{EL}$
  - No $\mathcal{EL}$-formula separates a_n and b_n
- Could it be that $\mathcal{EL}$ has a polynomial description problem?

What if we drop negation?

- AKS (2008) also investigate $\mathcal{EL}$ for content determination
  - Very weak fragment of $\mathcal{ML}$, only $\wedge$ and $\Diamond$ can be used
  - "Formulas with arbitrary negation are harder to realize"
- The proof of the lower bound for $\mathcal{ML}$ does not work for $\mathcal{EL}$
  - No $\mathcal{EL}$-formula separates a_n and b_n
- Could it be that $\mathcal{EL}$ has a polynomial description problem?
- We show it doesn't!

Bounds for the separation / description problems

$\mathcal{EL}$– the existential positive fragment of $\mathcal{ML}$

Theorem (Lower bound for $\mathcal{EL}$)

Any upper bound for the size of a solution for the separation or description problem for $\mathcal{EL}$ is at least exponential.

Bounds for the separation / description problems

$\mathcal{EL}$– the existential positive fragment of $\mathcal{ML}$

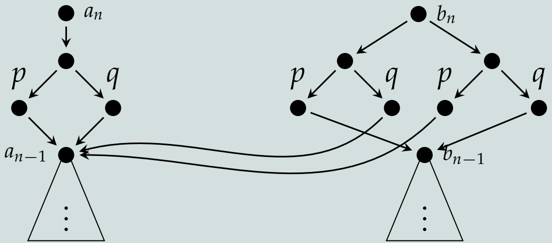
Theorem (Lower bound for $\mathcal{EL}$)

Any upper bound for the size of a solution for the separation or description problem for $\mathcal{EL}$ is at least exponential.

Proof idea.

Similar idea; the family of models is more complex, though:

$(\mathcal{N}_n, a_n, b_n), n > 0$



□

Succinctness

A quantitative way of comparing logics

“$\mathcal{L}_1$ is *more succinct* than $\mathcal{L}_2$”

- Fix a family of properties $(\Phi_n)_{n\in\mathbb{N}}$
- Express each Φ_n with a “short” formula φ_n of $\mathcal{L}_1$
- Prove: any ψ_n that expresses Φ_n in $\mathcal{L}_2$ is “much larger” than φ_n
- φ_n and ψ_n must be *logically equivalent*

Succinctness

A quantitative way of comparing logics

"$\mathcal{L}_1$ is *more succinct* than $\mathcal{L}_2$"

- Fix a family of properties $(\Phi_n)_{n\in\mathbb{N}}$
- Express each Φ_n with a "short" formula φ_n of $\mathcal{L}_1$
- Prove: any ψ_n that expresses Φ_n in $\mathcal{L}_2$ is "much larger" than φ_n
- φ_n and ψ_n must be *logically equivalent*

" $\mathcal{L}_1$ is *more succinct for separation* than $\mathcal{L}_2$"

- Fix a family of models and sets $(\mathcal{M}_n, C_n, D_n)_{n\in\mathbb{N}}$
- Separate C_n and D_n in $\mathcal{M}_n$ with a "short" formula φ_n of $\mathcal{L}_1$
- Prove: any ψ_n of $\mathcal{L}_2$ that separates C_n and D_n in $\mathcal{M}_n$ is "much larger" than φ_n
- φ_n and ψ_n need not be *logically equivalent*

Quantitative comparison of non-equivalent logics

Notation – $\mathcal{L}_1 \ll \mathcal{L}_2$

"$\mathcal{L}_1$ is exponentially more succinct for separation than $\mathcal{L}_2$"

Theorem

$$\mathcal{FO}^{=} \ll \mathcal{ML} \ll \mathcal{EL}^{\star} \ll \mathcal{EL}$$

$\mathcal{FO}^{=}$ First-order logic + equality (correspondence language)

$\mathcal{EL}^{\star}$ $\mathcal{EL}$ + atomic negation

Future work

- Computing the shortest formula that separate or describe
  - Application to Natural Language Generation
  - Good algorithms?
  - What's the complexity?
- Study other logics: relation with equality?
- Links with Kolmogorov complexity?

Aside: Колмогóров Complexity and Logics

What's the size of the shortest $\mathcal{L}$-formula that expresses property Φ?

$$E_{\mathcal{L}}(\Phi) = \min\{|\varphi| : \varphi \in \mathcal{L}, \mathcal{M} \text{ verifies } \Phi \text{ iff } \mathcal{M} \models \varphi\} \cup \{\infty\}$$

Kolmogorov Complexity: What's size of the the shortest M-program that computes string σ?

$$K_M(\sigma) = \min\{|p| : M(p) = \sigma\} \cup \{\infty\}$$

Algorithmic side

- Meaning of a program p:

  the output of p in M

- This meaning depends on M
- M extends M' then $K_M \leq K_{M'}$

Logical side

- Meaning of a formula φ:

  $\{\mathcal{M} : \mathcal{M} \models_{\mathcal{L}} \varphi\}$

- This meaning depends on $\mathcal{L}$
- $\mathcal{L}$ extends $\mathcal{L}'$ then $E_{\mathcal{L}} \leq E_{\mathcal{L}'}$

- Connections between Kolmogorov complexity and Logics?
- Transfer results from Kolmogorov complexity theory to Logics?
- How about other logical 'tasks' such as separation or description?