Modal Logics as Fragments of Classical Logic

Carlos Areces
carlos.areces@gmail.com

Spring Term 2018
Stanford

What we want to cover

- Kripke models vs. first-order models
- Translations into FOL
- Transfer results
- Optimize Translations
- Beyond FOL

Relevant Bibliography

- Chapter 2 of "Modal Logic," Blackburn, de Rijke & Venema. Look for the section 'Standard Translation' (Seccion 2.4).
- "Tree-Based Heuristics in Modal Theorem Proving," Areces, Gennari, Heguiabehere and de Rijke.
- "Unsorted Functional Translations," Areces and Gorín.

Kripke models vs. First-order models

Kripke models

First-order model

Kripke models vs. First-order models

Kripke models

- A non-empy domain W

First-order model

- A non-empy domain $\mathcal{D}$

Kripke models vs. First-order models

Kripke models

- A non-empy domain W
- One or more $R_i \subseteq W \times W$

First-order model

- A non-empy domain $\mathcal{D}$
- For each predicate P of arity n, a

$$P^{\mathcal{I}} \subseteq \underbrace{\mathcal{D} \times \ldots \times \mathcal{D}}_{n}$$

Kripke models vs. First-order models

Kripke models

- A non-empy domain W
- One or more $R_i \subseteq W \times W$
- A valuation function $V : \text{PROP} \to 2^W$

First-order model

- A non-empy domain $\mathcal{D}$
- For each predicate P of arity n, a

$$P^{\mathcal{I}} \quad \subseteq \quad \underbrace{\mathcal{D} \times \ldots \times \mathcal{D}}_{n}$$

Kripke models vs. First-order models

Kripke models

- A non-empy domain W
- One or more $R_i \subseteq W \times W$
- A valuation function $V : \text{PROP} \rightarrow 2^W$

First-order model

- A non-empy domain $\mathcal{D}$
- For each predicate P of arity n, a

$$P^{\mathcal{I}} \quad \subseteq \quad \underbrace{\mathcal{D} \times \ldots \times \mathcal{D}}_{n}$$

- It is easy to see that there is a one-to-one correspondence between the two.

Model correspondence

- Formally, a Kripke model

$$\mathcal{M} = \langle W, \{R_i\}_{i \in \text{MOD}}, V \rangle$$

defined over the signature $\mathcal{S} = \langle \text{PROP}, \text{MOD} \rangle$ corresponds to a first-oder model

$$\mathcal{I}^{\mathcal{M}} = \langle W, \cdot^{\mathcal{I}^{\mathcal{M}}} \rangle$$

over the (first-order) signature $\mathcal{S}' = \{P_i \mid i \in \text{PROP} \cup \text{MOD}\}$ (P_i is unary if $i \in \text{PROP}$; binary if $i \in \text{MOD}$) where

$$P_i^{\mathcal{I}^{\mathcal{M}}} = \begin{cases} V(i) & \text{si } i \in \text{PROP} \\ R_i & \text{si } i \in \text{MOD} \end{cases}$$

- $\mathcal{S}'$ is call the first-order correspondence language.

Formula correspondence

- We can also define a correspondence between formulas of the two languages.

Formula correspondence

- We can also define a correspondence between formulas of the two languages.
- We can define it as a translation between the two languages.

Formula correspondence

- We can also define a correspondence between formulas of the two languages.
- We can define it as a translation between the two languages.
- If the original formula and its translation are equivalent, then the original logic can be expressed by the target logic.

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle \varphi & \text{iff} & \text{exists } v \text{ s.t. } R_m wv \text{ and } \mathcal{M}, v \models \varphi
\end{array}
$$

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle \varphi & \text{iff} & \text{exists } v \text{ s.t. } R_m wv \text{ and } \mathcal{M}, v \models \varphi
\end{array}
$$

1. Rewrite "english" in "logic"…

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle \varphi & \text{iff} & \exists v \,.\, R_m wv \wedge \mathcal{M}, v \models \varphi
\end{array}
$$

1. Rewrite "english" in "logic"…

The standard translation

$$\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \wedge \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle \varphi & \text{iff} & \exists v \,.\, R_m wv \wedge \mathcal{M}, v \models \varphi
\end{array}$$

1. Rewrite “english” in “logic”…

The standard translation

$$\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \neg\mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \wedge \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle\varphi & \text{iff} & \exists v \,.\, R_m wv \wedge \mathcal{M}, v \models \varphi
\end{array}$$

1. Rewrite “english” in “logic”…

The standard translation

$$\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \neg\mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \wedge \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle\varphi & \text{iff} & \exists v \,.\, R_m wv \wedge \mathcal{M}, v \models \varphi
\end{array}$$

1. Rewrite “english” in “logic”…
2. Remember the equivalence between models …

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \neg\mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \wedge \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle \varphi & \text{iff} & \exists v \,.\, P_m(w, v) \wedge \mathcal{M}, v \models \varphi
\end{array}
$$

1. Rewrite "english" in "logic"…
2. Remember the equivalence between models …

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & P_p(w) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \neg\mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \wedge \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle \varphi & \text{iff} & \exists v \,.\, P_m(w, v) \wedge \mathcal{M}, v \models \varphi
\end{array}
$$

1. Rewrite “english” in “logic”. . .
2. Remember the equivalence between models . . .

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & P_p(w) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \neg\mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \wedge \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle m \rangle\varphi & \text{iff} & \exists v \,.\, P_m(w, v) \wedge \mathcal{M}, v \models \varphi
\end{array}
$$

1. Rewrite “english” in “logic”…
2. Remember the equivalence between models …
3. v and w are elements in the model, we represent them as variables…

The standard translation

$$\begin{array}{llll}
\mathcal{M}, x & \models p & \text{iff} & P_p(x) \\
\mathcal{M}, x & \models \neg\varphi & \text{iff} & \neg \mathcal{M}, x \models \varphi \\
\mathcal{M}, x & \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, x \models \varphi \wedge \mathcal{M}, x \models \psi \\
\mathcal{M}, x & \models \langle m \rangle \varphi & \text{iff} & \exists y \,.\, P_m(x,\ y) \wedge \mathcal{M}, y \models \varphi
\end{array}$$

1. Rewrite “english” in “logic”…
2. Remember the equivalence between models …
3. v and w are elements in the model, we represent them as variables…

The standard translation

$$
\begin{array}{lll}
\mathcal{M}, x \models p & \text{iff} & P_p(x) \\
\mathcal{M}, x \models \neg\varphi & \text{iff} & \neg\mathcal{M}, x \models \varphi \\
\mathcal{M}, x \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, x \models \varphi \wedge \mathcal{M}, x \models \psi \\
\mathcal{M}, x \models \langle m \rangle \varphi & \text{iff} & \exists y \,.\, P_m(x,\ y) \wedge \mathcal{M}, y \models \varphi
\end{array}
$$

1. Rewrite “english” in “logic”…
2. Remember the equivalence between models …
3. v and w are elements in the model, we represent them as variables…
4. Done!

The standard translation

$$
\begin{array}{lcl}
\mathrm{Trad}_x(p) & \equiv & P_p(x) \\
\mathrm{Trad}_x(\neg\varphi) & \equiv & \neg\mathrm{Trad}_x(\varphi) \\
\mathrm{Trad}_x(\varphi \wedge \psi) & \equiv & \mathrm{Trad}_x(\varphi) \wedge \mathrm{Trad}_x(\psi) \\
\mathrm{Trad}_x(\langle m \rangle\varphi) & \equiv & \exists y \,.\, P_m(x,\ y) \wedge \mathrm{Trad}_y(\varphi)
\end{array}
$$

1. Rewrite “english” in “logic”…
2. Remember the equivalence between models …
3. v and w are elements in the model, we represent them as variables…
4. Done!
5. This is the standard translation into FOL

The standard translation

$$
\begin{array}{lcl}
\mathsf{ST}_x(p) & \equiv & P_p(x) \\
\mathsf{ST}_x(\neg\varphi) & \equiv & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & \equiv & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\langle m \rangle\varphi) & \equiv & \exists y \,.\, P_m(x,\ y) \wedge \mathsf{ST}_y(\varphi)
\end{array}
$$

1. Rewrite “english” in “logic”…
2. Remember the equivalence between models …
3. v and w are elements in the model, we represent them as variables…
4. Done!
5. This is the standard translation into FOL

ST, the standard translation

- $\mathsf{ST}_x(\varphi)$ maps each formula φ, to a formula in FOL with exactly one free variable x
- This free variable will be instantiated by the point of evaluation of a modal formula (remember the “internal perspective)

ST, the standard translation

- $\mathsf{ST}_x(\varphi)$ maps each formula φ, to a formula in FOL with exactly one free variable x
- This free variable will be instantiated by the point of evaluation of a modal formula (remember the "internal perspective)

Theorem

For any formula φ in the basic modal logic, any model $\mathcal{M}$, any w in the domain of $\mathcal{M}$ an any assignment g,

$$\mathcal{M}, w \models \varphi \textit{ iff } \mathcal{M}, g[x \mapsto w] \models \mathit{ST}_x(\varphi)$$

ST, the standard translation

- $\mathsf{ST}_x(\varphi)$ maps each formula φ, to a formula in FOL with exactly one free variable x
- This free variable will be instantiated by the point of evaluation of a modal formula (remember the “internal perspective)

Theorem

For any formula φ in the basic modal logic, any model $\mathcal{M}$, any w in the domain of $\mathcal{M}$ an any assignment g,

$$\mathcal{M}, w \models \varphi \textit{ iff } \mathcal{M}, g[x \mapsto w] \models \mathsf{ST}_x(\varphi)$$

Proof.

Easy, by induction on φ. $\square$

Formula correspondence. From FOL to modal

- It is easy to see that ST is injective, but not surjective.
  - E.g., notice that each quantifier comes with a guard.

Formula correspondence. From FOL to modal

- It is easy to see that ST is injective, but not surjective.
  - E.g., notice that each quantifier comes with a guard.
- Can we give a similar translation from FOL to the basic modal logic?

Formula correspondence. From FOL to modal

- It is easy to see that ST is injective, but not surjective.
  - E.g., notice that each quantifier comes with a guard.
- Can we give a similar translation from FOL to the basic modal logic?
  - Let's have a look at the semantics

$$
\begin{array}{lll}
\mathcal{I}, g \models P(x_1, \ldots, x_n) & \text{iff} & (g(x_1), \ldots, g(x_n)) \in P^{\mathcal{I}} \\
\mathcal{I}, g \models \neg\varphi & \text{iff} & \mathcal{I}, g \not\models \varphi \\
\mathcal{I}, g \models \varphi \wedge \psi & \text{iff} & \mathcal{I}, g \models \varphi \text{ y } \mathcal{I}, g \models \psi \\
\mathcal{I}, g \models \exists x.\varphi & \text{iff} & \text{exists } w \text{ s.t. } \mathcal{I}, g[x \mapsto w] \models \varphi
\end{array}
$$

Formula correspondence. From FOL to modal

- It is easy to see that ST is injective, but not surjective.
  - E.g., notice that each quantifier comes with a guard.
- Can we give a similar translation from FOL to the basic modal logic?
  - Let's have a look at the semantics

$$
\begin{array}{lll}
\mathcal{I}, g \models P(x_1, \ldots, x_n) & \text{iff} & (g(x_1), \ldots, g(x_n)) \in P^{\mathcal{I}} \\
\mathcal{I}, g \models \neg\varphi & \text{iff} & \mathcal{I}, g \not\models \varphi \\
\mathcal{I}, g \models \varphi \wedge \psi & \text{iff} & \mathcal{I}, g \models \varphi \text{ y } \mathcal{I}, g \models \psi \\
\mathcal{I}, g \models \exists x.\varphi & \text{iff} & \text{exists } w \text{ s.t. } \mathcal{I}, g[x \mapsto w] \models \varphi
\end{array}
$$

  - Looks difficult, but how do we prove there is no translation.

Formula correspondence. From FOL to modal

- It is easy to see that ST is injective, but not surjective.
  - E.g., notice that each quantifier comes with a guard.
- Can we give a similar translation from FOL to the basic modal logic?
  - Let's have a look at the semantics

$$
\begin{array}{lll}
\mathcal{I}, g \models P(x_1, \ldots, x_n) & \text{iff} & (g(x_1), \ldots, g(x_n)) \in P^{\mathcal{I}} \\
\mathcal{I}, g \models \neg\varphi & \text{iff} & \mathcal{I}, g \not\models \varphi \\
\mathcal{I}, g \models \varphi \wedge \psi & \text{iff} & \mathcal{I}, g \models \varphi \text{ y } \mathcal{I}, g \models \psi \\
\mathcal{I}, g \models \exists x.\varphi & \text{iff} & \text{exists } w \text{ s.t. } \mathcal{I}, g[x \mapsto w] \models \varphi
\end{array}
$$

  - Looks difficult, but how do we prove there is no translation.
- We will see how later.

Transference results

- The ST let us import results from FOL.
- We will discuss two examples:
  1. Compactness
  2. Löwenheim-Skolem

(What is compactness? ← notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$*.*

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

- Compactness is nice because it ensures:
  - Reasoning in a compact logic always involves a finite number of premises.
  - It is a tool to prove (non constructive) model existence …
  - …and non-existence also.

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$, *then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)

$\square$

(What is compactness? ← notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)
- "Proof" of a) (assuming completeness):

□

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If $\Gamma \models \varphi$, then for some finite $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.*

b) *If every finite subset Γ_0 of Γ is satisfiable, then Γ is.*

c) *If Γ is unsatisfiable, then some finite $\Gamma_0 \subseteq \Gamma$ is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)
- "Proof" of a) (assuming completeness):
  1. $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$ (strong completeness)

$\square$

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)
- "Proof" of a) (assuming completeness):
  1. $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$ (strong completeness)
  2. But any proof of $\Gamma \vdash \varphi$ (by definition) is finite

$\square$

(What is compactness? ← notice the big parenthesis

Theorem (FOL is Compact)

a) *If $\Gamma \models \varphi$, then for some finite $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.*

b) *If every finite subset Γ_0 of Γ is satisfiable, then Γ is.*

c) *If Γ is unsatisfiable, then some finite $\Gamma_0 \subseteq \Gamma$ is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)
- "Proof" of a) (assuming completeness):
  1. $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$ (strong completeness)
  2. But any proof of $\Gamma \vdash \varphi$ (by definition) is finite
  3. Hence, only a finite set of formulas (Γ_0) from Γ appears in the proof

$\square$

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$*, then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)
- "Proof" of a) (assuming completeness):
  1. $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$ (strong completeness)
  2. But any proof of $\Gamma \vdash \varphi$ (by definition) is finite
  3. Hence, only a finite set of formulas (Γ_0) from Γ appears in the proof
  4. Hence $\Gamma_0 \vdash \varphi$ and $\Gamma_0 \models \varphi$

$\square$

(What is compactness? $\leftarrow$ notice the big parenthesis

Theorem (FOL is Compact)

a) *If* $\Gamma \models \varphi$, *then for some finite* $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \models \varphi$.

b) *If every finite subset* Γ_0 *of* Γ *is satisfiable, then* Γ *is.*

c) *If* Γ *is unsatisfiable, then some finite* $\Gamma_0 \subseteq \Gamma$ *is.*

Proof.

- a), b) y c) are equivalent (¡Exercise!)
- "Proof" of a) (assuming completeness):
  1. $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$ (strong completeness)
  2. But any proof of $\Gamma \vdash \varphi$ (by definition) is finite
  3. Hence, only a finite set of formulas (Γ_0) from Γ appears in the proof
  4. Hence $\Gamma_0 \vdash \varphi$ and $\Gamma_0 \models \varphi$
- Of course, proving strong completeness is the hard part. $\square$

Compactness in action!

- Consider the following formulas:

$$
\begin{aligned}
\mathsf{AtLeast}_2 &:= \exists x_1, x_2 \,.\, x_1 \neq x_2 \\
\mathsf{AtLeast}_3 &:= \exists x_1, x_2, x_3 \,.\, x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \\
&\vdots \\
\mathsf{AtLeast}_n &:= \exists x_1, \ldots, x_n \,.\, \bigwedge_{i \neq j} x_i \neq x_j
\end{aligned}
$$

- How should a model $\mathcal{I}$ be so that $\mathcal{I} \models \mathsf{AtLeast}_n$?

Compactness in action!

- Consider the following formulas:

$$\begin{array}{l} \mathsf{AtLeast}_2 := \exists x_1, x_2 \,.\, x_1 \neq x_2 \\ \mathsf{AtLeast}_3 := \exists x_1, x_2, x_3 \,.\, x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \\ \qquad\qquad\vdots \\ \mathsf{AtLeast}_n := \exists x_1, \ldots, x_n \,.\, \bigwedge_{i \neq j} x_i \neq x_j \end{array}$$

- How should a model $\mathcal{I}$ be so that $\mathcal{I} \models \mathsf{AtLeast}_n$?
- And $\mathcal{I} \models \mathsf{AtLeast}_n \wedge \neg\mathsf{AtLeast}_{n+1}$?

Compactness in action!

Question: Can FOL define finiteness? I.e., is there a formula φ s.t. $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ is a finite model?

Compactness in action!

Question: Can FOL define finiteness? I.e., is there a formula φ s.t. $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ is a finite model?

- It if exist, we could just show it explicitly…
- If it does not exist, how do we prove it?

Compactness in action!

Question: Can FOL define finiteness? I.e., is there a formula φ s.t. $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ is a finite model?

- It if exist, we could just show it explicitly…
- If it does not exist, how do we prove it?
  1. Suppose there is a φ as requested, and let

$$\Gamma := \{\varphi\} \cup \bigcup_{i=2}^{\infty} \{\mathsf{AtLeast}_i\}$$

Compactness in action!

Question: Can FOL define finiteness? I.e., is there a formula φ s.t. $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ is a finite model?

- It if exist, we could just show it explicitly…
- If it does not exist, how do we prove it?
  1. Suppose there is a φ as requested, and let

$$\Gamma := \{\varphi\} \cup \bigcup_{i=2}^{\infty} \{\mathsf{AtLeast}_i\}$$

  2. Every finite Γ_0, subset of Γ, is satisfiable…

Compactness in action!

Question: Can FOL define finiteness? I.e., is there a formula φ s.t. $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ is a finite model?

- It if exist, we could just show it explicitly…
- If it does not exist, how do we prove it?
  1. Suppose there is a φ as requested, and let

$$\Gamma := \{\varphi\} \cup \bigcup_{i=2}^{\infty} \{\mathsf{AtLeast}_i\}$$

  2. Every finite Γ_0, subset of Γ, is satisfiable…
  3. By compactness Γ is satisfiable, but it is not.

Compactness in action!

Question: Can FOL define finiteness? I.e., is there a formula φ s.t. $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ is a finite model?

- It if exist, we could just show it explicitly…
- If it does not exist, how do we prove it?
  1. Suppose there is a φ as requested, and let

$$\Gamma := \{\varphi\} \cup \bigcup_{i=2}^{\infty} \{\mathsf{AtLeast}_i\}$$

  2. Every finite Γ_0, subset of Γ, is satisfiable…
  3. By compactness Γ is satisfiable, but it is not.
  4. The contradiction comes from assuming the existence of φ

To think about...) $\leftarrow$ and we close the parenthesis!

- We just show that certain things cannot be expressed in FOL.

To think about. . .) $\leftarrow$ and we close the parenthesis!

- We just show that certain things cannot be expressed in FOL.
- We could move to higher-order logics. . .

To think about. . .) $\leftarrow$ and we close the parenthesis!

- We just show that certain things cannot be expressed in FOL.
- We could move to higher-order logics. . .
- . . . at the price of losing nice meta-logic properties (e.g., compacidad) which makes them hard to use.

To think about. . .) $\leftarrow$ and we close the parenthesis!

- We just show that certain things cannot be expressed in FOL.
- We could move to higher-order logics. . .
- . . . at the price of losing nice meta-logic properties (e.g., compacidad) which makes them hard to use.
- *Then, what gives?*

To think about. . .) $\leftarrow$ and we close the parenthesis!

- We just show that certain things cannot be expressed in FOL.
- We could move to higher-order logics. . .
- . . . at the price of losing nice meta-logic properties (e.g., compacidad) which makes them hard to use.
- *Then, what gives?*
- That's life. Chose your logic.

To think about. . .) $\leftarrow$ and we close the parenthesis!

- We just show that certain things cannot be expressed in FOL.
- We could move to higher-order logics. . .
- . . . at the price of losing nice meta-logic properties (e.g., compacidad) which makes them hard to use.
- *Then, what gives?*
- That's life. Chose your logic.
- There will always be compromises between expressivity, good meta-logical properties, easy of use, computational complexity, etc.

Back to BML: Compactness Transference

Theorem (Compactness for the Basic Modal Logic)

If every finite set of Γ *is satisfiable, then* Γ *is.*

Back to BML: Compactness Transference

Theorem (Compactness for the Basic Modal Logic)

If every finite set of Γ *is satisfiable, then* Γ *is.*

Proof.

1. Define, for each set of formulas Δ,

$$\mathsf{ST}_x(\Delta) := \{\mathsf{ST}_x(\varphi) \mid \varphi \in \Delta\}$$

$\square$

Back to BML: Compactness Transference

Theorem (Compactness for the Basic Modal Logic)
If every finite set of Γ *is satisfiable, then* Γ *is.*

Proof.

1. Define, for each set of formulas Δ,

$$\mathsf{ST}_x(\Delta) := \{\mathsf{ST}_x(\varphi) \mid \varphi \in \Delta\}$$

2. Let Γ_0 be a finite subset of Γ; we know that $\mathsf{ST}_x(\Gamma_0)$ is satisfiable iff Γ_0 is

$\square$

Back to BML: Compactness Transference

Theorem (Compactness for the Basic Modal Logic)

If every finite set of Γ is satisfiable, then Γ is.

Proof.

1. Define, for each set of formulas Δ,

$$\mathsf{ST}_x(\Delta) := \{\mathsf{ST}_x(\varphi) \mid \varphi \in \Delta\}$$

2. Let Γ_0 be a finite subset of Γ; we know that $\mathsf{ST}_x(\Gamma_0)$ is satisfiable iff Γ_0 is
3. Then if every finite Γ_0 is satisfiable, every $\mathsf{ST}_x(\Gamma_0)$ is; and by FOL compactness, $\mathsf{ST}_x(\Gamma)$ is satisfiable

$\square$

Back to BML: Compactness Transference

Theorem (Compactness for the Basic Modal Logic)

If every finite set of Γ *is satisfiable, then* Γ *is.*

Proof.

1. Define, for each set of formulas Δ,

$$\mathsf{ST}_x(\Delta) := \{\mathsf{ST}_x(\varphi) \mid \varphi \in \Delta\}$$

2. Let Γ_0 be a finite subset of Γ; we know that $\mathsf{ST}_x(\Gamma_0)$ is satisfiable iff Γ_0 is
3. Then if every finite Γ_0 is satisfiable, every $\mathsf{ST}_x(\Gamma_0)$ is; and by FOL compactness, $\mathsf{ST}_x(\Gamma)$ is satisfiable
4. By then Γ is satisfiable. $\square$

(Löwenheim-Skolem $\leftarrow$ another parenthesis…

Theorem (Löwenheim-Skolem)

If Γ is a satisfiable set of FOL formulas, then Γ is satisfiable in a countable model.

(Infinite cardinals for dummies)

- There are as many natural numbers as odd numbers.
- There are as many natural numbers as rationals.
- But there are more real numbers than natural numbers (Cantor diagonal)
- I.e., there are“infinites bigger than others” (sometimes it is tricky to order them)
- If C is a set then, 2^C has always strictly more elements than C.

(Infinite cardinals for dummies)

- There are as many natural numbers as odd numbers.
- There are as many natural numbers as rationals.
- But there are more real numbers than natural numbers (Cantor diagonal)
- I.e., there are"infinites bigger than others" (sometimes it is tricky to order them)
- If C is a set then, 2^C has always strictly more elements than C. Hence, there are always bigger infinites.

Löwenheim-Skolem) $\leftarrow$ and we close the last parenthesis…

Theorem (Löwenheim-Skolem)

If Γ *is a satisfiable set of FOL formulas, then* Γ *is satisfiable in a countable model.*

Löwenheim-Skolem) $\leftarrow$ and we close the last parenthesis…

Theorem (Löwenheim-Skolem)

If Γ is a satisfiable set of FOL formulas, then Γ is satisfiable in a countable model.

Corollary:

No formula of FO can define uncountable.

Modal Löwenheim-Skolem

Theorem (Löwenheim-Skolem for BML)

If Γ *is a satisfiable set of BML formulas, then* Γ *is satisfiable in a countable model.*

Modal Löwenheim-Skolem

Theorem (Löwenheim-Skolem for BML)

If Γ *is a satisfiable set of BML formulas, then* Γ *is satisfiable in a countable model.*

Proof.

We proceed as with compacity:

1. If Γ is satisfiable, then $\mathsf{ST}_x(\Gamma)$ is.
2. By Lẅenheim-Skolem for FOL, there is countable $\mathcal{I}$ and assignment g, s.t. $\mathcal{I}, g \models \mathsf{ST}_x(\Gamma)$
3. But then, $\mathcal{I}, g(x) \models \Gamma$

□

Another application of ST: Theorem proving

- A prover is a program that
  - takes an input a formula
  - upon termination, it says if the formula is valid or not
- Building provers is difficult
- Luckily there are good provers for FOL
- Using ST, we obtain “free” a prover for BML

$$\varphi \Longrightarrow \forall x.\mathsf{ST}_x(\varphi) \Longrightarrow \boxed{\text{FOL prover}} \Longrightarrow \text{Answer}$$

A better $\mathsf{ST}\ldots$

- Let us have a closer look at ST

$$
\begin{array}{lcl}
\mathsf{ST}_x(p) & \equiv & P_p(x) \\
\mathsf{ST}_x(\neg\varphi) & \equiv & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & \equiv & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\langle m\rangle\varphi) & \equiv & \exists y\,.\,P_m(x,y) \wedge \mathsf{ST}_y(\varphi)
\end{array}
$$

- In ST_x, y is a new variable.

$$
\begin{array}{lcl}
\mathsf{ST}_y(\langle m\rangle\varphi) & \equiv & \exists z\,.\,P_m(y,z) \wedge \mathsf{ST}_z(\varphi) \\
\mathsf{ST}_z(\langle m\rangle\varphi) & \equiv & \exists w\,.\,P_m(y,w) \wedge \mathsf{ST}_w(\varphi) \\
 & \vdots &
\end{array}
$$

A better ST…

- Let us have a closer look at ST

$$
\begin{array}{lcl}
\mathsf{ST}_x(p) & \equiv & P_p(x) \\
\mathsf{ST}_x(\neg\varphi) & \equiv & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & \equiv & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\langle m \rangle\varphi) & \equiv & \exists y \,.\, P_m(x, y) \wedge \mathsf{ST}_y(\varphi)
\end{array}
$$

- In ST_x, y is a new variable.

$$
\begin{array}{lcl}
\mathsf{ST}_y(\langle m \rangle\varphi) & \equiv & \exists z \,.\, P_m(y, z) \wedge \mathsf{ST}_z(\varphi) \\
\mathsf{ST}_z(\langle m \rangle\varphi) & \equiv & \exists w \,.\, P_m(y, w) \wedge \mathsf{ST}_w(\varphi) \\
 & \vdots &
\end{array}
$$

- But, x does not appear in $\mathsf{ST}_y(\varphi)$ (neither free nor bound)

A better ST…

- Let us have a closer look at ST

$$\begin{array}{lcl}
\mathsf{ST}_x(p) & \equiv & P_p(x) \\
\mathsf{ST}_x(\neg\varphi) & \equiv & \neg\mathsf{ST}_x(\varphi) \\
\mathsf{ST}_x(\varphi \wedge \psi) & \equiv & \mathsf{ST}_x(\varphi) \wedge \mathsf{ST}_x(\psi) \\
\mathsf{ST}_x(\langle m\rangle\varphi) & \equiv & \exists y \,.\, P_m(x,y) \wedge \mathsf{ST}_y(\varphi)
\end{array}$$

- In ST_x, y is a new variable.

$$\begin{array}{lcl}
\mathsf{ST}_y(\langle m\rangle\varphi) & \equiv & \exists z \,.\, P_m(y,z) \wedge \mathsf{ST}_z(\varphi) \\
\mathsf{ST}_z(\langle m\rangle\varphi) & \equiv & \exists w \,.\, P_m(y,w) \wedge \mathsf{ST}_w(\varphi) \\
 & \vdots &
\end{array}$$

- But, x does not appear in $\mathsf{ST}_y(\varphi)$ (neither free nor bound)
- We could re-use it:

$$\mathsf{ST}_y(\langle m\rangle\varphi) \quad \equiv \quad \exists x \,.\, P_m(y,x) \wedge \mathsf{ST}_x(\varphi)$$

. . . Why?

In this way ST uses only two variables. Useful?

. . . Why?

In this way ST uses only two variables. Useful?

1. First, it shows there is more than one translation. Some might be better than others.

... Why?

In this way ST uses only two variables. Useful?

1. First, it shows there is more than one translation. Some might be better than others.
2. More important, the 2-variable translation will let us transfer a decidability (actually complexity) result.

But first. . . ¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

But first…¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem
FOL is not decidable.

But first… ¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem
FOL is not decidable.

Proof.
(Idea) Given a Turing machine $\mathcal{T}$, we can write a FOL formula $\varphi_{\mathcal{T}}$ such that

- $\varphi_{\mathcal{T}}$ is satisfiable iff $\mathcal{T}$ terminates on all inputs.

(See, e.g., 'Mathematical Logic', Ebbinghaus, Flum y Thomas) □

But first…¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem
FOL is not decidable.

Just a moment! An the FOL provers?

But first. . . ¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem

FOL is not decidable.

Just a moment! An the FOL provers?

- Complete axiomatization $\Longrightarrow$ a tool to list all theorems

But first…¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem

FOL is not decidable.

Just a moment! An the FOL provers?

- Complete axiomatization $\Longrightarrow$ a tool to list all theorems
- φ is valid $\Longrightarrow$ eventually φ is listed as a theorem $\surd$

But first… ¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem

FOL is not decidable.

Just a moment! An the FOL provers?

- Complete axiomatization $\Longrightarrow$ a tool to list all theorems
- φ is valid $\Longrightarrow$ eventually φ is listed as a theorem $\surd$
- φ is a contradiction $\Longrightarrow$ eventually $\neg\varphi$ is listed $\surd$

But first. . . ¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem
FOL is not decidable.

Just a moment! An the FOL provers?

- Complete axiomatization $\Longrightarrow$ a tool to list all theorems
- φ is valid $\Longrightarrow$ eventually φ is listed as a theorem $\surd$
- φ is a contradiction $\Longrightarrow$ eventually $\neg\varphi$ is listed $\surd$
- Otherwise $\Longrightarrow$ we might have to wait forever $\times$

But first. . . ¿decidability of what?

Definition: Usually, we say that a logic is decidable if the problem of determining the validity of its formulas is decidable. (For FOL, we can interchange validity and satisfiability.)

Theorem

FOL is not decidable.

Just a moment! An the FOL provers?

- Complete axiomatization $\Longrightarrow$ a tool to list all theorems
- φ is valid $\Longrightarrow$ eventually φ is listed as a theorem $\surd$
- φ is a contradiction $\Longrightarrow$ eventually $\neg\varphi$ is listed $\surd$
- Otherwise $\Longrightarrow$ we might have to wait forever $\times$
- This kind of problems are called semi-decidible.

More transference: Decidability of BML

Theorem

The fragment defined as the set of all formulas of FOL with only two variables (FOL2) is decidable.

More transference: Decidability of BML

Theorem

The fragment defined as the set of all formulas of FOL with only two variables (FOL2) is decidable.

Proof.

Leap of faith.

More transference: Decidability of BML

Theorem

The fragment defined as the set of all formulas of FOL with only two variables (FOL2) is decidable.

Proof.

Leap of faith. The original proof is by Scott, 1962 (‘A decision method for validity of sentences in two variables’) for FO2 without equality. The result with equality is by Mortimer, 1975 (‘On languages with two variables’) □

More transference: Decidability of BML

Theorem

The fragment defined as the set of all formulas of FOL with only two variables (FOL2) is decidable.

Proof.

Leap of faith. The original proof is by Scott, 1962 ('A decision method for validity of sentences in two variables') for FO2 without equality. The result with equality is by Mortimer, 1975 ('On languages with two variables') $\square$

Theorem

BML is decidable.

More transference: Decidability of BML

Theorem

The fragment defined as the set of all formulas of FOL with only two variables (FOL2) is decidable.

Proof.

Leap of faith. The original proof is by Scott, 1962 ('A decision method for validity of sentences in two variables') for FO2 without equality. The result with equality is by Mortimer, 1975 ('On languages with two variables') $\square$

Theorem

BML is decidable.

Proof.

Given φ, we use ST using only two variables and use the decision method for FOL2 with $\forall x.\mathsf{ST}_x(\varphi)$ as input. $\square$

Decidability and expressive power

- We had a pending question:

$$\boxed{\text{Is there a translation from FOL into BML?}}$$

Decidability and expressive power

- We had a pending question:

  Is there a translation from FOL into BML?

- And now we have the answer: No.

Decidability and expressive power

- We had a pending question:

$$\boxed{\text{Is there a translation from FOL into BML?}}$$

- And now we have the answer: No.
- If such a translation exists, then FOL would be decidable.

Decidability and expressive power

- We had a pending question:

Is there a translation from FOL into BML?

- And now we have the answer: No.
- If such a translation exists, then FOL would be decidable.
- This result uses decidability to meassure expressivity

Decidability and expressive power

- We had a pending question:

Is there a translation from FOL into BML?

- And now we have the answer: No.
- If such a translation exists, then FOL would be decidable.
- This result uses decidability to meassure expressivity
- It is not “constructive”(it does not tell us what it is exactly that which cannot be expressed)

Decidability and expressive power

- We had a pending question:

Is there a translation from FOL into BML?

- And now we have the answer: No.
- If such a translation exists, then FOL would be decidable.
- This result uses decidability to meassure expressivity
- It is not "constructive"(it does not tell us what it is exactly that which cannot be expressed)
- A new question:

Is there a translation from FOL2 into BML?

Going beyond BML

- It is easy to extend ST to other modal logics and obtain similar transference results

Example

1. $\mathcal{M}, w \models \mathsf{E}\varphi$ iff exists v s.t. $\mathcal{M}, v \models \varphi$

2. $\mathcal{M}, w \models \langle m \rangle^{-1}\varphi$ iff exists v s.t. $R_m vw$ y $\mathcal{M}, v \models \varphi$

Going beyond BML

- It is easy to extend ST to other modal logics and obtain similar transference results

Example

1. $$\begin{array}{rcl} \mathcal{M}, w \models \mathsf{E}\varphi & \text{iff} & \text{exists } v \text{ s.t. } \mathcal{M}, v \models \varphi \\ \mathsf{ST}_x(\mathsf{E}\varphi) & \equiv & \exists y.\mathsf{ST}_y(\varphi) \end{array}$$

2. $$\mathcal{M}, w \models \langle m \rangle^{-1}\varphi \quad \text{iff} \quad \text{exists } v \text{ s.t. } R_m vw \text{ y } \mathcal{M}, v \models \varphi$$

Going beyond BML

- It is easy to extend ST to other modal logics and obtain similar transference results

Example

1. $$\begin{array}{rcl} \mathcal{M}, w \models \mathsf{E}\varphi & \text{iff} & \text{exists } v \text{ s.t. } \mathcal{M}, v \models \varphi \\ \mathsf{ST}_x(\mathsf{E}\varphi) & \equiv & \exists x.\mathsf{ST}_x(\varphi) \end{array}$$

2. $$\mathcal{M}, w \models \langle m \rangle^{-1}\varphi \quad \text{iff} \quad \text{exists } v \text{ s.t. } R_m vw \text{ y } \mathcal{M}, v \models \varphi$$

Going beyond BML

- It is easy to extend ST to other modal logics and obtain similar transference results

Example

1. $$\begin{array}{rcl} \mathcal{M}, w \models \mathsf{E}\varphi & \text{iff} & \text{exists } v \text{ s.t. } \mathcal{M}, v \models \varphi \\ \mathsf{ST}_x(\mathsf{E}\varphi) & \equiv & \exists x.\mathsf{ST}_x(\varphi) \\ \mathsf{ST}_y(\mathsf{E}\varphi) & \equiv & \exists y.\mathsf{ST}_y(\varphi) \end{array}$$

2. $$\mathcal{M}, w \models \langle m \rangle^{-1}\varphi \quad \text{iff} \quad \text{exists } v \text{ s.t. } R_m vw \text{ y } \mathcal{M}, v \models \varphi$$

Going beyond BML

- It is easy to extend ST to other modal logics and obtain similar transference results

Example

1. $\mathcal{M}, w \models \mathsf{E}\varphi$ iff exists v s.t. $\mathcal{M}, v \models \varphi$

$$\begin{aligned} \mathsf{ST}_x(\mathsf{E}\varphi) &\equiv \exists x.\mathsf{ST}_x(\varphi) \\ \mathsf{ST}_y(\mathsf{E}\varphi) &\equiv \exists y.\mathsf{ST}_y(\varphi) \end{aligned}$$

2. $\mathcal{M}, w \models \langle m \rangle^{-1}\varphi$ iff exists v s.t. $R_m vw$ y $\mathcal{M}, v \models \varphi$

$$\mathsf{ST}_x(\langle m \rangle^{-1}\varphi) \equiv \exists y.R_m(y, x) \wedge \mathsf{ST}_y(\varphi)$$

Going beyond BML

- It is easy to extend ST to other modal logics and obtain similar transference results

Example

1. $$\begin{array}{rcl} \mathcal{M}, w \models \mathsf{E}\varphi & \text{iff} & \text{exists } v \text{ s.t. } \mathcal{M}, v \models \varphi \\ \mathsf{ST}_x(\mathsf{E}\varphi) & \equiv & \exists x.\mathsf{ST}_x(\varphi) \\ \mathsf{ST}_y(\mathsf{E}\varphi) & \equiv & \exists y.\mathsf{ST}_y(\varphi) \end{array}$$

2. $$\begin{array}{rcl} \mathcal{M}, w \models \langle m \rangle^{-1}\varphi & \text{iff} & \text{exists } v \text{ s.t. } R_m vw \text{ y } \mathcal{M}, v \models \varphi \\ \mathsf{ST}_x(\langle m \rangle^{-1}\varphi) & \equiv & \exists y.R_m(y,x) \wedge \mathsf{ST}_y(\varphi) \\ \mathsf{ST}_y(\langle m \rangle^{-1}\varphi) & \equiv & \exists x.R_m(x,y) \wedge \mathsf{ST}_x(\varphi) \end{array}$$

Going beyond BML

Example (cont.)

3. $\mathcal{M}, w \models \langle\pi\rangle\varphi$ iff exists v s.t. $(w, v) \in \overline{\pi}$ and $\mathcal{M}, v \models \varphi$

Where

$$
\begin{array}{rcl}
\overline{a} & := & R_a \\
\overline{\pi_1 \cup \pi_2} & := & \overline{\pi_1} \cup \overline{\pi_2} \\
\overline{\pi_1 ; \pi_2} & := & \overline{\pi_1} \circ \overline{\pi_2} \\
\overline{\pi^*} & := & \overline{\pi}^*
\end{array}
$$

Going beyond BML

Example (cont.)

3. $\mathcal{M}, w \models \langle\pi\rangle\varphi$ iff exists v s.t. $(w, v) \in \overline{\pi}$ and $\mathcal{M}, v \models \varphi$

Where

$$\begin{array}{rcl} \overline{a} & := & R_a \\ \overline{\pi_1 \cup \pi_2} & := & \overline{\pi_1} \cup \overline{\pi_2} \\ \overline{\pi_1 ; \pi_2} & := & \overline{\pi_1} \circ \overline{\pi_2} \\ \overline{\pi^*} & := & \overline{\pi}^* \end{array}$$

- It would be enough to show TR s.t.

$$\mathcal{I}, g \models \mathsf{TR}_\pi(x, y) \quad \text{sii} \quad (g(x), g(y)) \in \overline{\pi}$$

- because then

$$\mathsf{ST}_x(\langle\pi\rangle\varphi) := \exists y \,.\, \mathsf{TR}_\pi(x, y) \wedge \mathsf{ST}_y(\varphi)$$

Translating PDL relations

$$
\begin{aligned}
\mathsf{TR}_a(x,y) &:= \\
\mathsf{TR}_{\pi_1 \cup \pi_2}(x,y) &:= \\
\mathsf{TR}_{\pi_1;\pi_2}(x,y) &:= \\
\mathsf{TR}_{\pi^*}(x,y) &:=
\end{aligned}
$$

Translating PDL relations

$$
\begin{array}{lcl}
\mathsf{TR}_a(x,y) & := & P_a(x,y) \\
\mathsf{TR}_{\pi_1 \cup \pi_2}(x,y) & := & \\
\mathsf{TR}_{\pi_1;\pi_2}(x,y) & := & \\
\mathsf{TR}_{\pi^*}(x,y) & := &
\end{array}
$$

Translating PDL relations

$$
\begin{array}{lll}
\mathsf{TR}_a(x,y) & := & P_a(x,y) \\
\mathsf{TR}_{\pi_1 \cup \pi_2}(x,y) & := & \mathsf{TR}_{\pi_1}(x,y) \vee \mathsf{TR}_{\pi_2}(x,y) \\
\mathsf{TR}_{\pi_1 ; \pi_2}(x,y) & := & \\
\mathsf{TR}_{\pi^*}(x,y) & := &
\end{array}
$$

Translating PDL relations

$$
\begin{array}{lcl}
\mathsf{TR}_a(x,y) & := & P_a(x,y) \\
\mathsf{TR}_{\pi_1 \cup \pi_2}(x,y) & := & \mathsf{TR}_{\pi_1}(x,y) \vee \mathsf{TR}_{\pi_2}(x,y) \\
\mathsf{TR}_{\pi_1 ; \pi_2}(x,y) & := & \exists z \,.\, \mathsf{TR}_{\pi_1}(x,z) \wedge \mathsf{TR}_{\pi_2}(z,y) \\
\mathsf{TR}_{\pi^*}(x,y) & := &
\end{array}
$$

Translating PDL relations

$$
\begin{array}{lcl}
\mathsf{TR}_a(x,y) & := & P_a(x,y) \\
\mathsf{TR}_{\pi_1 \cup \pi_2}(x,y) & := & \mathsf{TR}_{\pi_1}(x,y) \vee \mathsf{TR}_{\pi_2}(x,y) \\
\mathsf{TR}_{\pi_1 ; \pi_2}(x,y) & := & \exists z \,.\, \mathsf{TR}_{\pi_1}(x,z) \wedge \mathsf{TR}_{\pi_2}(z,y) \\
\mathsf{TR}_{\pi^*}(x,y) & := & \text{Well, not into FOL we wont\ldots}
\end{array}
$$

Transitive closure

- Let Γ be the following (infinite) set of formulas

$$\begin{array}{r} \langle\pi^*\rangle\neg p \\ p \\ [\pi]p \\ [\pi][\pi]p \\ [\pi][\pi][\pi]p \\ \vdots \end{array}$$

Transitive closure

- Let Γ be the following (infinite) set of formulas

$$\begin{array}{r} \langle\pi^*\rangle\neg p \\ p \\ [\pi]\,p \\ [\pi][\pi]\,p \\ [\pi][\pi][\pi]\,p \\ \vdots \end{array}$$

- Each finite Γ_0, subset of Γ, is satisfiable

Transitive closure

- Let Γ be the following (infinite) set of formulas

$$\begin{array}{r} \langle\pi^*\rangle\neg p \\ p \\ [\pi]\,p \\ [\pi][\pi]\,p \\ [\pi][\pi][\pi]\,p \\ \vdots \end{array}$$

- Each finite Γ_0, subset of Γ, is satisfiable
- But Γ is not

Transitive closure

- Let Γ be the following (infinite) set of formulas

$$\begin{array}{r} \langle\pi^*\rangle\neg p \\ p \\ [\pi]\,p \\ [\pi][\pi]\,p \\ [\pi][\pi][\pi]\,p \\ \vdots \end{array}$$

- Each finite Γ_0, subset of Γ, is satisfiable
- But Γ is not
- I.e., PDL is not compact

Transitive closure

- Let Γ be the following (infinite) set of formulas

$$\begin{array}{r} \langle\pi^*\rangle\neg p \\ p \\ [\pi]\,p \\ [\pi][\pi]\,p \\ [\pi][\pi][\pi]\,p \\ \vdots \end{array}$$

- Each finite Γ_0, subset of Γ, is satisfiable
- But Γ is not
- I.e., PDL is not compact
- Hence, we won't be able to obtain a translation into FOL.

Transitive closure

- Let Γ be the following (infinite) set of formulas

$$\begin{array}{r} \langle\pi^*\rangle\neg p \\ p \\ [\pi]\,p \\ [\pi][\pi]\,p \\ [\pi][\pi][\pi]\,p \\ \vdots \end{array}$$

- Each finite Γ_0, subset of Γ, is satisfiable
- But Γ is not
- I.e., PDL is not compact
- Hence, we won't be able to obtain a translation into FOL.
- Note: we just discover something concrete that cannot be expressed by FOL: the transitive closure of a relation.

To close

We saw that…

- BML is a (proper) fragment of FOL, and other modal logics are too.

To close

We saw that…

- BML is a (proper) fragment of FOL, and other modal logics are too.
- there is always a balance between expressive power, complexity, good behavior, etc.

To close

We saw that...

- BML is a (proper) fragment of FOL, and other modal logics are too.
- there is always a balance between expressive power, complexity, good behavior, etc.
- modal logics are not restricted to FOL
  - (PDL is a decidable fragment of second-order logic.)