Quantification in Dynamic Epistemic Logic
Day 5

Rustam Galimullin & Louwe B. Kuijer

ESSLLI 2023

Table of Contents

Remembering the past

- On Monday we discussed public announcements and arrow updates (among other things).

Remembering the past

- On Monday we discussed public announcements and arrow updates (among other things).
- Recall: public announcements change S, arrow updates change R, both result in simpler models.

Remembering the past

- On Monday we discussed public announcements and arrow updates (among other things).
- Recall: public announcements change S, arrow updates change R, both result in simpler models.
- Today, we consider the more powerful *action models* and *arrow update models*.

Remembering the past

- On Monday we discussed public announcements and arrow updates (among other things).
- Recall: public announcements change S, arrow updates change R, both result in simpler models.
- Today, we consider the more powerful *action models* and *arrow update models*.
- These generally increase complexity of a model.

Table of Contents

From announcements to action models

- Have you ever looked at a public announcement and wondered

From announcements to action models

- Have you ever looked at a public announcement and wondered
  "what if we could, you know,

From announcements to action models

- Have you ever looked at a public announcement and wondered *"what if we could, you know, like,*

From announcements to action models

- Have you ever looked at a public announcement and wondered
  "what if we could, you know, like, do multiple announcements at the same time, and like,

From announcements to action models

- Have you ever looked at a public announcement and wondered
  "what if we could, you know, like, do multiple announcements at the same time, and like, not tell anyone which announcement we actually did?"

From announcements to action models

- Have you ever looked at a public announcement and wondered
  "what if we could, you know, like, do multiple announcements at the same time, and like, not tell anyone which announcement we actually did?"
- Well, then it wouldn't be a public announcement anymore, now would it.

From announcements to action models

- Have you ever looked at a public announcement and wondered *"what if we could, you know, like, do multiple announcements at the same time, and like, not tell anyone which announcement we actually did?"*
- Well, then it wouldn't be a public announcement anymore, now would it.
- Instead, it would be an action model.

Events and outcomes

- Action models (a.k.a. event models) are complex events.

Events and outcomes

- Action models (a.k.a. event models) are complex events.
- A single such event can have multiple different outcomes.

Events and outcomes

- Action models (a.k.a. event models) are complex events.
- A single such event can have multiple different outcomes.
- Example: tossing a coin, with outcomes “heads” and “tails”.

Events and outcomes

- Action models (a.k.a. event models) are complex events.
- A single such event can have multiple different outcomes.
- Example: tossing a coin, with outcomes “heads” and “tails”.
- Notation: event E has outcomes $o_1, \cdots, o_n$.

Distinguishing outcomes

- Certain outcomes may be indistinguishable for some agents.

Distinguishing outcomes

- Certain outcomes may be indistinguishable for some agents.
- Example: if I toss the coin and look at it, then I can distinguish the outcomes. You in the audience, unless you have very good eyes, cannot.

Distinguishing outcomes

- Certain outcomes may be indistinguishable for some agents.
- Example: if I toss the coin and look at it, then I can distinguish the outcomes. You in the audience, unless you have very good eyes, cannot.
- This gives accessibility relations (one per agent) on the outcomes.

Preconditions

- One final thing: not every outcome is possible in every world.

Preconditions

- One final thing: not every outcome is possible in every world.
- Example, recall from Monday: me looking at the first card.

Preconditions

- One final thing: not every outcome is possible in every world.
- Example, recall from Monday: me looking at the first card.
- In action model representation: two outcomes,
    1. o_1 =“I look at the first card and it is red.”
    2. o_2 =“I look at the first card and it is black.”

Preconditions

- One final thing: not every outcome is possible in every world.
- Example, recall from Monday: me looking at the first card.
- In action model representation: two outcomes,
  1. $o_1 =$"I look at the first card and it is red."
  2. $o_2 =$"I look at the first card and it is black."
- Obviously, o_1 is only possible if the card is, in fact, red.

Preconditions

- One final thing: not every outcome is possible in every world.
- Example, recall from Monday: me looking at the first card.
- In action model representation: two outcomes,
  1. o_1 = "I look at the first card and it is red."
  2. o_2 = "I look at the first card and it is black."
- Obviously, o_1 is only possible if the card is, in fact, red.
- This means outcome o_1 has *precondition* r_1 without which it cannot occur.

Preconditions

- One final thing: not every outcome is possible in every world.
- Example, recall from Monday: me looking at the first card.
- In action model representation: two outcomes,
  1. o_1 = "I look at the first card and it is red."
  2. o_2 = "I look at the first card and it is black."
- Obviously, o_1 is only possible if the card is, in fact, red.
- This means outcome o_1 has *precondition* r_1 without which it cannot occur.
- Similarly, o_2 has precondition $\neg r_1$.

Action models

- Putting the three things together: $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ where
  - O is a set of outcomes,
  - for every $a \in A$, $\mathsf{R}_a \subseteq \mathsf{O} \times \mathsf{O}$ is an accessibility relation and
  - $\mathsf{Pre} : \mathsf{O} \to \mathcal{L}$ assigns each outcome a precondition.

Action models

- Putting the three things together: $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ where
  - O is a set of outcomes,
  - for every $a \in A$, $\mathsf{R}_a \subseteq \mathsf{O} \times \mathsf{O}$ is an accessibility relation and
  - $\mathsf{Pre} : \mathsf{O} \to \mathcal{L}$ assigns each outcome a precondition.
- It looks a lot like a model. Hence "action model".

Action models

- Putting the three things together: $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ where
  - O is a set of outcomes,
  - for every $a \in A$, $\mathsf{R}_a \subseteq \mathsf{O} \times \mathsf{O}$ is an accessibility relation and
  - $\mathsf{Pre} : \mathsf{O} \to \mathcal{L}$ assigns each outcome a precondition.
- It looks a lot like a model. Hence "action model".
- Important: the variant of actions models we consider here does *not* change basic facts.
- So atoms do not change value.

Action models

- Putting the three things together: $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ where
  - O is a set of outcomes,
  - for every $a \in A$, $\mathsf{R}_a \subseteq \mathsf{O} \times \mathsf{O}$ is an accessibility relation and
  - $\mathsf{Pre} : \mathsf{O} \to \mathcal{L}$ assigns each outcome a precondition.
- It looks a lot like a model. Hence "action model".
- Important: the variant of actions models we consider here does *not* change basic facts.
- So atoms do not change value.
- Variant that does change basic facts exists ("postconditions").

Action models

- Putting the three things together: $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ where
  - O is a set of outcomes,
  - for every $a \in A$, $\mathsf{R}_a \subseteq \mathsf{O} \times \mathsf{O}$ is an accessibility relation and
  - $\mathsf{Pre} : \mathsf{O} \to \mathcal{L}$ assigns each outcome a precondition.
- It looks a lot like a model. Hence "action model".
- Important: the variant of actions models we consider here does *not* change basic facts.
- So atoms do not change value.
- Variant that does change basic facts exists ("postconditions").
- Note the different font to distinguish from normal models.

Action models: Example 1

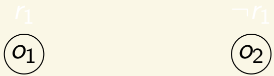
- Example from before: I look at the color of the first card.

Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).

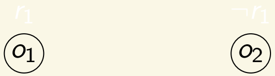
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).



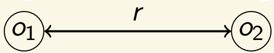
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.



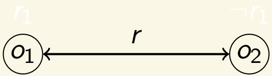
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.
- I can distinguish.



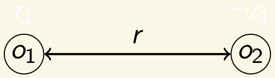
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.
- I can distinguish.
- (Note: reflexive arrows not drawn for clarity.)



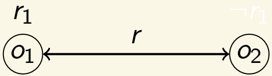
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.
- I can distinguish.
- (Note: reflexive arrows not drawn for clarity.)
- Precondition of o_1: the first card being red.



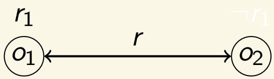
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.
- I can distinguish.
- (Note: reflexive arrows not drawn for clarity.)
- Precondition of o_1: the first card being red.



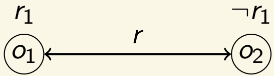
Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.
- I can distinguish.
- (Note: reflexive arrows not drawn for clarity.)
- Precondition of o_1: the first card being red.
- Precondition of o_2: the first card being black.



Action models: Example 1

- Example from before: I look at the color of the first card.
- Two outcomes: I look at a red card (o_1) or I look at a black card (o_2).
- Rustam cannot distinguish between the outcomes.
- I can distinguish.
- (Note: reflexive arrows not drawn for clarity.)
- Precondition of o_1: the first card being red.
- Precondition of o_2: the first card being black.



Action models: Example 2

- First, I privately announce either φ or ψ to Alice.
- Then, I privately tell Bob either (i) what I told Alice or (ii) nothing.

Action models: Example 2

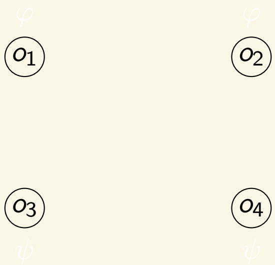
- First, I privately announce either φ or ψ to Alice.
- Then, I privately tell Bob either (i) what I told Alice or (ii) nothing.
- 4 outcomes:

$$
\begin{aligned}
o_1 &: \text{announce } \varphi\text{, don't tell Bob.}\\
o_2 &: \text{announce } \varphi\text{, tell Bob.}\\
o_3 &: \text{announce } \psi\text{, don't tell Bob.}\\
o_4 &: \text{announce } \psi\text{, tell Bob.}
\end{aligned}
$$

Action models: Example 2

- First, I privately announce either φ or ψ to Alice.
- Then, I privately tell Bob either (i) what I told Alice or (ii) nothing.
- 4 outcomes:

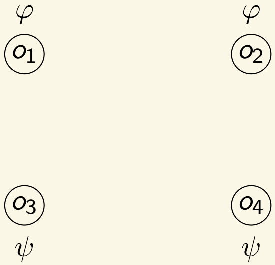
o_1 : announce φ, don't tell Bob.
o_2 : announce φ, tell Bob.
o_3 : announce ψ, don't tell Bob.
o_4 : announce ψ, tell Bob.



Action models: Example 2

- First, I privately announce either φ or ψ to Alice.
- Then, I privately tell Bob either (i) what I told Alice or (ii) nothing.
- 4 outcomes:

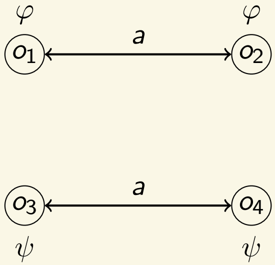
o_1 : announce φ, don't tell Bob.
o_2 : announce φ, tell Bob.
o_3 : announce ψ, don't tell Bob.
o_4 : announce ψ, tell Bob.



Action models: Example 2

- First, I privately announce either φ or ψ to Alice.
- Then, I privately tell Bob either (i) what I told Alice or (ii) nothing.
- 4 outcomes:

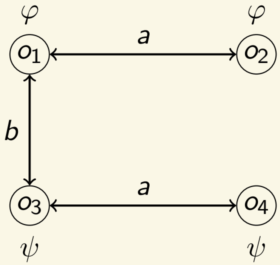
o_1 : announce φ, don't tell Bob.
o_2 : announce φ, tell Bob.
o_3 : announce ψ, don't tell Bob.
o_4 : announce ψ, tell Bob.



Action models: Example 2

- First, I privately announce either φ or ψ to Alice.
- Then, I privately tell Bob either (i) what I told Alice or (ii) nothing.
- 4 outcomes:

o_1 : announce φ, don't tell Bob.
o_2 : announce φ, tell Bob.
o_3 : announce ψ, don't tell Bob.
o_4 : announce ψ, tell Bob.



Effects of Action Models

- If we apply action model $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ to a model $M = (S, R, V)$, what do we get?

Effects of Action Models

- If we apply action model $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ to a model $M = (S, R, V)$, what do we get?
- First note: properties of a world in the new model depend on
  1. Properties of the corresponding worlds in the old model
  2. Which outcome occurred.

Effects of Action Models

- If we apply action model $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ to a model $M = (S, R, V)$, what do we get?
- First note: properties of a world in the new model depend on
  1. Properties of the corresponding worlds in the old model
  2. Which outcome occurred.
- Hence: new worlds are of the form (s, o).

Effects of Action Models

- If we apply action model $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ to a model $M = (S, R, V)$, what do we get?
- First note: properties of a world in the new model depend on
  1. Properties of the corresponding worlds in the old model
  2. Which outcome occurred.
- Hence: new worlds are of the form (s, o).
- But: not every world s is compatible with every outcome o.

Effects of Action Models

- If we apply action model $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ to a model $M = (S, R, V)$, what do we get?
- First note: properties of a world in the new model depend on
  1. Properties of the corresponding worlds in the old model
  2. Which outcome occurred.
- Hence: new worlds are of the form (s, o).
- But: not every world s is compatible with every outcome o.
- A pair (s, o) only results in a world in the new model if $M, s \models \mathsf{Pre}(o)$.

Effects of Action Models

- If we apply action model $\mathsf{E} = (\mathsf{O}, \mathsf{R}, \mathsf{Pre})$ to a model $M = (S, R, V)$, what do we get?
- First note: properties of a world in the new model depend on
  1. Properties of the corresponding worlds in the old model
  2. Which outcome occurred.
- Hence: new worlds are of the form (s, o).
- But: not every world s is compatible with every outcome o.
- A pair (s, o) only results in a world in the new model if $M, s \models \mathsf{Pre}(o)$.
- So: new set of worlds given by

$$W * \mathsf{E} = \{(s, o) \in S \times \mathsf{O} \mid M, s \models \mathsf{Pre}(o)\}.$$

Effects of Action Models (II)

- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.

Effects of Action Models (II)

- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.
- When can a distinguish between (s_1, o_1) and (s_2, o_2)?

Effects of Action Models (II)

- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.
- When can a distinguish between (s_1, o_1) and (s_2, o_2)?
- Two possibilities:

Effects of Action Models (II)

- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.
- When can a distinguish between (s_1, o_1) and (s_2, o_2)?
- Two possibilities:
  1. a could already distinguish between s_1 and s_2 before E happened,

Effects of Action Models (II)

- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.
- When can a distinguish between (s_1, o_1) and (s_2, o_2)?
- Two possibilities:
  1. a could already distinguish between s_1 and s_2 before E happened,
  2. a could tell the difference between o_1 and o_2.

Effects of Action Models (II)

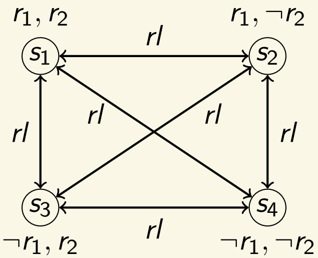
- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.
- When can a distinguish between (s_1, o_1) and (s_2, o_2)?
- Two possibilities:
  1. a could already distinguish between s_1 and s_2 before E happened,
  2. a could tell the difference between o_1 and o_2.
- Therefore: $(s_1, o_1) R * \mathsf{E}_a (s_2, o_2)$ iff $s_1 R_a s_2$ and $o_1 \mathsf{R}_a o_2$.

Effects of Action Models (II)

- Left to do: determine $R * \mathsf{E}$ and $V * \mathsf{E}$.
- When can a distinguish between (s_1, o_1) and (s_2, o_2)?
- Two possibilities:
  1. a could already distinguish between s_1 and s_2 before E happened,
  2. a could tell the difference between o_1 and o_2.
- Therefore: $(s_1, o_1) R * \mathsf{E}_a (s_2, o_2)$ iff $s_1 R_a s_2$ and $o_1 \mathsf{R}_a o_2$.
- Finally, valuation doesn't change: $(s, o) \in V * \mathsf{E}(p)$ iff $s \in V(p)$.

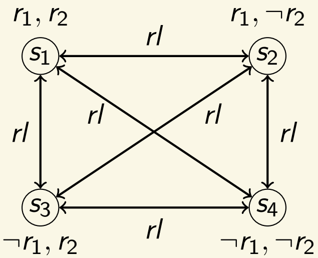
Example

- We return to our good old friend: me looking at the color of the first card.



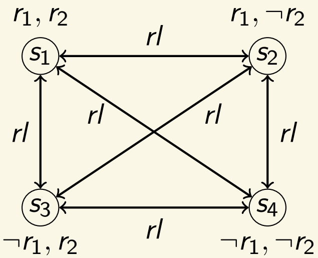
Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top



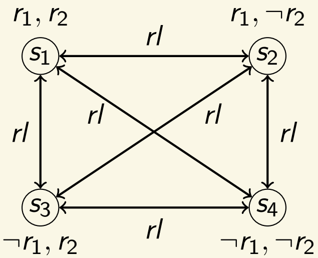
Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top



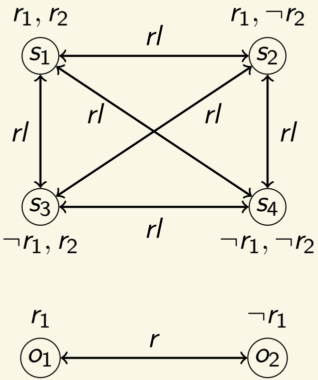
Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.



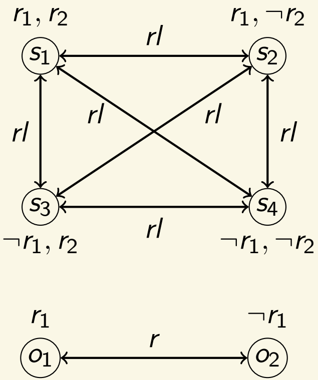
Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.



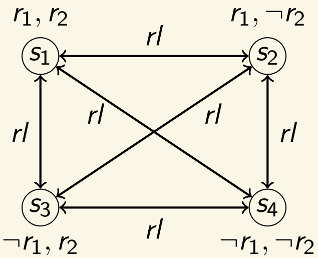
Example

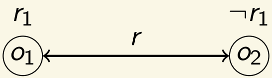
- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.
- $M * \mathsf{E}$ is constructed on the right.

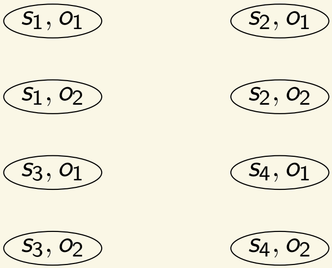


Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.
- $M * \mathsf{E}$ is constructed on the right.

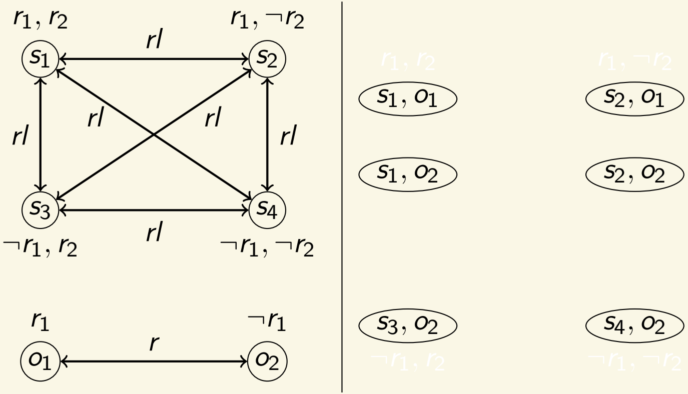






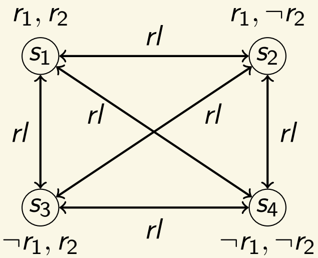
Example

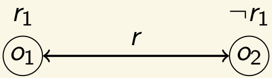
- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.
- $M * \mathsf{E}$ is constructed on the right.

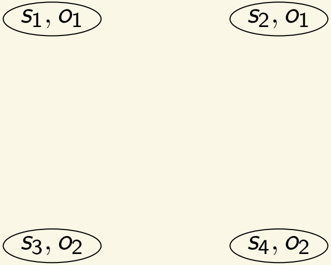


Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.
- $M * \mathrm{E}$ is constructed on the right.

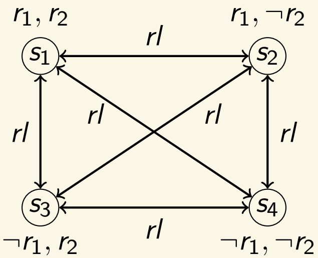


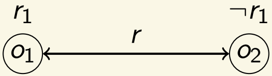


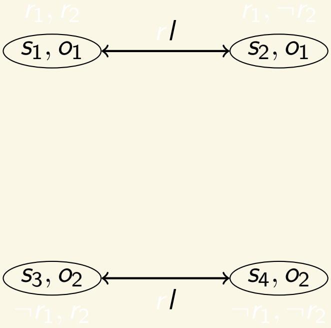


Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.
- $M * \mathrm{E}$ is constructed on the right.

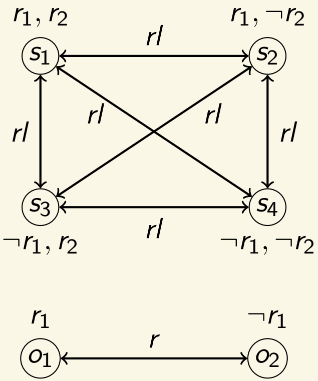






Example

- We return to our good old friend: me looking at the color of the first card.
- Recall: M is as shown on top and E as shown below.
- $M * \mathsf{E}$ is constructed on the right.



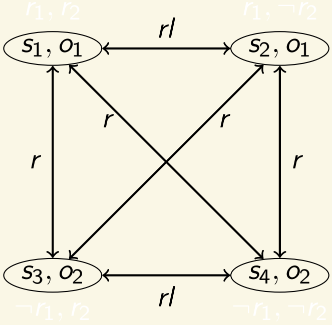


Table of Contents

Generalizing arrow updates

- Action models generalize public announcements.

Generalizing arrow updates

- Action models generalize public announcements.
- Arrow update models similarly generalize arrow updates.

Generalizing arrow updates

- Action models generalize public announcements.
- Arrow update models similarly generalize arrow updates.
- ''"Action models are to public announcements as arrow update models are to arrow updates."

Introducing Arrow Update Models

- Like action models, arrow update models have a set of outcomes, and relations between outcomes.

Introducing Arrow Update Models

- Like action models, arrow update models have a set of outcomes, and relations between outcomes.
- But: where action models place preconditions on the outcomes, arrow update models place conditions on the relations.

Introducing Arrow Update Models

- Like action models, arrow update models have a set of outcomes, and relations between outcomes.
- But: where action models place preconditions on the outcomes, arrow update models place conditions on the relations.
- $\mathsf{U} = (\mathsf{O}, \mathsf{R})$
  - O is a set of outcomes,
  - R is a set of arrow conditions of the form $(o_1, \varphi) \overset{a}{\mapsto} (o_2, \psi)$.

Introducing Arrow Update Models

- Like action models, arrow update models have a set of outcomes, and relations between outcomes.
- But: where action models place preconditions on the outcomes, arrow update models place conditions on the relations.
- $\mathsf{U} = (\mathsf{O}, \mathsf{R})$
  - O is a set of outcomes,
  - R is a set of arrow conditions of the form $(o_1, \varphi) \overset{a}{\mapsto} (o_2, \psi)$.
- $(o_1, \varphi) \overset{a}{\mapsto} (o_2, \psi)$ is read as "if φ is true in s_1 and ψ is true in s_2, then o_1 happening in s_1 is indistinguishable from o_2 happening in s_2 for a."

Effects of arrow update models

- As with action models: new worlds are of the form (s, o).

Effects of arrow update models

- As with action models: new worlds are of the form (s, o).
- But in this case: no conditions on outcomes, so $S * \mathsf{U} = S \times \mathsf{O}$.

Effects of arrow update models

- As with action models: new worlds are of the form (s, o).
- But in this case: no conditions on outcomes, so $S * \mathsf{U} = S \times \mathsf{O}$.
- $(s_1, o_1) R * \mathsf{U}_a (s_2, o_2)$ iff
  1. $s_1 R_a s_2$ and
  2. $\exists (o_1, \varphi) \overset{a}{\mapsto} (o_2, \psi) \in \mathsf{R}$ s.t. $M, s_1 \models \varphi$ and $M, s_2 \models \psi$.

Example

- Example: same old "looking at the first card" example.

Example

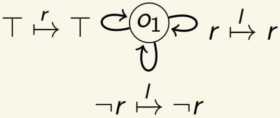
- Example: same old "looking at the first card" example.
- In this case, "I look at a red card" and "I look at a black card" are represented by same outcome o_1.

Example

- Example: same old "looking at the first card" example.
- In this case, "I look at a red card" and "I look at a black card" are represented by same outcome o_1.
- Instead, reflexive arrows on o_1 with different labels represent distinguishability.

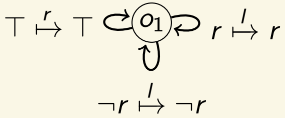
Example

- Example: same old "looking at the first card" example.
- In this case, "I look at a red card" and "I look at a black card" are represented by same outcome o_1.
- Instead, reflexive arrows on o_1 with different labels represent distinguishability.



Example

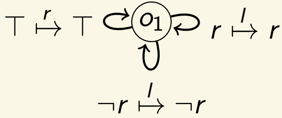
- Example: same old "looking at the first card" example.
- In this case, "I look at a red card" and "I look at a black card" are represented by same outcome o_1.
- Instead, reflexive arrows on o_1 with different labels represent distinguishability.



- Resulting model after update:

Example

- Example: same old "looking at the first card" example.
- In this case, "I look at a red card" and "I look at a black card" are represented by same outcome o_1.
- Instead, reflexive arrows on o_1 with different labels represent distinguishability.



- Resulting model after update:

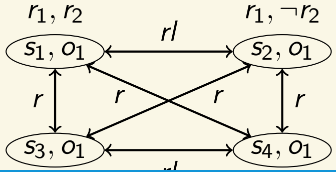


Table of Contents

One final thing

- We now have most of what we need to define Action Model Logic (AML) and Arrow Update Model Logic (AUML).

One final thing

- We now have most of what we need to define Action Model Logic (AML) and Arrow Update Model Logic (AUML).
- But: one final thing to consider.

One final thing

- We now have most of what we need to define Action Model Logic (AML) and Arrow Update Model Logic (AUML).
- But: one final thing to consider.
- Outcomes o_1 and o_2 of E may have very different results.

One final thing

- We now have most of what we need to define Action Model Logic (AML) and Arrow Update Model Logic (AUML).
- But: one final thing to consider.
- Outcomes o_1 and o_2 of E may have very different results.
- Therefore: we specify *pointed* action models/arrow update models.

One final thing

- We now have most of what we need to define Action Model Logic (AML) and Arrow Update Model Logic (AUML).
- But: one final thing to consider.
- Outcomes o_1 and o_2 of E may have very different results.
- Therefore: we specify *pointed* action models/arrow update models.
- (E, o) or (U, o).

One final thing

- We now have most of what we need to define Action Model Logic (AML) and Arrow Update Model Logic (AUML).
- But: one final thing to consider.
- Outcomes o_1 and o_2 of E may have very different results.
- Therefore: we specify *pointed* action models/arrow update models.
- (E, o) or (U, o).
- Compare: pointed models in epistemic logic.

Single/multi-pointed

- One important choice: do we use single-pointed or multi-pointed events?

Single/multi-pointed

- One important choice: do we use single-pointed or multi-pointed events?
- (E, o) or (E, X), where X is a set of outcomes?
- (U, o) or (U, X)?

Single/multi-pointed

- One important choice: do we use single-pointed or multi-pointed events?
- (E, o) or (E, X), where X is a set of outcomes?
- (U, o) or (U, X)?
- In some sense it doesn't matter: $[\mathsf{E}, X]\varphi$ can be seen as abbreviation for $\bigwedge_{o \in X}[\mathsf{E}, o]\varphi$.

Single/multi-pointed

- One important choice: do we use single-pointed or multi-pointed events?
- (E, o) or (E, X), where X is a set of outcomes?
- (U, o) or (U, X)?
- In some sense it doesn't matter: $[\mathsf{E}, X]\varphi$ can be seen as abbreviation for $\bigwedge_{o \in X}[\mathsf{E}, o]\varphi$.
- But: it matters for update expressivity.

Single/multi-pointed

- One important choice: do we use single-pointed or multi-pointed events?
- (E, o) or (E, X), where X is a set of outcomes?
- (U, o) or (U, X)?
- In some sense it doesn't matter: $[\mathsf{E}, X]\varphi$ can be seen as abbreviation for $\bigwedge_{o \in X}[\mathsf{E}, o]\varphi$.
- But: it matters for update expressivity.
- We will return to this point later.

Single/multi-pointed

- One important choice: do we use single-pointed or multi-pointed events?
- (E, o) or (E, X), where X is a set of outcomes?
- (U, o) or (U, X)?
- In some sense it doesn't matter: $[\mathsf{E}, X]\varphi$ can be seen as abbreviation for $\bigwedge_{o \in X}[\mathsf{E}, o]\varphi$.
- But: it matters for update expressivity.
- We will return to this point later.
- For now: use multi-pointed models.

Languages

Definition

The language of action model logic (AML) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\mathsf{E}, X]\varphi,$$

where $a \in A, p \in P$ and $[\mathsf{E}, X]$ is a finite multi-pointed action model.

Languages

Definition

The language of action model logic (AML) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\mathsf{E}, X]\varphi,$$

where $a \in A$, $p \in P$ and $[\mathsf{E}, X]$ is a finite multi-pointed action model.

Definition

The language of arrow update model logic (AUML) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\mathsf{U}, X]\varphi,$$

where $a \in A$, $p \in P$ and $[\mathsf{U}, X]$ is a finite multi-pointed arrow update model.

Semantics

Satisfaction relation $\models$ is extended with

Semantics

Satisfaction relation $\models$ is extended with

- $M, s \models [\mathsf{E}, X]\varphi$ iff for every $o \in X$, if $M * \mathsf{E}, (s, o)$ exists then $M * \mathsf{E}, (s, o) \models \varphi$.

Semantics

Satisfaction relation $\models$ is extended with

- $M, s \models [\mathsf{E}, X]\varphi$ iff for every $o \in X$, if $M * \mathsf{E}, (s, o)$ exists then $M * \mathsf{E}, (s, o) \models \varphi$.
- $M, s \models [\mathsf{U}, X]\varphi$ iff for every $o \in X$, $M * \mathsf{U}, (s, o) \models \varphi$.

Expressivity

- As with PAL/AUL: reduction axioms exist.

Expressivity

- As with PAL/AUL: reduction axioms exist.
- Therefore: AML, AUML, PAL, AUL, EL all have same expressive power.

Expressivity

- As with PAL/AUL: reduction axioms exist.
- Therefore: AML, AUML, PAL, AUL, EL all have same expressive power.
- We first consider the easy axioms for AML and AUML.

Reduction axioms – reducing to single pointed

- Remember: updates can be seen as model transformers.

Reduction axioms – reducing to single pointed

- Remember: updates can be seen as model transformers.
- But: $[\mathsf{E}, X]$ and $[\mathsf{U}, X]$ are not functions, since they are on-on-many.

Reduction axioms – reducing to single pointed

- Remember: updates can be seen as model transformers.
- But: $[\mathsf{E}, X]$ and $[\mathsf{U}, X]$ are not functions, since they are on-on-many.
- First step therefore to reduce multi-pointed to single-pointed.

Reduction axioms – reducing to single pointed

- Remember: updates can be seen as model transformers.
- But: $[\mathsf{E}, X]$ and $[\mathsf{U}, X]$ are not functions, since they are on-on-many.
- First step therefore to reduce multi-pointed to single-pointed.
-
  - $[\mathsf{E}, X]\varphi \leftrightarrow \bigwedge_{o \in X}[\mathsf{E}, o]\varphi$
  - $[\mathsf{U}, X]\varphi \leftrightarrow \bigwedge_{o \in X}[\mathsf{E}, o]\varphi$

Reduction axioms – the easy ones

- Now, it suffices to consider single pointed models.
- And those are functions ($[\mathsf{U}, o]$) or partial functions ($[\mathsf{E}, o]$).

Reduction axioms – the easy ones

- Now, it suffices to consider single pointed models.
- And those are functions ($[\mathsf{U}, o]$) or partial functions ($[\mathsf{E}, o]$).
- So, same as with PAL/AUL, we have the following axioms:

$$[\mathsf{E}, o]p \leftrightarrow (\mathsf{Pre}(o) \rightarrow p)$$
$$[\mathsf{E}, o]\neg\varphi \leftrightarrow (\mathsf{Pre}(o) \rightarrow \neg[\mathsf{E}, o]\varphi)$$
$$[\mathsf{E}, o](\varphi \vee \psi) \leftrightarrow ([\mathsf{E}, o]\varphi \vee [\mathsf{E}, o]\psi)$$

$$[\mathsf{U}, o]p \leftrightarrow p$$
$$[\mathsf{U}, o]\neg\varphi \leftrightarrow \neg[\mathsf{U}, o]\varphi$$
$$[\mathsf{U}, o](\varphi \vee \psi) \leftrightarrow ([\mathsf{U}, o]\varphi \vee [\mathsf{U}, o]\psi)$$

Reduction axioms – the easy ones

- Now, it suffices to consider single pointed models.
- And those are functions ($[\mathsf{U}, o]$) or partial functions ($[\mathsf{E}, o]$).
- So, same as with PAL/AUL, we have the following axioms:

$$[\mathsf{E}, o]p \leftrightarrow (\mathsf{Pre}(o) \to p)$$
$$[\mathsf{E}, o]\neg\varphi \leftrightarrow (\mathsf{Pre}(o) \to \neg[\mathsf{E}, o]\varphi)$$
$$[\mathsf{E}, o](\varphi \vee \psi) \leftrightarrow ([\mathsf{E}, o]\varphi \vee [\mathsf{E}, o]\psi)$$

$$[\mathsf{U}, o]p \leftrightarrow p$$
$$[\mathsf{U}, o]\neg\varphi \leftrightarrow \neg[\mathsf{U}, o]\varphi$$
$$[\mathsf{U}, o](\varphi \vee \psi) \leftrightarrow ([\mathsf{U}, o]\varphi \vee [\mathsf{U}, o]\psi)$$

- Left to do: reduction axioms for $[\mathsf{E}, o]\Box_a\varphi$ and $[\mathsf{U}, o]\Box_a\varphi$.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \rightarrow \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \rightarrow \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a \varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a\varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a(\mathsf{Pre}(o') \to [\mathsf{E}, o']\varphi))$$

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a\varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a(\mathsf{Pre}(o') \to [\mathsf{E}, o']\varphi))$$

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a\varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a(\mathsf{Pre}(o') \to [\mathsf{E}, o']\varphi))$$

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a\varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a(\mathsf{Pre}(o') \to [\mathsf{E}, o']\varphi))$$

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a \varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a \varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a (\mathsf{Pre}(o') \to [\mathsf{E}, o']\varphi))$$

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a\varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a(\mathsf{Pre}(o') \to [\mathsf{E}, o']\varphi))$$

The final axiom for AML

- We need to characterize when $M, s \models [\mathsf{E}, o]\Box_a\varphi$.
- It is true in two cases:
  1. if $M * E, (s, o)$ doesn't exist or
  2. if every a-successor of (s, o) in $M * \mathsf{E}$ satisfies φ.
- Case 1 is dealt with by starting with $\mathsf{Pre}(o) \to \cdots$.
- For case 2: need to check successors of (s, o) in $M * \mathsf{E}$.
- What are those successors?
  - (s', o') with $M, s' \models \mathsf{Pre}(o')$,
  - s' is an a-successor of s in M,
  - o' is an a-successor of o in E.
- In (s', o') we must have φ. So in s' we must have $[\mathsf{E}, o']\varphi$.
- Putting it all together:

$$[\mathsf{E}, o]\Box_a\varphi \leftrightarrow (\mathsf{Pre}(o) \to \bigwedge_{o' \in \mathsf{R}_a(o)} \Box_a[\mathsf{E}, o']\varphi)$$

The final axiom for AUML

- Final axiom for AUML is constructed similarly.

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
    - s' is an a-successor of s, and

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
    - s' is an a-successor of s, and
    - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathrm{R}$ such that
  - $M, s \models \psi$ and

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathrm{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.
- If those conditions are satisfied, we must have $M, s' \models [\mathsf{U}, o']\varphi$.

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.
- If those conditions are satisfied, we must have $M, s' \models [\mathsf{U}, o']\varphi$.
- Putting it all together:

$$[\mathsf{U}, o]\Box_a\varphi \leftrightarrow \bigwedge_{(o,\psi)\overset{a}{\mapsto}(o',\chi)} (\psi \to \Box_a(\chi \to [\mathsf{U}, o]\varphi))$$

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.
- If those conditions are satisfied, we must have $M, s' \models [\mathsf{U}, o']\varphi$.
- Putting it all together:

$$[\mathsf{U}, o]\Box_a\varphi \leftrightarrow \bigwedge_{(o,\psi)\overset{a}{\mapsto}(o',\chi)} (\psi \to \Box_a(\chi \to [\mathsf{U}, o]\varphi))$$

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.
- If those conditions are satisfied, we must have $M, s' \models [\mathsf{U}, o']\varphi$.
- Putting it all together:

$$[\mathsf{U}, o]\Box_a\varphi \leftrightarrow \bigwedge_{(o,\psi)\overset{a}{\mapsto}(o',\chi)} (\psi \rightarrow \Box_a(\chi \rightarrow [\mathsf{U}, o]\varphi))$$

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.
- If those conditions are satisfied, we must have $M, s' \models [\mathsf{U}, o']\varphi$.
- Putting it all together:

$$[\mathsf{U}, o]\Box_a\varphi \leftrightarrow \bigwedge_{(o,\psi)\overset{a}{\mapsto}(o',\chi)} (\psi \to \Box_a(\chi \to [\mathsf{U}, o]\varphi))$$

The final axiom for AUML

- Final axiom for AUML is constructed similarly.
- (s', o') is an a-successor of (s, o) iff
  - s' is an a-successor of s, and
  - there is $(o, \psi) \overset{a}{\mapsto} (o', \chi) \in \mathsf{R}$ such that
  - $M, s \models \psi$ and
  - $M, s' \models \chi$.
- If those conditions are satisfied, we must have $M, s' \models [\mathsf{U}, o']\varphi$.
- Putting it all together:

$$[\mathsf{U}, o]\Box_a\varphi \leftrightarrow \bigwedge_{(o,\psi)\overset{a}{\mapsto}(o',\chi)} (\psi \rightarrow \Box_a(\chi \rightarrow [\mathsf{U}, o]\varphi))$$

A side note

- Note that no one expects you to know these axioms by heart.

A side note

- Note that no one expects you to know these axioms by heart.
- It is more than sufficient to (1) know that they exist and (2) know more or less how to derive them.

Reduction

- Remember: reduction axioms give us free
  - completeness
  - expressivity results
  - decidability

Reduction

- Remember: reduction axioms give us free
  - completeness
  - expressivity results
  - decidability (but computationally expensive)

Update expressivity

- Recall definition of update expressivity:

Definition

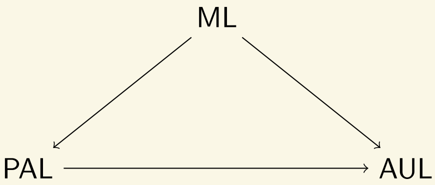
Let $e_1 : \mathfrak{M} \to \mathfrak{M}$ and $e_2 : \mathfrak{M} \to \mathfrak{M}$ be given. We say that e_2 dominates e_1, denoted $e_1 \rightsquigarrow e_2$ if for all M, s, if $e_1(M, s)$ exists, then $e_2(M, s)$ exists and the two pointed models are bisimilar.

Definition

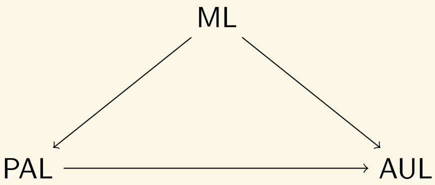
Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets $\mathcal{E}_1$ and $\mathcal{E}_2$ of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$, denoted $\mathcal{L}_1 \preceq \mathcal{L}_2$ if:

For every $e_1 \in \mathcal{E}_1$ there is an $e_2 \in \mathcal{E}_2$ such that $e_1 \rightsquigarrow e_2$.

Update expressivity, single pointed



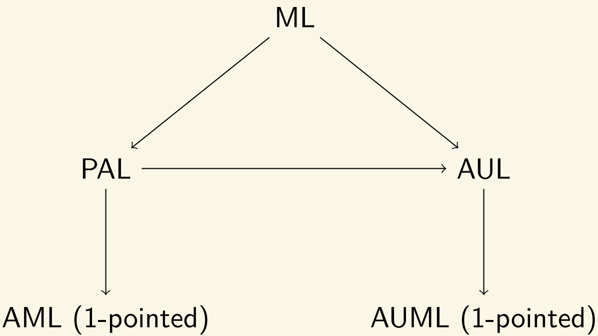
Update expressivity, single pointed



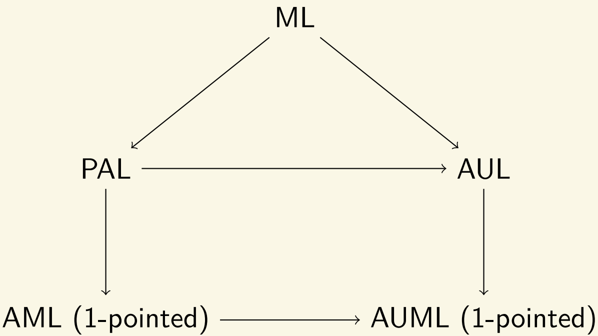
AML (1-pointed)

AUML (1-pointed)

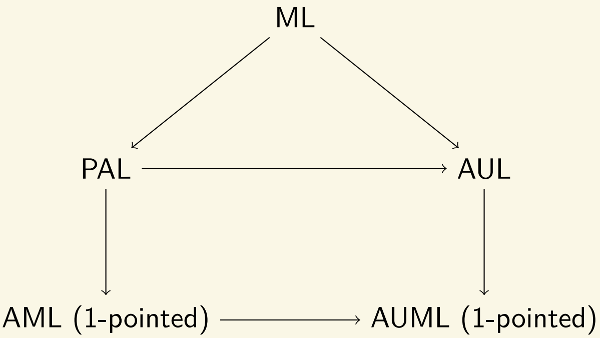
Update expressivity, single pointed



Update expressivity, single pointed



Update expressivity, single pointed



(Transitive arrows omitted)

Adapting the definition

- Previously used definition of update expressivity does not apply for one-on-many relations.
- So we can't use it to compare multi-pointed action models/arrow update models.

Adapting the definition

- Previously used definition of update expressivity does not apply for one-on-many relations.
- So we can't use it to compare multi-pointed action models/arrow update models.
- But: we can generalize the definition.

Adapting the definition

- Previously used definition of update expressivity does not apply for one-on-many relations.
- So we can't use it to compare multi-pointed action models/arrow update models.
- But: we can generalize the definition.

Definition

Let $e_1 \subseteq \mathfrak{M} \times \mathfrak{M}$ and $e_2 \subseteq \mathfrak{M} \times \mathfrak{M}$ be given. We say that e_2 dominates e_1, denoted $e_1 \rightsquigarrow e_2$ if

Adapting the definition

- Previously used definition of update expressivity does not apply for one-on-many relations.
- So we can't use it to compare multi-pointed action models/arrow update models.
- But: we can generalize the definition.

Definition

Let $e_1 \subseteq \mathfrak{M} \times \mathfrak{M}$ and $e_2 \subseteq \mathfrak{M} \times \mathfrak{M}$ be given. We say that e_2 dominates e_1, denoted $e_1 \rightsquigarrow e_2$ if for every M, s such that $e_1(M, s) \neq \varnothing$:

Adapting the definition

- Previously used definition of update expressivity does not apply for one-on-many relations.
- So we can't use it to compare multi-pointed action models/arrow update models.
- But: we can generalize the definition.

Definition

Let $e_1 \subseteq \mathfrak{M} \times \mathfrak{M}$ and $e_2 \subseteq \mathfrak{M} \times \mathfrak{M}$ be given. We say that e_2 dominates e_1, denoted $e_1 \rightsquigarrow e_2$ if for every M, s such that $e_1(M, s) \neq \varnothing$:

- for every $(M_1, s_1) \in e_1(M, s)$ there is a bisimilar $(Model_2, s_2) \in e_2(M, s)$ and

Adapting the definition

- Previously used definition of update expressivity does not apply for one-on-many relations.
- So we can't use it to compare multi-pointed action models/arrow update models.
- But: we can generalize the definition.

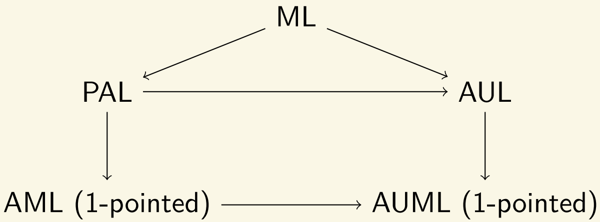
Definition

Let $e_1 \subseteq \mathfrak{M} \times \mathfrak{M}$ and $e_2 \subseteq \mathfrak{M} \times \mathfrak{M}$ be given. We say that e_2 dominates e_1, denoted $e_1 \rightsquigarrow e_2$ if for every M, s such that $e_1(M, s) \neq \varnothing$:

- for every $(M_1, s_1) \in e_1(M, s)$ there is a bisimilar $(Model_2, s_2) \in e_2(M, s)$ and
- for every $(M_2, s_2) \in e_2(M, s)$ there is a bisimilar $(M_1, s_1) \in e_1(M, s)$.

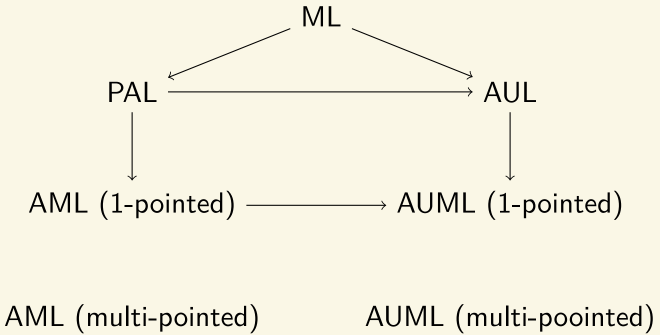
Update expressivity, multi pointed

- Technical details not very important. (Included only for completeness.)
- What is important: general view of how poweful different updates are.



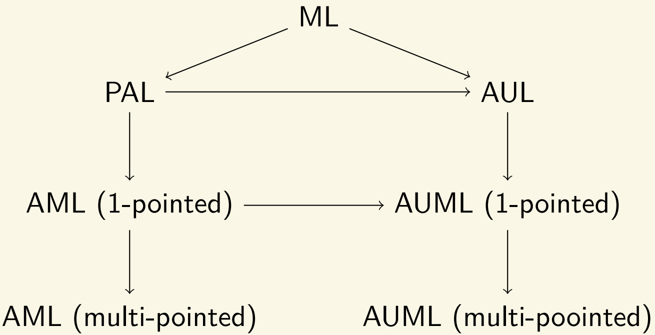
Update expressivity, multi pointed

- Technical details not very important. (Included only for completeness.)
- What is important: general view of how poweful different updates are.



Update expressivity, multi pointed

- Technical details not very important. (Included only for completeness.)
- What is important: general view of how poweful different updates are.



Update expressivity, multi pointed

- Technical details not very important. (Included only for completeness.)
- What is important: general view of how poweful different updates are.

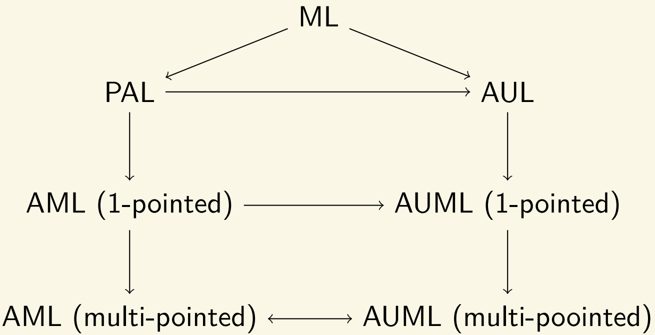


Table of Contents

AAML and AAUML

- PAL was extended to APAL (with quantifier $[!]$).
- AUL was extended to AAUL (with quantifier $[\updownarrow]$).

AAML and AAUML

- PAL was extended to APAL (with quantifier $[!]$).
- AUL was extended to AAUL (with quantifier $[\updownarrow]$).
- Similarly, AML can be extended to AAML with a quantifier $[\otimes]$ (dual: $\langle\otimes\rangle$).

AAML and AAUML

- PAL was extended to APAL (with quantifier $[!]$).
- AUL was extended to AAUL (with quantifier $[\updownarrow]$).
- Similarly, AML can be extended to AAML with a quantifier $[\otimes]$ (dual: $\langle\otimes\rangle$).
- AUML can be extended to AAUML with quantifier $[\Updownarrow]$ (dual: $\langle\Updownarrow\rangle$.

AAML and AAUML

- PAL was extended to APAL (with quantifier $[!]$).
- AUL was extended to AAUL (with quantifier $[\updownarrow]$).
- Similarly, AML can be extended to AAML with a quantifier $[\otimes]$ (dual: $\langle\otimes\rangle$).
- AUML can be extended to AAUML with quantifier $[\Updownarrow]$ (dual: $\langle\Updownarrow\rangle$.
- Details (e.g., restriction on domain of quantification) same as with APAL/AAUL.

Synthesis

- AAML/AAUML are very similar to APAL/AAUL.

Synthesis

- AAML/AAUML are very similar to APAL/AAUL.
- But there is one important difference.

Synthesis

- AAML/AAUML are very similar to APAL/AAUL.
- But there is one important difference.
- In AAML/AAUML we can do *(global) synthesis*, while in APAL/AAUL we cannot.

Synthesis (II)

Definition

The *synthesis* problem for AAML is given as follows:

- **Input** A goal formula φ.
- **Output** An action model E, X such that $\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}, X \rangle \varphi$.

Synthesis (II)

Definition

The *synthesis* problem for AAML is given as follows:

- Input: A goal formula φ.
- Output: An action model E, X such that $\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}, X \rangle \varphi$.

Similarly for AAUML:

Synthesis (II)

Definition

The *synthesis* problem for AAML is given as follows:

Input A goal formula φ.

Output An action model E, X such that $\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}, X \rangle \varphi$.

Similarly for AAUML:

Definition

The *synthesis* problem for AAUML is given as follows:

Input A goal formula φ.

Output An arrow update model U, X such that $\models \langle \Updownarrow \rangle \varphi \leftrightarrow \langle \mathsf{U}, X \rangle \varphi$.

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.
- We are asked to find a *single* action model E, X such that

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.
- We are asked to find a *single* action model E, X such that in *every* pointed model M, s,

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.
- We are asked to find a *single* action model E, X such that in *every* pointed model M, s, if there is some action model E', X' such that $M, s \models \langle \mathsf{E}', X' \rangle \varphi$

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.
- We are asked to find a *single* action model E, X such that in *every* pointed model M, s, if there is some action model E', X' such that $M, s \models \langle \mathsf{E}', X' \rangle \varphi$ then $M, s \models \langle \mathsf{E}, X \rangle \varphi$.

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.
- We are asked to find a *single* action model E, X such that in *every* pointed model M, s, if there is some action model E', X' such that $M, s \models \langle \mathsf{E}', X' \rangle \varphi$ then $M, s \models \langle \mathsf{E}, X \rangle \varphi$.
- In other words: for a fixed goal φ, there must be an event E, S that makes φ true whenever possible.

Synthesis (III)

- Let's take a moment to consider how *weird* it is that global synthesis is possible for AAML/AAUML.
- We are asked to find a *single* action model E, X such that in *every* pointed model M, s, if there is some action model E', X' such that $M, s \models \langle \mathsf{E}', X' \rangle \varphi$ then $M, s \models \langle \mathsf{E}, X \rangle \varphi$.
- In other words: for a fixed goal φ, there must be an event E, S that makes φ true whenever possible.
- Yet other words: there is a *uniform* strategy that achieves φ (whenever possible).

Synthesis (IV)

- We will show how to do synthesis for AAUML.

Synthesis (IV)

- We will show how to do synthesis for AAUML.
- We will omit some details, but the broad idea will be clear.

Synthesis (IV)

- We will show how to do synthesis for AAUML.
- We will omit some details, but the broad idea will be clear.
- Synthesis for AAML is similar. So we don't discuss it in detail.

Synthesis: example

- Start with an example.

Synthesis: example

- Start with an example.
- Suppose $\varphi = p \wedge \Diamond_a q \wedge \Box_a r$.

Synthesis: example

- Start with an example.
- Suppose $\varphi = p \wedge \Diamond_a q \wedge \Box_a r$.
- First thing to note: we can't always make φ true.
  - If $M, s \not\models p$, then $M, s \not\models \langle \mathrm{U}, X\rangle p$ for every U, X.
  - If $M, s \not\models \Diamond_a(q \wedge r)$, then $M, s \not\models \langle \mathrm{U}, X\rangle \Diamond_a q$ or $M, s \not\models \langle \mathrm{U}, X\rangle \Box_a r$ for every U, X.

Synthesis: example

- Start with an example.
- Suppose $\varphi = p \wedge \Diamond_a q \wedge \Box_a r$.
- First thing to note: we can't always make φ true.
  - If $M, s \not\models p$, then $M, s \not\models \langle \mathrm{U}, X\rangle p$ for every U, X.
  - If $M, s \not\models \Diamond_a(q \wedge r)$, then $M, s \not\models \langle \mathrm{U}, X\rangle \Diamond_a q$ or $M, s \not\models \langle \mathrm{U}, X\rangle \Box_a r$ for every U, X.
- This is fine. We don't need to make φ true everywhere, just everywhere *possible*.

Synthesis: example

- Start with an example.
- Suppose $\varphi = p \wedge \Diamond_a q \wedge \Box_a r$.
- First thing to note: we can't always make φ true.
  - If $M, s \not\models p$, then $M, s \not\models \langle \mathrm{U}, X\rangle p$ for every U, X.
  - If $M, s \not\models \Diamond_a(q \wedge r)$, then $M, s \not\models \langle \mathrm{U}, X\rangle \Diamond_a q$ or $M, s \not\models \langle \mathrm{U}, X\rangle \Box_a r$ for every U, X.
- This is fine. We don't need to make φ true everywhere, just everywhere *possible*.
- Possible in this case means: $M, s \models p \wedge \Diamond_a(q \wedge r)$.

Synthesis: example (part 2)

- $\varphi = p \land \Diamond_a q \land \Box_a r$.
- In $M * \mathsf{U}$ we need two worlds: (1) original world where p is true and (2) a-successor where q is true.
- Furthermore, in every a-successor r must be true. Note that this does not increase the number of successors that we need.

Synthesis: example (part 2)

- $\varphi = p \wedge \Diamond_a q \wedge \Box_a r$.
- In $M * \mathsf{U}$ we need two worlds: (1) original world where p is true and (2) a-successor where q is true.
- Furthermore, in every a-successor r must be true. Note that this does not increase the number of successors that we need.
- Need two states. Best way to obtain them: take two outcomes $\mathsf{O} = \{o_1, o_2\}$.

Synthesis: example (part 2)

- $\varphi = p \land \Diamond_a q \land \Box_a r$.
- In $M * \mathsf{U}$ we need two worlds: (1) original world where p is true and (2) a-successor where q is true.
- Furthermore, in every a-successor r must be true. Note that this does not increase the number of successors that we need.
- Need two states. Best way to obtain them: take two outcomes $\mathsf{O} = \{o_1, o_2\}$.
- Arrow $(o_1, \top) \xrightarrow{a} (o_2, r)$.

Synthesis: example (part 3)

- Suppose φ is achievable. So $M, s \models p \wedge \Diamond_a(q \wedge r)$. Let s' be the $q \wedge r$ successor.

Synthesis: example (part 3)

- Suppose φ is achievable. So $M, s \models p \wedge \Diamond_a(q \wedge r)$. Let s' be the $q \wedge r$ successor.
- Then
  1. $M * \mathsf{U}, (s, o_1) \models p$,

Synthesis: example (part 3)

- Suppose φ is achievable. So $M, s \models p \wedge \Diamond_a(q \wedge r)$. Let s' be the $q \wedge r$ successor.
- Then
  1. $M * \mathrm{U}, (s, o_1) \models p$,
  2. The only possible arrows in $M * \mathrm{U}$ come from $(o_1, \top) \stackrel{a}{\mapsto} (o_2, r)$, so every a-successor satisfies r.
  3. Hence $M * \mathrm{U}, (w, o_1) \models \Box_a r$.

Synthesis: example (part 3)

- Suppose φ is achievable. So $M, s \models p \land \Diamond_a(q \land r)$. Let s' be the $q \land r$ successor.
- Then
  1. $M * \mathsf{U}, (s, o_1) \models p$,
  2. The only possible arrows in $M * \mathsf{U}$ come from $(o_1, \top) \overset{a}{\mapsto} (o_2, r)$, so every a-successor satisfies r.
  3. Hence $M * \mathsf{U}, (w, o_1) \models \Box_a r$.
  4. The world (w, o_1) satisfies $\top$ and (s', o_2) satisfies r. Therefore: (s', o_2) is an a-successor of (s, o_1).
  5. Because (s', o_2) satisfies q: $M * \mathsf{U}, (s, o_1) \models \Diamond_a r$.

Synthesis: example (part 3)

- Suppose φ is achievable. So $M, s \models p \wedge \Diamond_a(q \wedge r)$. Let s' be the $q \wedge r$ successor.
- Then
  1. $M * \mathsf{U}, (s, o_1) \models p$,
  2. The only possible arrows in $M * \mathsf{U}$ come from $(o_1, \top) \xrightarrow{a} (o_2, r)$, so every a-successor satisfies r.
  3. Hence $M * \mathsf{U}, (w, o_1) \models \Box_a r$.
  4. The world (w, o_1) satisfies $\top$ and (s', o_2) satisfies r. Therefore: (s', o_2) is an a-successor of (s, o_1).
  5. Because (s', o_2) satisfies q: $M * \mathsf{U}, (s, o_1) \models \Diamond_a r$.
- Putting it together: $M * \mathsf{U}, (s, o_1) \models \varphi$.

Synthesis: more generally

- Synthesis is done inductively.
- So for every ψ less complex than φ we assume U_ψ, o_ψ.

Synthesis: more generally

- Synthesis is done inductively.
- So for every ψ less complex than φ we assume U_ψ, o_ψ.
- We will work with φ in a normal form: $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$, where

$$\psi_i = \chi_0 \land \bigwedge_{a \in A} (\bigwedge_{1 \leq j \leq k_a} \Diamond_a \chi_j \land \Box_a \xi).$$

Synthesis: more generally

- Synthesis is done inductively.
- So for every ψ less complex than φ we assume U_ψ, o_ψ.
- We will work with φ in a normal form: $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$, where

$$\psi_i = \chi_0 \wedge \bigwedge_{a \in A} (\bigwedge_{1 \leq j \leq k_a} \Diamond_a \chi_j \wedge \Box_a \xi).$$

- Example: $\varphi = (p \wedge \Diamond_a p \wedge \Box_a \top \wedge \Diamond_b q \wedge \Box_b r) \vee (p \wedge q \wedge \Diamond_a r \wedge \Box_a r \wedge \Box_b \top)$

Synthesis: conjunctive part

- We start with the $\bigwedge_{1 \leq j \leq k} \Diamond_a \chi_j \wedge \Box_a \xi$ part.

Synthesis: conjunctive part

- We start with the $\bigwedge_{1\leq j\leq k} \Diamond_a\chi_j \wedge \Box_a\xi$ part.
- Let U_j, o_j be the arrow update model synthesized for $\chi_j \wedge \xi$.

Synthesis: conjunctive part

- We start with the $\bigwedge_{1\leq j\leq k} \Diamond_a \chi_j \wedge \Box_a \xi$ part.
- Let U_j, o_j be the arrow update model synthesized for $\chi_j \wedge \xi$.
- So whenever $\chi_j \wedge \xi$ can be made true, $\langle U_j, o_j \rangle$ will make it true.

Synthesis: conjunctive part

- We start with the $\bigwedge_{1\leq j\leq k}\Diamond_a\chi_j\wedge\Box_a\xi$ part.
- Let U_j, o_j be the arrow update model synthesized for $\chi_j\wedge\xi$.
- So whenever $\chi_j\wedge\xi$ can be made true, $\langle U_j, o_j\rangle$ will make it true.
- Now, let U be disjoint union of $\mathsf{U}_1,\cdots,\mathsf{U}_k$ plus one extra outcome o.
- Add arrows $(o,\top)\overset{a}{\mapsto}(o_j,\langle\mathsf{U}_j,o_j\rangle\xi)$.

Synthesis: conjunctive part (cont.)

- So: what is effect of this U?

Synthesis: conjunctive part (cont.)

- So: what is effect of this U?
- If (s', o') is a successor of (s, o) then we must have
  - $o' = o_j$ for some j and,
  - $M, s' \models \langle \mathrm{U}_j, o_j \rangle \xi$ and therefore $M, (s', o_j) \models \xi$.

Synthesis: conjunctive part (cont.)

- So: what is effect of this U?
- If (s', o') is a successor of (s, o) then we must have
  - $o' = o_j$ for some j and,
  - $M, s' \models \langle \mathrm{U}_j, o_j \rangle \xi$ and therefore $M, (s', o_j) \models \xi$.
- It follows that $M * \mathrm{U}, (s, o) \models \Box_a \xi$.

Synthesis: conjunctive part (cont.)

- So: what is effect of this U?
- If (s', o') is a successor of (s, o) then we must have
  - $o' = o_j$ for some j and,
  - $M, s' \models \langle \mathsf{U}_j, o_j \rangle \xi$ and therefore $M, (s', o_j) \models \xi$.
- It follows that $M * \mathsf{U}, (s, o) \models \Box_a \xi$.
- Furthermore: if ξ and χ_j can be made true simultaneously in s', then they are true in (s', o_j).

Synthesis: conjunctive part (cont.)

- So: what is effect of this U?
- If (s', o') is a successor of (s, o) then we must have
  - $o' = o_j$ for some j and,
  - $M, s' \models \langle \mathsf{U}_j, o_j \rangle \xi$ and therefore $M, (s', o_j) \models \xi$.
- It follows that $M * \mathsf{U}, (s, o) \models \Box_a \xi$.
- Furthermore: if ξ and χ_j can be made true simultaneously in s', then they are true in (s', o_j).
- Hence: U, o always satisfies $\Box_a \xi$ and satisfies $\Diamond_a \chi_j$ whenever possible.

Synthesis: conjunctive part (cont.)

- In the previous slide: we constructed the arrow update model for $\bigwedge_{1\leq j\leq k} \Diamond_a \chi_j \wedge \Box_a \xi$.

Synthesis: conjunctive part (cont.)

- In the previous slide: we constructed the arrow update model for $\bigwedge_{1\leq j\leq k} \Diamond_a \chi_j \wedge \Box_a \xi$.
- So that was the single agent case. Multi-agent generalization is trivial: arrows for different agents don't interact.

Synthesis: conjunctive part (cont.)

- In the previous slide: we constructed the arrow update model for $\bigwedge_{1 \leq j \leq k} \Diamond_a \chi_j \wedge \Box_a \xi$.
- So that was the single agent case. Multi-agent generalization is trivial: arrows for different agents don't interact.
- Left to do: disjunctive part.

Synthesis: disjunctive part

- We now $\mathsf{U}_{\psi_i}, o_{\psi_i}$ for every ψ_i.

Synthesis: disjunctive part

- We now $\mathsf{U}_{\psi_i}, o_{\psi_i}$ for every ψ_i.
- Left to do: design for $\bigvee_{1 \leq i \leq n} \psi_i$.

Synthesis: disjunctive part

- We now $\mathsf{U}_{\psi_i}, o_{\psi_i}$ for every ψ_i.
- Left to do: design for $\bigvee_{1 \leq i \leq n} \psi_i$.
- Two ways to do this: easy and hard.

Synthesis: disjunctive part

- We now $\mathsf{U}_{\psi_i}, o_{\psi_i}$ for every ψ_i.
- Left to do: design for $\bigvee_{1 \leq i \leq n} \psi_i$.
- Two ways to do this: easy and hard.
- We start with easy.

Synthesis: disjunctive part (cont.)

- Suppose $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$ is achievable by some arrow update model.

Synthesis: disjunctive part (cont.)

- Suppose $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$ is achievable by some arrow update model.
- Then some ψ_i is achievable.

Synthesis: disjunctive part (cont.)

- Suppose $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$ is achievable by some arrow update model.
- Then some ψ_i is achievable.
- Therefore, it would be achieved by $\langle \mathsf{U}_{\psi_i}, o_i \rangle$.

Synthesis: disjunctive part (cont.)

- Suppose $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$ is achievable by some arrow update model.
- Then some ψ_i is achievable.
- Therefore, it would be achieved by $\langle \mathsf{U}_{\psi_i}, o_i \rangle$.
- So if we could do all U_{ψ_i} at the same time, we would achieve φ!

Synthesis: disjunctive part (cont.)

- Suppose $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$ is achievable by some arrow update model.
- Then some ψ_i is achievable.
- Therefore, it would be achieved by $\langle \mathsf{U}_{\psi_i}, o_i \rangle$.
- So if we could do all U_{ψ_i} at the same time, we would achieve φ!
- Doing them all at the same time = using a multi-pointed model.

Synthesis: disjunctive part (cont.)

- Suppose $\varphi = \bigvee_{1 \leq i \leq n} \psi_i$ is achievable by some arrow update model.
- Then some ψ_i is achievable.
- Therefore, it would be achieved by $\langle \mathsf{U}_{\psi_i}, o_i \rangle$.
- So if we could do all U_{ψ_i} at the same time, we would achieve φ!
- Doing them all at the same time = using a multi-pointed model.
- So: let $\mathsf{U} = \bigcup_{1 \leq i \leq n} \mathsf{U}_{\psi_i}$. Then

$$\models \langle \Updownarrow \rangle \varphi \leftrightarrow \langle \mathsf{U}, \{o_{\psi_1}, \cdots, o_{\psi_n}\} \rangle \varphi$$

Synthesis: disjunctive part (cont.)

- But: using a multi-pointed model feels like cheating.
- So we'd like to avoid it if possible.

Synthesis: disjunctive part (cont.)

- But: using a multi-pointed model feels like cheating.
- So we'd like to avoid it if possible.
- And it is possible!

Synthesis: disjunctive part (cont.)

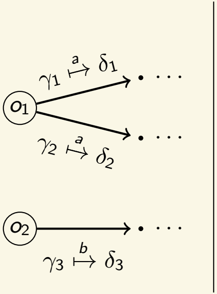
- But: using a multi-pointed model feels like cheating.
- So we'd like to avoid it if possible.
- And it is possible!
- We just have to do a little more work.

Synthesis: disjunctive part (cont.)

- We do the two-disjunct case. Repeat process in case of more disjuncts.

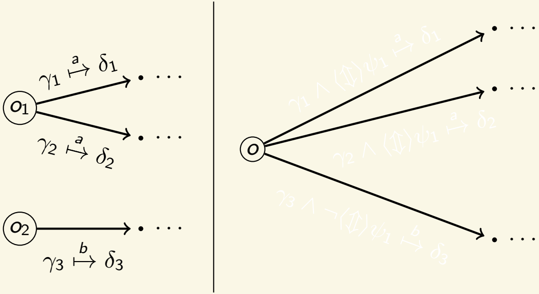
Synthesis: disjunctive part (cont.)

- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



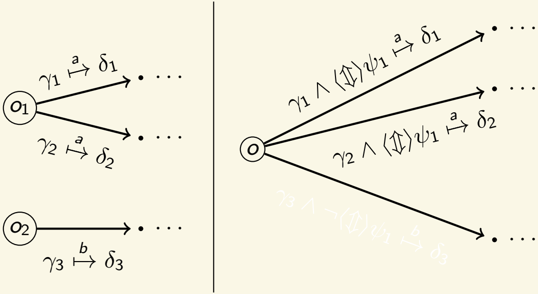
Synthesis: disjunctive part (cont.)

- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



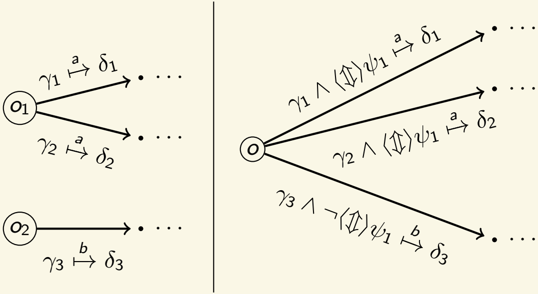
Synthesis: disjunctive part (cont.)

- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



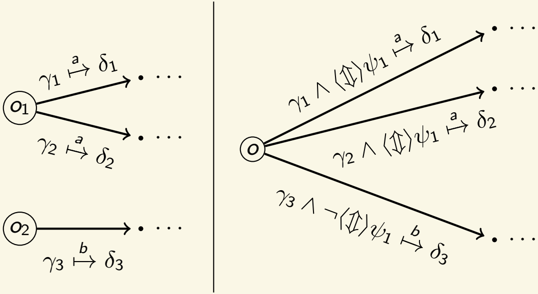
Synthesis: disjunctive part (cont.)

- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



Synthesis: disjunctive part (cont.)

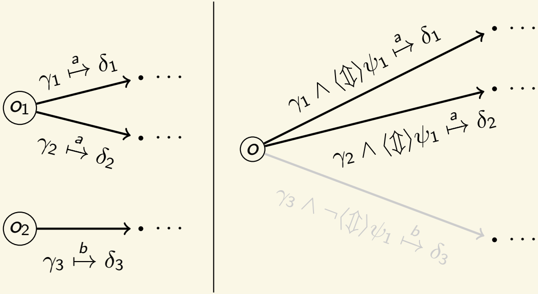
- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



- If ψ_1 is achievable, it will be achieved.

Synthesis: disjunctive part (cont.)

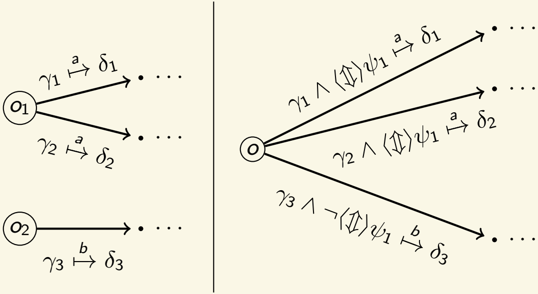
- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



- If ψ_1 is achievable, it will be achieved.

Synthesis: disjunctive part (cont.)

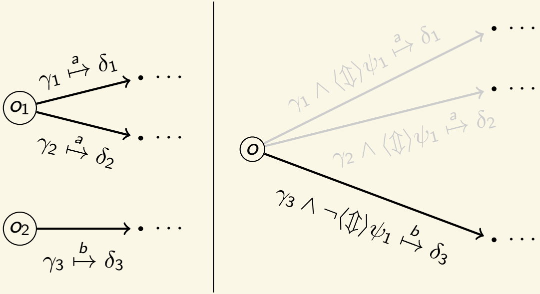
- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



- If ψ_1 is achievable, it will be achieved.
- If ψ_1 is not achievable, it tries to make ψ_2 true.

Synthesis: disjunctive part (cont.)

- We do the two-disjunct case. Repeat process in case of more disjuncts.
- Model with origin o_1 works for ψ_1, with origin o_2 works for ψ_2.



- If ψ_1 is achievable, it will be achieved.
- If ψ_1 is not achievable, it tries to make ψ_2 true.

AAUML synthesis: summary

- The disjunctive step was the last one: we have done all steps for AAUML synthesis.

AAUML synthesis: summary

- The disjunctive step was the last one: we have done all steps for AAUML synthesis.
- That is to say: we can construct $\mathrm{U}_\varphi, o_\varphi$ that achieves φ whenever possible,

AAUML synthesis: summary

- The disjunctive step was the last one: we have done all steps for AAUML synthesis.
- That is to say: we can construct $\mathsf{U}_\varphi, o_\varphi$ that achieves φ whenever possible, i.e.,

$$\models \langle \Updownarrow \rangle \varphi \leftrightarrow \langle \mathsf{U}_\varphi, o_\varphi \rangle \varphi$$

AAML synthesis

- Synthesis of action models can be mostly be done in the same way.

AAML synthesis

- Synthesis of action models can be mostly be done in the same way.
- But: one exception.

AAML synthesis

- Synthesis of action models can be mostly be done in the same way.
- But: one exception.
- The step where we construct a single-pointed arrow update model from two separate arrow update models is impossible with action models.

AAML synthesis

- Synthesis of action models can be mostly be done in the same way.
- But: one exception.
- The step where we construct a single-pointed arrow update model from two separate arrow update models is impossible with action models.
- Hence: AAML synthesis is only possible with multi-pointed action models.

AAML synthesis

- Synthesis of action models can be mostly be done in the same way.
- But: one exception.
- The step where we construct a single-pointed arrow update model from two separate arrow update models is impossible with action models.
- Hence: AAML synthesis is only possible with multi-pointed action models.
- So we find $\mathsf{E}_\varphi, X_\varphi$ such that
$$\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}_\varphi, X_\varphi \rangle \varphi.$$

Table of Contents

Synthesis and reduction

- We have

$$\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}_\varphi, X_\varphi \rangle \varphi$$
$$\models \langle \Updownarrow \rangle \varphi \leftrightarrow \langle \mathsf{U}_\varphi, o_\varphi \rangle \varphi$$

Synthesis and reduction

- We have

$$\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}_\varphi, X_\varphi \rangle \varphi$$
$$\models \langle \Updownarrow \rangle \varphi \leftrightarrow \langle \mathsf{U}_\varphi, o_\varphi \rangle \varphi$$

- Note: these are reduction axioms!

Synthesis and reduction

- We have

$$\models \langle \otimes \rangle \varphi \leftrightarrow \langle \mathsf{E}_\varphi, X_\varphi \rangle \varphi$$
$$\models \langle \Updownarrow \rangle \varphi \leftrightarrow \langle \mathsf{U}_\varphi, o_\varphi \rangle \varphi$$

- Note: these are reduction axioms!
- This means we get all the goodies:
  - Sound and complete axiomatizations for AAML and AAUML!
  - AAML and AAUML are decidable!
  - AAML and AAUML are no more expressive than EL!

Expressivity

- Let me repeat that: AAML and AAUML are no more expressive than EL.

Expressivity

- Let me repeat that: AAML and AAUML are no more expressive than EL.
- Are you shocked? Because you should be!

Expressivity

- Let me repeat that: AAML and AAUML are no more expressive than EL.
- Are you shocked? Because you should be!
- APAL and AAUL are (i) more expressive than PAL and AUL and (ii) undecidable.

Expressivity

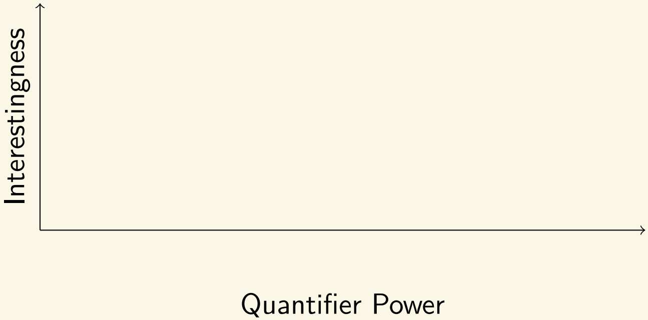
- Let me repeat that: AAML and AAUML are no more expressive than EL.
- Are you shocked? Because you should be!
- APAL and AAUL are (i) more expressive than PAL and AUL and (ii) undecidable.
- What makes AAML and AAUML so different?

Expressivity

- Let me repeat that: AAML and AAUML are no more expressive than EL.
- Are you shocked? Because you should be!
- APAL and AAUL are (i) more expressive than PAL and AUL and (ii) undecidable.
- What makes AAML and AAUML so different?
- Answer: they are a bit too powerful.

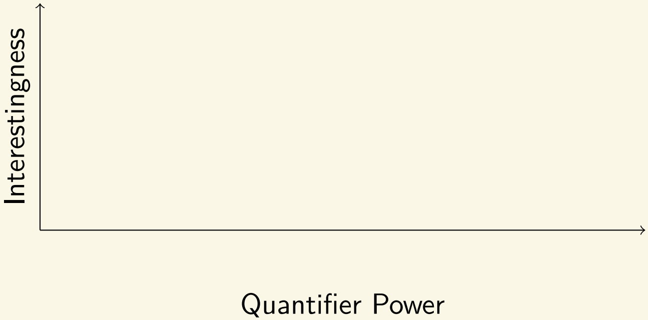
Power curve

- Suppose an agent is completely powerless. The only available action is ”do nothing”.



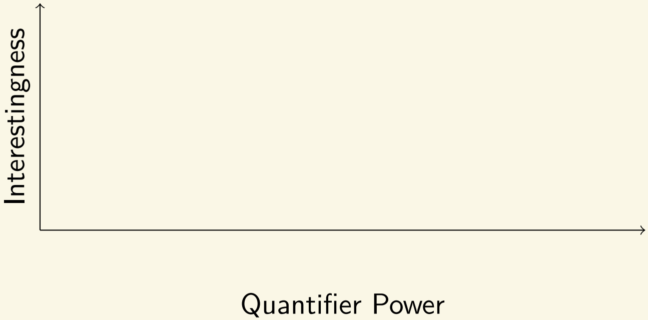
Power curve

- Suppose an agent is completely powerless. The only available action is ”do nothing”.
- Let $[X]$ quantify over this agent's actions.



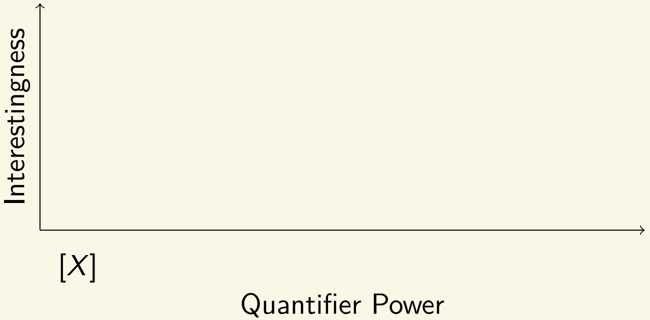
Power curve

- Suppose an agent is completely powerless. The only available action is "do nothing".
- Let $[X]$ quantify over this agent's actions.
- This quantifier is boring: $\models [X]\varphi \leftrightarrow \varphi$.



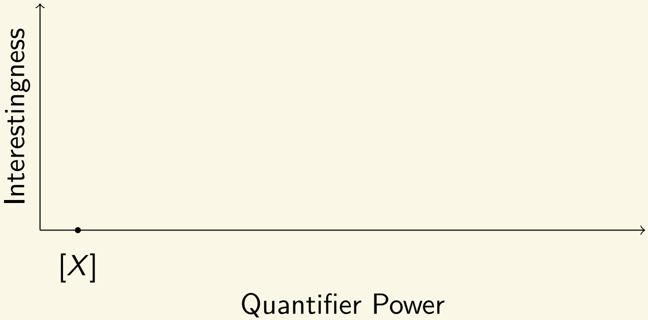
Power curve

- Suppose an agent is completely powerless. The only available action is "do nothing".
- Let $[X]$ quantify over this agent's actions.
- This quantifier is boring: $\models [X]\varphi \leftrightarrow \varphi$.



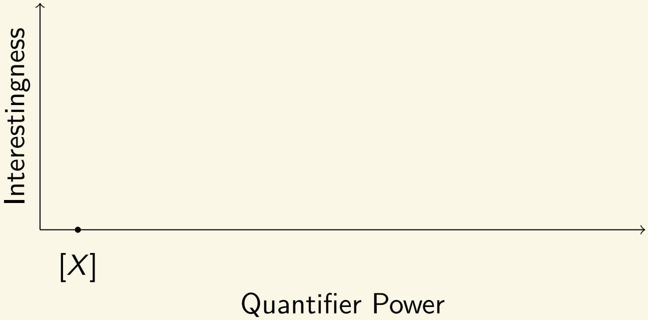
Power curve

- Suppose an agent is completely powerless. The only available action is "do nothing".
- Let $[X]$ quantify over this agent's actions.
- This quantifier is boring: $\models [X]\varphi \leftrightarrow \varphi$.



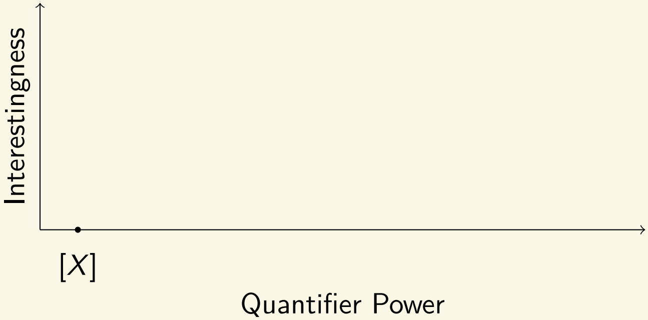
Power curve

- Another agent is omnipotent. In order to make φ true they merely have to snap their fingers.



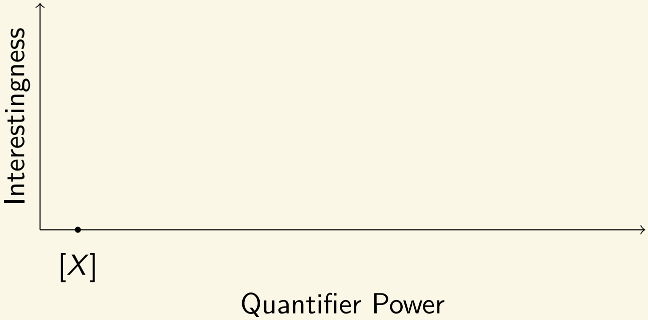
Power curve

- Another agent is omnipotent. In order to make φ true they merely have to snap their fingers.
- Let $[Y]$ quantify over this agent's actions.



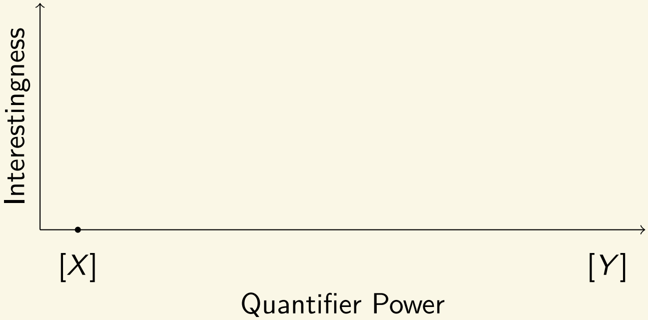
Power curve

- Another agent is omnipotent. In order to make φ true they merely have to snap their fingers.
- Let $[Y]$ quantify over this agent's actions.
- This quantifier is also boring: $\models [Y]\varphi \leftrightarrow \top$.



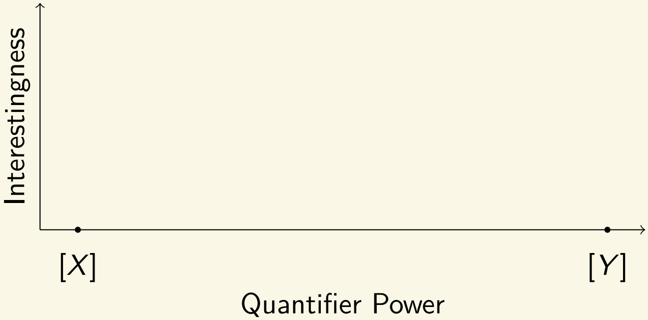
Power curve

- Another agent is omnipotent. In order to make φ true they merely have to snap their fingers.
- Let $[Y]$ quantify over this agent's actions.
- This quantifier is also boring: $\models [Y]\varphi \leftrightarrow \top$.



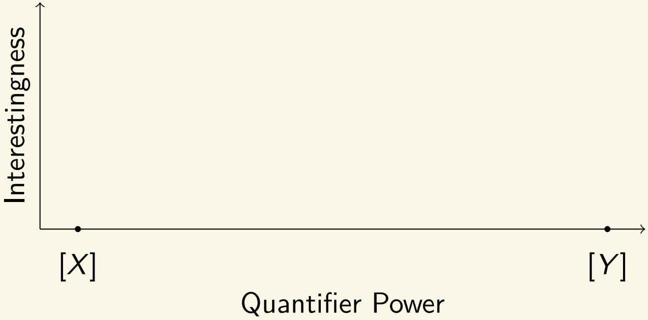
Power curve

- Another agent is omnipotent. In order to make φ true they merely have to snap their fingers.
- Let $[Y]$ quantify over this agent's actions.
- This quantifier is also boring: $\models [Y]\varphi \leftrightarrow \top$.



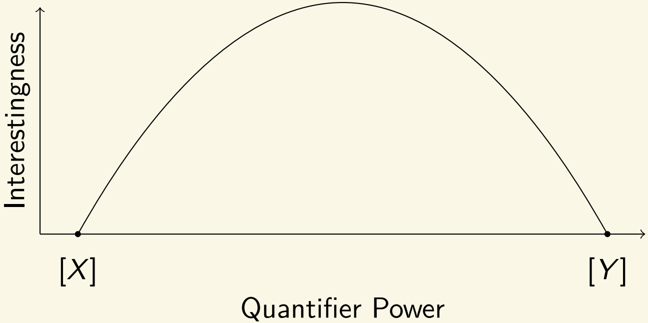
Power curve

- In between those power extremes things get more interesting.



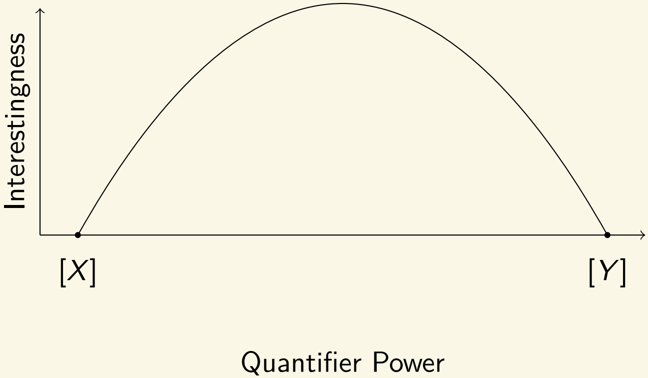
Power curve

- In between those power extremes things get more interesting.



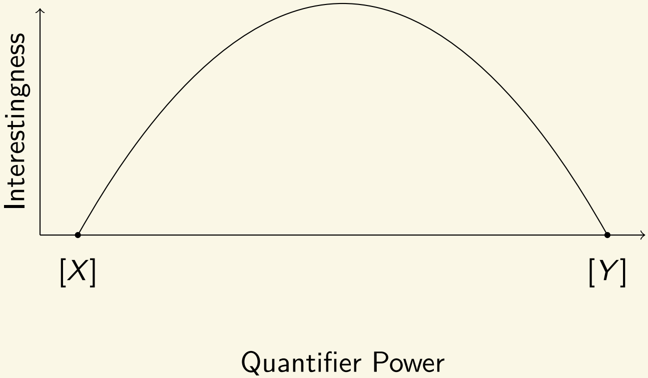
Power curve

- Where to place the various other quantifiers?



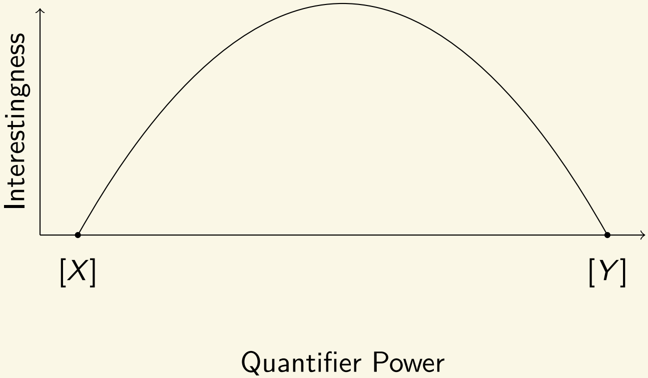
Power curve

- Where to place the various other quantifiers?
- Unknown whether they are left or right of the peak.



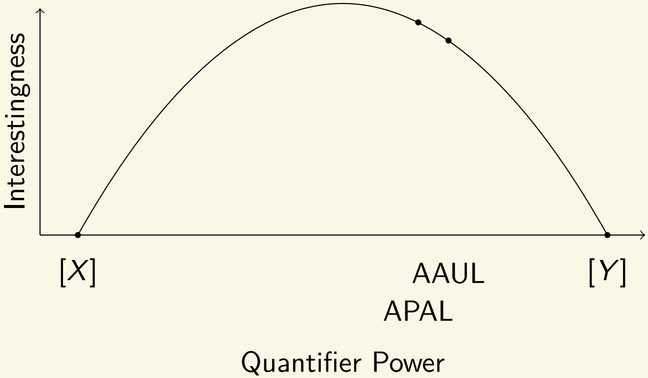
Power curve

- Where to place the various other quantifiers?
- Unknown whether they are left or right of the peak.
- But either way: pretty close to the top.



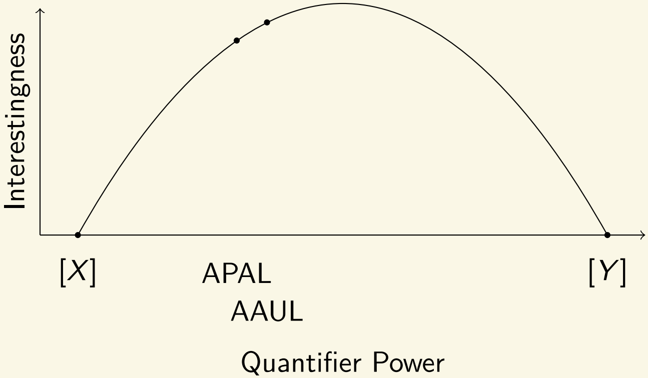
Power curve

- Where to place the various other quantifiers?
- Unknown whether they are left or right of the peak.
- But either way: pretty close to the top.



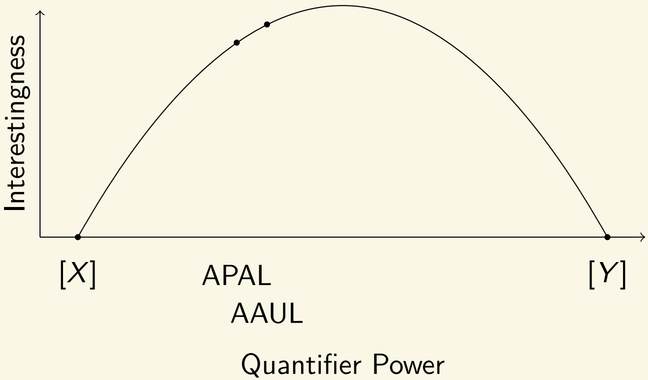
Power curve

- Where to place the various other quantifiers?
- Unknown whether they are left or right of the peak.
- But either way: pretty close to the top.



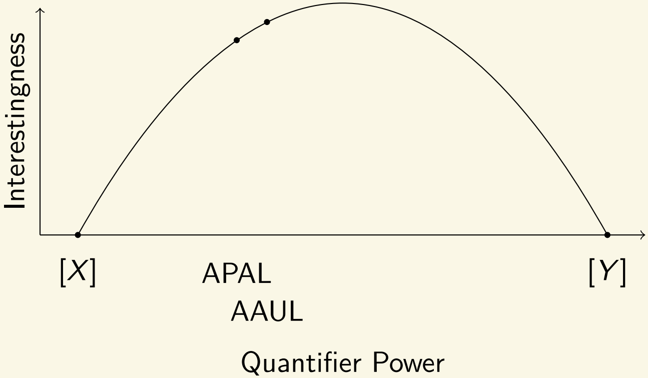
Power curve

- What about AAML/AAUML?



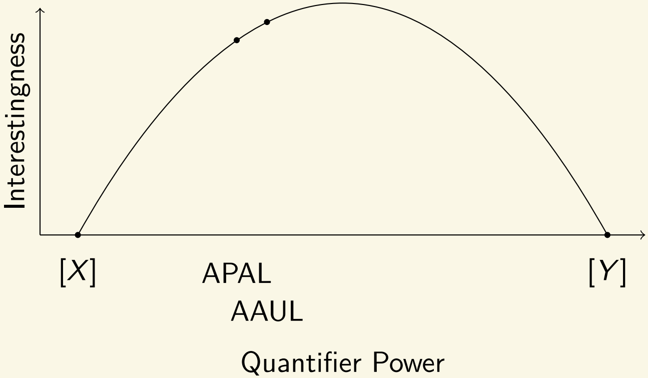
Power curve

- What about AAML/AAUML?
- Existence of reduction suggest they are not as interesting as APAL/AUML.



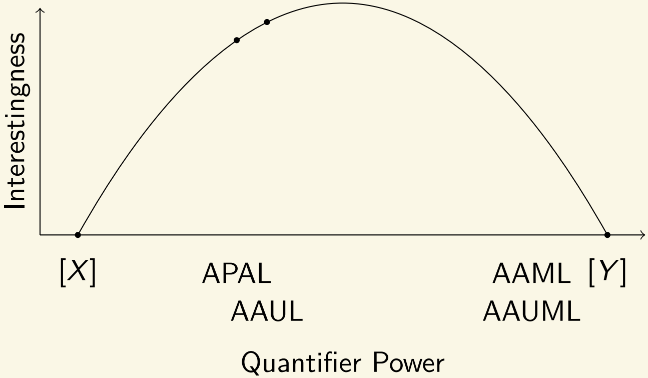
Power curve

- What about AAML/AAUML?
- Existence of reduction suggest they are not as interesting as APAL/AUML.
- We should therefore place them somewhat over here.



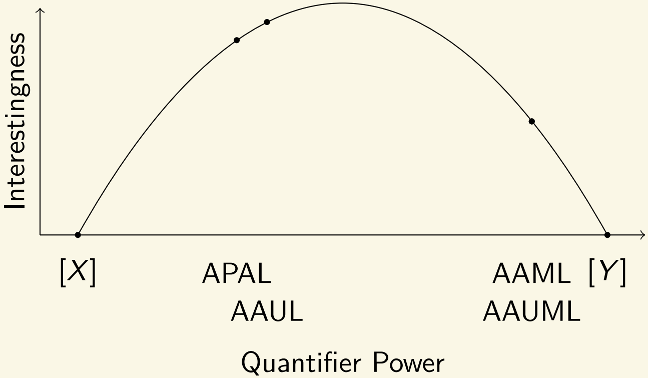
Power curve

- What about AAML/AAUML?
- Existence of reduction suggest they are not as interesting as APAL/AUML.
- We should therefore place them somewhat over here.



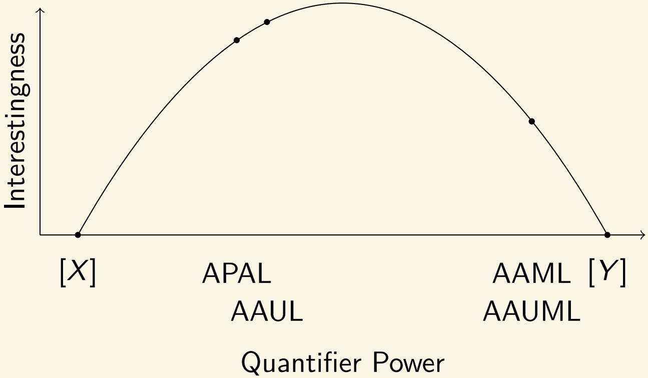
Power curve

- What about AAML/AAUML?
- Existence of reduction suggest they are not as interesting as APAL/AUML.
- We should therefore place them somewhat over here.



Power curve

- What about AAML/AAUML?
- Existence of reduction suggest they are not as interesting as APAL/AUML.
- We should therefore place them somewhat over here.
- They are not boring, but clearly over the top of interestingness.



Consequences of the curve

- What conclusions should we draw from this curve?

Consequences of the curve

- What conclusions should we draw from this curve?
- AAML/AAUML are over the "peak" of interestingness.

Consequences of the curve

- What conclusions should we draw from this curve?
- AAML/AAUML are over the "peak" of interestingness.
- The fact that synthesis and reduction are possible, ,

Consequences of the curve

- What conclusions should we draw from this curve?
- AAML/AAUML are over the "peak" of interestingness.
- The fact that synthesis and reduction are possible, while itself interesting.,

Consequences of the curve

- What conclusions should we draw from this curve?
- AAML/AAUML are over the "peak" of interestingness.
- The fact that synthesis and reduction are possible, while itself interesting., makes the operators less interesting (but not completely boring).

Consequences of the curve

- What conclusions should we draw from this curve?
- AAML/AAUML are over the "peak" of interestingness.
- The fact that synthesis and reduction are possible, while itself interesting., makes the operators less interesting (but not completely boring).
- Main effect: if we want to define other quantified update operators, we should use updates that are *less* powerful than action models/arrow update models.

Consequences of the curve

- What conclusions should we draw from this curve?
- AAML/AAUML are over the "peak" of interestingness.
- The fact that synthesis and reduction are possible, while itself interesting., makes the operators less interesting (but not completely boring).
- Main effect: if we want to define other quantified update operators, we should use updates that are *less* powerful than action models/arrow update models.
- Note that group announcement and coalition announcements fall in this category.

Table of Contents

Summary

Today's overall message:

Summary

Today's overall message:

- Action models and arrow update models are generalizations of public announcements and arrow updates, respectively.

Summary

Today's overall message:

- Action models and arrow update models are generalizations of public announcements and arrow updates, respectively.
- $M * [\mathsf{E}, o]$ and $M * [\mathsf{U}, o]$ can be more complex than M.

Summary

Today's overall message:

- Action models and arrow update models are generalizations of public announcements and arrow updates, respectively.
- $M * [\mathsf{E}, o]$ and $M * [\mathsf{U}, o]$ can be more complex than M.
- Quantified versions: AAML and AAUML can be defined.

Summary

Today's overall message:

- Action models and arrow update models are generalizations of public announcements and arrow updates, respectively.
- $M * [\mathsf{E}, o]$ and $M * [\mathsf{U}, o]$ can be more complex than M.
- Quantified versions: AAML and AAUML can be defined.
- Surprisingly: global synthesis is possible for AAML and AAUML.

Summary

Today's overall message:

- Action models and arrow update models are generalizations of public announcements and arrow updates, respectively.
- $M * [\mathsf{E}, o]$ and $M * [\mathsf{U}, o]$ can be more complex than M.
- Quantified versions: AAML and AAUML can be defined.
- Surprisingly: global synthesis is possible for AAML and AAUML.
- As a result: both logics have the same expressivity as EL.

Summary

Today's overall message:

- Action models and arrow update models are generalizations of public announcements and arrow updates, respectively.
- $M * [\mathsf{E}, o]$ and $M * [\mathsf{U}, o]$ can be more complex than M.
- Quantified versions: AAML and AAUML can be defined.
- Surprisingly: global synthesis is possible for AAML and AAUML.
- As a result: both logics have the same expressivity as EL.
- This suggests: we should look for interesting updates that are less powerful than action models/arrow update models, not more powerful.