Quantification in Dynamic Epistemic Logic
Day 1

Rustam Galimullin & Louwe B. Kuijer

ESSLLI 2023

Hi.

This course

- Welcome to the “Quantification in Dynamic Epistemic Logic” course.
- I am Louwe Kuijer.
- I will be teaching this course together with Rustam Galimullin.

Table of Contents

Course overview

- 5 days, 1 lecture each.

  Day 1: Non-quantified DEL.

  Day 2: APAL and friends.

  Day 3: GAL and CAL.

  Day 4: Group knowledge.

  Day 5: AAML and AAUML.

- See course website for more details.
  (Linked from Discord and ESSLLI course catalogue.)

Further reading

- Most of this course is based directly on research papers (as opposed to textbooks and handbooks).
- As a result: not a lot of easy reading on this topic.
- Website does provide list of papers for further reading.
- But: expect those to be highly detailed and technical.

Exercises

- We have written some exercises that you can do to test yourself.
- They are, of course, completely optional.
- Solutions will not be published or discussed during the lectures.
- If you want to discuss the exercises: talk to us before or after the lecture.

Table of Contents

Epistemic logic

- Our starting point: epistemic logic (EL).
- Used to represent the information state of one or more agents *at a specific point in time*.

Epistemic logic: language

Definition

The *language of epistemic logic* (EL) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi$$

where $a \in A$ and $p \in P$.

Epistemic logic: language

Definition

The *language of epistemic logic* (EL) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi$$

where $a \in A$ and $p \in P$.

- As usual: $\wedge, \rightarrow, \leftrightarrow$ as abbreviations. Also: $\Diamond$ as dual of $\Box$.

Epistemic logic: language

Definition

The *language of epistemic logic* (EL) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi$$

where $a \in A$ and $p \in P$.

- As usual: $\wedge, \rightarrow, \leftrightarrow$ as abbreviations. Also: $\Diamond$ as dual of $\Box$.
- $\Box_a \varphi$ read as "agent a knows that φ (is true)".
- $\Diamond_a \varphi$ read as "agent a considers it possible that φ (is true)".

Epistemic logic: models

Definition

A *model* of epistemic logic is a triple $M = (S, \{R_a\}_{a \in A}, V)$ where

- S is a set of states (also called worlds),
- for each $a \in A$, $R_a \subseteq S \times S$ is an accessibility relations and
- $V : P \to 2^S$ is a valuation function.

Epistemic logic: models

Definition

A *model* of epistemic logic is a triple $M = (S, \{R_a\}_{a \in A}, V)$ where

- S is a set of states (also called worlds),
- for each $a \in A$, $R_a \subseteq S \times S$ is an accessibility relations and
- $V : P \to 2^S$ is a valuation function.

Note: in general, no reflexivity/transitivity/symmetry assumptions on R_a.

Epistemic logic: models

Definition

A *model* of epistemic logic is a triple $M = (S, \{R_a\}_{a \in A}, V)$ where

- S is a set of states (also called worlds),
- for each $a \in A$, $R_a \subseteq S \times S$ is an accessibility relations and
- $V : P \to 2^S$ is a valuation function.

Note: in general, no reflexivity/transitivity/symmetry assumptions on R_a.
When we do assume that the relation is an equivalence, write $\sim_a$ for R_a.

Epistemic logic: semantics

Semantics are as usual.

Epistemic logic: semantics

Semantics are as usual.

Definition

The *satisfaction relation* $\models$ is given by

$$
\begin{array}{lcl}
M, s \models p & \Leftrightarrow & s \in V(p), \\
M, s \models \neg\varphi & \Leftrightarrow & M, s \not\models \varphi, \\
M, s \models \varphi \vee \psi & \Leftrightarrow & M, s \models \varphi \text{ or } M, s \models \psi, \\
M, s \models \Box_a\varphi & \Leftrightarrow & \forall s' \in S\text{: if } (s, s') \in R_a \text{ then } M, s' \models \varphi.
\end{array}
$$

Example: cards (simple)

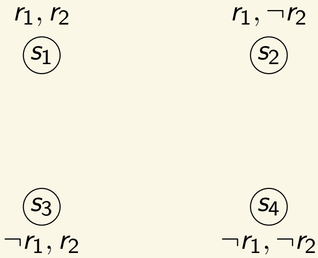
Situation:

- Agents: Rustam (r) and Louwe (l).
- Two cards from standard deck of playing cards, placed face down on table.
- We only care about whether the cards are red or black.

Example: cards (simple)

Situation:

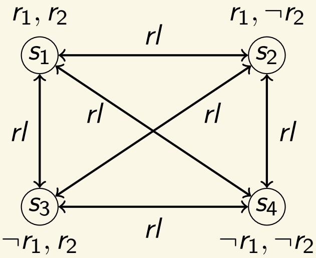
- Agents: Rustam (r) and Louwe (l).
- Two cards from standard deck of playing cards, placed face down on table.
- We only care about whether the cards are red or black.



Example: cards (simple)

Situation:

- Agents: Rustam (r) and Louwe (l).
- Two cards from standard deck of playing cards, placed face down on table.
- We only care about whether the cards are red or black.



Information change

- Interesting thing about knowledge and information: they tend to change over time.

Information change

- Interesting thing about knowledge and information: they tend to change over time.
- Model represents specific information state.

Information change

- Interesting thing about knowledge and information: they tend to change over time.
- Model represents specific information state.
- Information change therefore requires model change.

Example: cards (still simple)

Situation:

- Card distribution as before.

Example: cards (still simple)

Situation:

- Card distribution as before.
- Now: I look at the first card, without showing it to Rustam.

Example: cards (still simple)

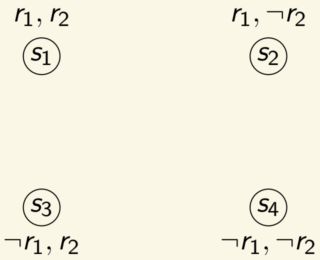
Situation:

- Card distribution as before.
- Now: I look at the first card, without showing it to Rustam.
- Set of worlds.

Example: cards (still simple)

Situation:

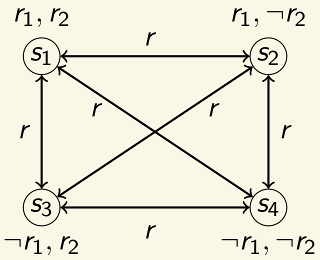
- Card distribution as before.
- Now: I look at the first card, without showing it to Rustam.
- Set of worlds.
- Arrows for Rustam.



Example: cards (still simple)

Situation:

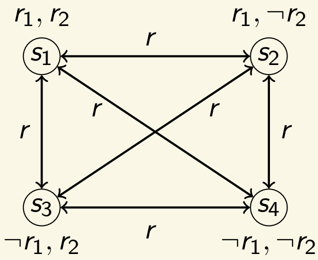
- Card distribution as before.
- Now: I look at the first card, without showing it to Rustam.
- Set of worlds.
- Arrows for Rustam.



Example: cards (still simple)

Situation:

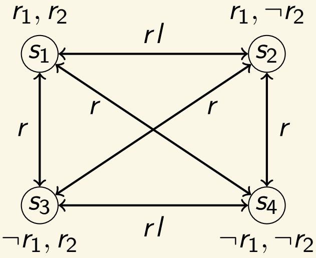
- Card distribution as before.
- Now: I look at the first card, without showing it to Rustam.
- Set of worlds.
- Arrows for Rustam.
- Arrows for Louwe.



Example: cards (still simple)

Situation:

- Card distribution as before.
- Now: I look at the first card, without showing it to Rustam.
- Set of worlds.
- Arrows for Rustam.
- Arrows for Louwe.



Getting to the point

- Oy, Louwe! Those examples are insultingly simple, why did you show them to us?

Getting to the point

- Oy, Louwe! Those examples are insultingly simple, why did you show them to us?
- Answer: while they are simple, there is a point to them.

Getting to the point

- Oy, Louwe! Those examples are insultingly simple, why did you show them to us?
- Answer: while they are simple, there is a point to them.
- Note that we *can* reason about information change using EL as opposed to DEL. (We just did.)
- But: it's relatively hard.

Reasoning about information change: EL vs. DEL

EL	
Ad-hoc	
Analyze twice	
Lots of effort	
Meta-logical	

Table of Contents

Model transformers

- Suppose we want to do information change in a systematic way.
- How would we do this?

Model transformers

- Suppose we want to do information change in a systematic way.
- How would we do this?
- Take information changing event e.
- Effect of e is to change information state,

Model transformers

- Suppose we want to do information change in a systematic way.
- How would we do this?
- Take information changing event e.
- Effect of e is to change information state,
- Information state = pointed Kripke model.
- Initial model M_s turns into model $M * e_s$.

Model transformers

- Suppose we want to do information change in a systematic way.
- How would we do this?
- Take information changing event e.
- Effect of e is to change information state,
- Information state = pointed Kripke model.
- Initial model M_s turns into model $M * e_s$.
- In other words: e is a function that transforms models.

Updates as functions

- Let $\mathfrak{M}$ be the class of pointed models.
- Event e is a function $e : \mathfrak{M} \to \mathfrak{M}$.

Updates as functions

- Let $\mathfrak{M}$ be the class of pointed models.
- Event e is a function $e : \mathfrak{M} \to \mathfrak{M}$.
- (Actually: partial function.)

What does an update change?

- Left to do: define the behaviour of function e.

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.
- First choice: what part of the model should e change?

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.
- First choice: what part of the model should e change?
- $M = (S, R, V)$.

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.
- First choice: what part of the model should e change?
- $M = (S, R, V)$.
- Option 1: event changes S: public announcement.

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.
- First choice: what part of the model should e change?
- $M = (S, R, V)$.
- Option 1: event changes S: public announcement.
- Option 2: event changes R_a: arrow updates.

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.
- First choice: what part of the model should e change?
- $M = (S, R, V)$.
- Option 1: event changes S: public announcement.
- Option 2: event changes R_a: arrow updates.
- Option 3: event changes V: substitutions.

What does an update change?

- Left to do: define the behaviour of function e.
- Admittedly not a minor task.
- First choice: what part of the model should e change?
- $M = (S, R, V)$.
- Option 1: event changes S: public announcement.
- Option 2: event changes R_a: arrow updates.
- Option 3: event changes V: substitutions.
- Option 4: all of the above: action models, arrow update models. (Discussed later this week.)

Simplifying vs. "complexifying"

- Public announcements, arrow updates and substitution reduce, or at least do not increase, the complexity of a model.
- Action models and arrow update models do increase complexity.
- We start by considering the three simplifying update types.

Table of Contents

Public announcements

- Public announcements change a model by *restricting the set of worlds*.

Public announcements

- Public announcements change a model by *restricting the set of worlds*.
- Not just any restriction, though: must be definable.

Public announcements

- Public announcements change a model by *restricting the set of worlds*.
- Not just any restriction, though: must be definable.
- Specifically: announcement ψ restricts S to $S \cap [\![\psi]\!]_M$.

Public announcements: formally

Definition

Let $M = (S, R, V)$ be a model and ψ a formula. Then $M * \psi = (S * \psi, R * \psi, V * \psi)$ where

Public announcements: formally

Definition

Let $M = (S, R, V)$ be a model and ψ a formula. Then $M * \psi = (S * \psi, R * \psi, V * \psi)$ where

- $S * \psi = \{s \in S \mid M, s \models \psi\}$,

Public announcements: formally

Definition

Let $M = (S, R, V)$ be a model and ψ a formula. Then $M * \psi = (S * \psi, R * \psi, V * \psi)$ where

- $S * \psi = \{s \in S \mid M, s \models \psi\}$,
- $(R * \psi)_a = R_a \cap (S * \psi \times S * \psi)$,

Public announcements: formally

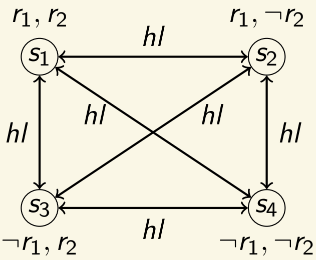
Definition

Let $M = (S, R, V)$ be a model and ψ a formula. Then $M * \psi = (S * \psi, R * \psi, V * \psi)$ where

- $S * \psi = \{s \in S \mid M, s \models \psi\}$,
- $(R * \psi)_a = R_a \cap (S * \psi \times S * \psi)$,
- $V * \psi(p) = V(p) \cap S * \psi$.

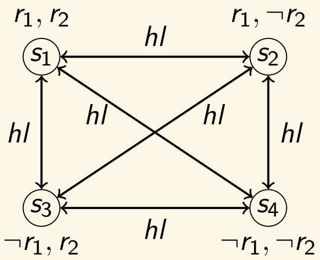
Public announcements: simple example

- Same cards example as before.



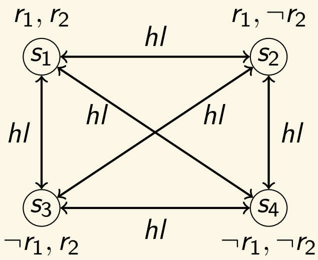
Public announcements: simple example

- Same cards example as before.
- Now, instead of privately looking at a card, I publicly show that the first card is red.



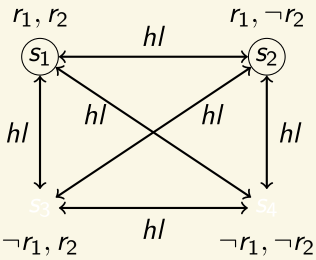
Public announcements: simple example

- Same cards example as before.
- Now, instead of privately looking at a card, I publicly show that the first card is red.
- Announcement: r_1.



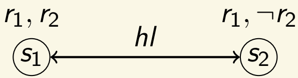
Public announcements: simple example

- Same cards example as before.
- Now, instead of privately looking at a card, I publicly show that the first card is red.
- Announcement: r_1.



Public announcements: simple example

- Same cards example as before.
- Now, instead of privately looking at a card, I publicly show that the first card is red.
- Announcement: r_1.



Not yet PAL

- Note: so far we have *not* defined Public Announcement Logic.
- No public announcement operator in the language yet!

Not yet PAL

- Note: so far we have *not* defined Public Announcement Logic.
- No public announcement operator in the language yet!
- But: we have already defined the function $[\psi] : \mathfrak{M} \to \mathfrak{M}$.
- So information change is already systematic.

Not yet PAL

- Note: so far we have *not* defined Public Announcement Logic.
- No public announcement operator in the language yet!
- But: we have already defined the function $[\psi] : \mathfrak{M} \to \mathfrak{M}$.
- So information change is already systematic.

EL	Not yet PAL
Ad-hoc	Systematic
Analyze twice	Analyze two things
Lots of effort	Easy(ish)
Meta-logical	Meta-logical

Not yet PAL

- Note: so far we have *not* defined Public Announcement Logic.
- No public announcement operator in the language yet!
- But: we have already defined the function $[\psi] : \mathfrak{M} \to \mathfrak{M}$.
- So information change is already systematic.

EL	Not yet PAL
Ad-hoc	Systematic
Analyze twice	Analyze two things
Lots of effort	Easy(ish)
Meta-logical	Meta-logical

- But eventually we do of course want to add announcements to the language.

Public Announcement Logic

Definition

The *language of public announcement logic* (PAL) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a\varphi \mid [\varphi]\varphi$$

where $a \in A$ and $p \in P$.

Public Announcement Logic

Definition

The *language of public announcement logic* (PAL) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\varphi]\varphi$$

where $a \in A$ and $p \in P$.

- $\langle\varphi\rangle$ as dual of $[\varphi]$.

PAL: semantics

Definition

The *satisfaction relation* $\models$ is extended with

$$M, s \models [\varphi]\psi \quad \Leftrightarrow \quad [\varphi](M, s) \models \psi$$

PAL: semantics

Definition

The *satisfaction relation* $\models$ is extended with

$$M, s \models [\varphi]\psi \quad \Leftrightarrow \quad \text{if } [\varphi](M, s) \text{ exists then } [\varphi](M, s) \models \psi$$

PAL: semantics

Definition

The *satisfaction relation* $\models$ is extended with

$$M, s \models [\varphi]\psi \quad \Leftrightarrow \quad \text{if } [\varphi](M, s) \text{ exists then } [\varphi](M, s) \models \psi$$

$$\text{Equivalent to:} \quad M, s \models [\varphi]\psi \quad \Leftrightarrow \quad \text{if } M, s \models \varphi \text{ then } M * [\varphi], s \models \psi.$$

Ways to do information change

EL	Not yet PAL	PAL
Ad-hoc	Systematic	Systematic
Analyze twice	Analyze two things	Analyze two things
Lots of effort	Easy(ish)	Easy(ish)
Meta-logical	Meta-logical	In object language

Table of Contents

Next up: substitutions

- We have discussed public announcements.
- Arrow updates are more complicated, so we leave them for later.
- First, we discuss substitutions (a.k.a. assignments).

Factual change

- Public announcements change S.[1]

Factual change

- Public announcements change S.[1]

[1] And, in a trivial way, R and V.

Factual change

- Public announcements change S.[1]
- Arrow updates change R.

[1] And, in a trivial way, R and V.

Factual change

- Public announcements change S.[1]
- Arrow updates change R.
- Substitutions change V.

[1] And, in a trivial way, R and V.

Factual change

- Public announcements change S.[1]
- Arrow updates change R.
- Substitutions change V.
- This means that substitutions represent *factual change* instead of *information change*.

[1] And, in a trivial way, R and V.

Factual change

- Public announcements change S.[1]
- Arrow updates change R.
- Substitutions change V.
- This means that substitutions represent *factual change* instead of *information change*.
- This course is about information change, so we won't say much about substitutions.
- But we do briefly discuss them for the sake of completeness.

[1] And, in a trivial way, R and V.

Substitutions

- Substitutions take form $[p_1 := \varphi_1, \cdots, p_n := \varphi_n]$.

Substitutions

- Substitutions take form $[p_1 := \varphi_1, \cdots, p_n := \varphi_n]$.
- Effect: atom $V(p_i)$ changes to $[\![\varphi_i]\!]_M$.

Substitutions

- Substitutions take form $[p_1 := \varphi_1, \cdots, p_n := \varphi_n]$.
- Effect: atom $V(p_i)$ changes to $[\![\varphi_i]\!]_M$.
- Formally: let $\sigma = [p_1 := \varphi_1, \cdots, p_n := \varphi_n]$. Then $M * \sigma = (S, R, V * \sigma)$ where

$$V * \sigma(p) = \begin{cases} [\![\varphi_i]\!]_M & \text{if } p = p_i \\ V(p) & \text{otherwise} \end{cases}$$

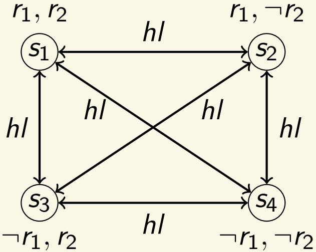
Substitutions

- Substitutions take form $[p_1 := \varphi_1, \cdots, p_n := \varphi_n]$.
- Effect: atom $V(p_i)$ changes to $[\![\varphi_i]\!]_M$.
- Formally: let $\sigma = [p_1 := \varphi_1, \cdots, p_n := \varphi_n]$. Then $M * \sigma = (S, R, V * \sigma)$ where

$$V * \sigma(p) = \begin{cases} [\![\varphi_i]\!]_M & \text{if } p = p_i \\ V(p) & \text{otherwise} \end{cases}$$

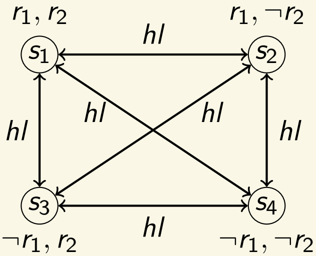
- Effect is global, i.e., common knowledge.

Substitutions: example 1



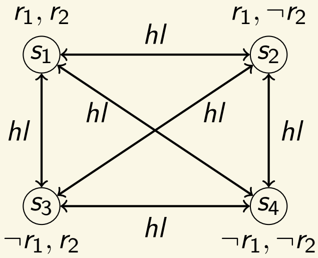
Substitutions: example 1

- Suppose I replace the first card by a black one.



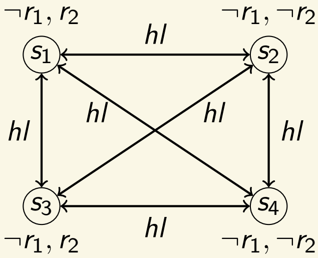
Substitutions: example 1

- Suppose I replace the first card by a black one.
- Represented by $[r_1 := \bot]$

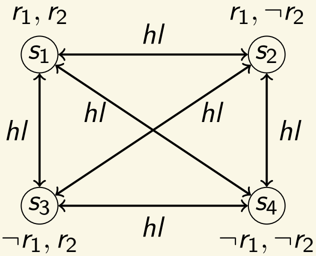


Substitutions: example 1

- Suppose I replace the first card by a black one.
- Represented by $[r_1 := \bot]$

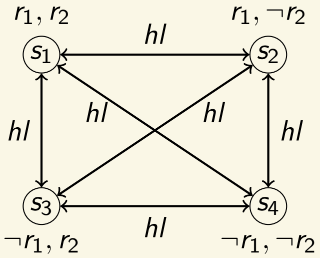


Substitutions: example 2



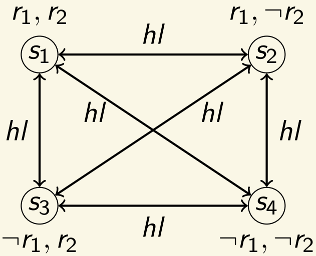
Substitutions: example 2

- Suppose I switch around the two cards.



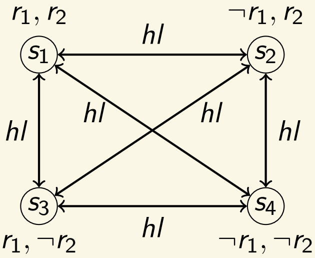
Substitutions: example 2

- Suppose I switch around the two cards.
- Represented by $[r_1 := r_2, r_2 := r_1]$



Substitutions: example 2

- Suppose I switch around the two cards.
- Represented by $[r_1 := r_2, r_2 := r_1]$



Substitutions in a logical language

Definition

The *language of epistemic logic with factual change* (EL+$[\sigma]$) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\sigma]\varphi$$

$$\sigma ::= \epsilon \mid \sigma, p := \varphi$$

where $a \in A$, $p \in P$ and ϵ is the empty sequence.

Substitutions in a logical language

Definition

The *language of epistemic logic with factual change* (EL+$[\sigma]$) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\sigma]\varphi$$

$$\sigma ::= \epsilon \mid \sigma, p := \varphi$$

where $a \in A$, $p \in P$ and ϵ is the empty sequence.

The *satisfaction relation* $\models$ is extended with

$$M, s \models [\sigma]\varphi \quad \Leftrightarrow \quad [\sigma](M, s) \models \varphi$$

Substitutions in a logical language

Definition

The *language of epistemic logic with factual change* (EL+$[\sigma]$) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [\sigma]\varphi$$

$$\sigma ::= \epsilon \mid \sigma, p := \varphi$$

where $a \in A$, $p \in P$ and ϵ is the empty sequence.

The *satisfaction relation* $\models$ is extended with

$$M, s \models [\sigma]\varphi \quad \Leftrightarrow \quad [\sigma](M, s) \models \varphi$$

where $[\sigma](M, s) = M * [\sigma], s$.

Systematic

- Note: as with PAL, we do not need substitutions in the language to do factual change systematically.
- But having them in the language still helps, by allowing in-logic reasoning.

Table of Contents

And now, arrow updates

- Finally, we arrive at arrow updates.

And now, arrow updates

- Finally, we arrive at arrow updates.
- I personally really like them.

And now, arrow updates

- Finally, we arrive at arrow updates.
- I personally really like them.
- But I must admit: they are rather complicated.

And now, arrow updates

- Finally, we arrive at arrow updates.
- I personally really like them.
- But I must admit: they are rather complicated.
- So before looking at the details: brief high level overview.

Introducing: arrow updates

- An information changing event is a public announcement if the following conditions are satisfied:

Introducing: arrow updates

- An information changing event is a public announcement if the following conditions are satisfied:
  1. Information is gained, not lost.
  2. All agents gain the same information.
  3. These three conditions are common knowledge.

Introducing: arrow updates

- An information changing event is a public announcement if the following conditions are satisfied:
    1. Information is gained, not lost.
    2. All agents gain the same information.
    3. These three conditions are common knowledge.
- Arrow updates relax the 2nd condition: agents may gain different information.
- As a result: not common knowledge what information is gained.

Introducing: arrow updates

- An information changing event is a public announcement if the following conditions are satisfied:
  1. Information is gained, not lost.
  2. All agents gain the same information.
  3. These three conditions are common knowledge.
- Arrow updates relax the 2nd condition: agents may gain different information.
- As a result: not common knowledge what information is gained.
- But: still required to be common knowledge what information is gained under what circumstances.

Arrow updates: example

- We already saw an example of an arrow update earlier.
- Recall: example of me looking at the first card.

Arrow updates: example

- We already saw an example of an arrow update earlier.
- Recall: example of me looking at the first card.
- Not all agents gain the same information (Rustam does not see the card, I do).

Arrow updates: example

- We already saw an example of an arrow update earlier.
- Recall: example of me looking at the first card.
- Not all agents gain the same information (Rustam does not see the card, I do).
- But Rustam does know the conditions for my information gain: if the card is read I will learn r_1, if it is black I will learn $\neg r_1$.

Arrow updates: example

- We already saw an example of an arrow update earlier.
- Recall: example of me looking at the first card.
- Not all agents gain the same information (Rustam does not see the card, I do).
- But Rustam does know the conditions for my information gain: if the card is read I will learn r_1, if it is black I will learn $\neg r_1$.
- Hence this is an arrow update.

Arrow updates: syntax

- An arrow update must specify what an agent will learn under what conditions.
- So three parts: condition, agent and information learned.

Arrow updates: syntax

- An arrow update must specify what an agent will learn under what conditions.
- So three parts: condition, agent and information learned.
- Left to decide: specify information learned as (i) what remains possible or (ii) what becomes impossible.
- With public announcements, we specify what remains possible ($[\varphi]$ means φ worlds remain).

Arrow updates: syntax

- An arrow update must specify what an agent will learn under what conditions.
- So three parts: condition, agent and information learned.
- Left to decide: specify information learned as (i) what remains possible or (ii) what becomes impossible.
- With public announcements, we specify what remains possible ($[\varphi]$ means φ worlds remain).
- We follow that convention for arrow updates.

Arrow updates: syntax (continued)

- Clauses of the form: $\varphi \overset{a}{\mapsto} \psi$.
- Meaning: if φ is true, then from a's point of view ψ remains possible.

Arrow updates: syntax (continued)

- Clauses of the form: $\varphi \overset{a}{\mapsto} \psi$.
- Meaning: if φ is true, then from *a*'s point of view ψ remains possible.
- Semantically: $\varphi \overset{a}{\mapsto} \psi$ means that *a*-arrow from φ world to ψ world is retained.

Arrow updates: syntax (continued)

- Clauses of the form: $\varphi \overset{a}{\mapsto} \psi$.
- Meaning: if φ is true, then from *a*'s point of view ψ remains possible.
- Semantically: $\varphi \overset{a}{\mapsto} \psi$ means that *a*-arrow from φ world to ψ world is retained.
- Arrow update consists of set of such clauses.

Arrow updates: syntax (continued)

- Clauses of the form: $\varphi \overset{a}{\mapsto} \psi$.
- Meaning: if φ is true, then from *a*'s point of view ψ remains possible.
- Semantically: $\varphi \overset{a}{\mapsto} \psi$ means that *a*-arrow from φ world to ψ world is retained.
- Arrow update consists of set of such clauses.
- Every arrow matching no clause is deleted.

Arrow updates: formally

Definition

The language of arrow update logic (AUL) is given by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box_a \varphi \mid [U]\varphi$$
$$U ::= \epsilon \mid U, \varphi \overset{a}{\mapsto} \psi$$

where $a \in A$, $p \in P$ and ϵ is the empty sequence.

Arrow updates: semantics

- $M * [U] = (W, R * [U], V)$
- $(s_1, s_2) \in R * [U]_a$ iff $(s_1, s_2) \in R_a$ and
  $\exists(\varphi \overset{a}{\mapsto} \psi) \in U : M, s_1 \models \varphi$ and $M, s_2 \models \psi$.

Satisfaction relation $\models$ is extended with

- $M, s \models [U]\varphi$ iff $M * [U], s \models \varphi$.

Arrow updates: example part II

- Example of me looking at the first card,

Arrow updates: example part II

- Example of me looking at the first card,
- For Rustam: no change, i.e., in every situation every other situation remains possible.

Arrow updates: example part II

- Example of me looking at the first card,
- For Rustam: no change, i.e., in every situation every other situation remains possible.
- Clause: $\top \overset{r}{\mapsto} \top$.

Arrow updates: example part II

- Example of me looking at the first card,
- For Rustam: no change, i.e., in every situation every other situation remains possible.
- Clause: $\top \overset{r}{\mapsto} \top$.
- For me: if r_1 is true, then I learn that $\neg r_1$ is false, so r_1 is all that remains possible.

Arrow updates: example part II

- Example of me looking at the first card,
- For Rustam: no change, i.e., in every situation every other situation remains possible.
- Clause: $\top \overset{r}{\mapsto} \top$.
- For me: if r_1 is true, then I learn that $\neg r_1$ is false, so r_1 is all that remains possible.
- Clause: $r_1 \overset{l}{\mapsto} r_1$.

Arrow updates: example part II

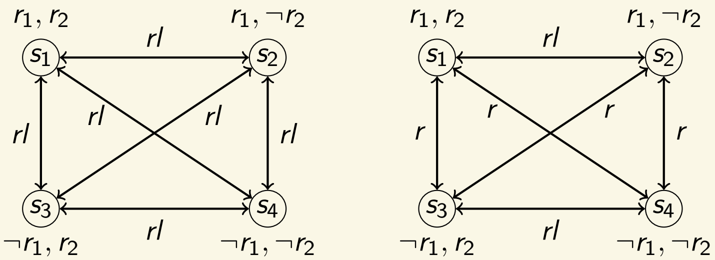
- Example of me looking at the first card,
- For Rustam: no change, i.e., in every situation every other situation remains possible.
- Clause: $\top \overset{r}{\mapsto} \top$.
- For me: if r_1 is true, then I learn that $\neg r_1$ is false, so r_1 is all that remains possible.
- Clause: $r_1 \overset{l}{\mapsto} r_1$.
- Similarly: $\neg r_1 \overset{l}{\mapsto} \neg r_1$.

Arrow updates: example part II

- Example of me looking at the first card,
- For Rustam: no change, i.e., in every situation every other situation remains possible.
- Clause: $\top \overset{r}{\mapsto} \top$.
- For me: if r_1 is true, then I learn that $\neg r_1$ is false, so r_1 is all that remains possible.
- Clause: $r_1 \overset{l}{\mapsto} r_1$.
- Similarly: $\neg r_1 \overset{l}{\mapsto} \neg r_1$.
- No further clauses: update U given by $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.

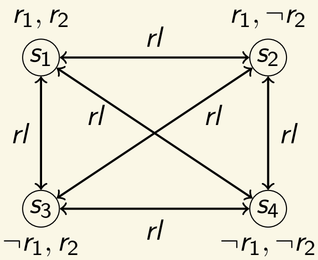
Arrow updates: example part III

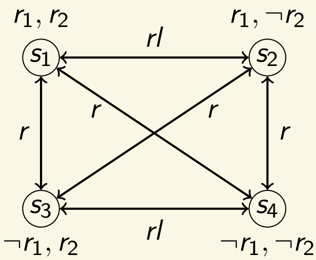
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.



Arrow updates: example part III

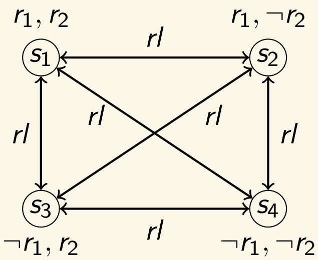
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.

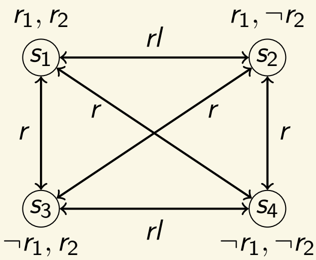




Arrow updates: example part III

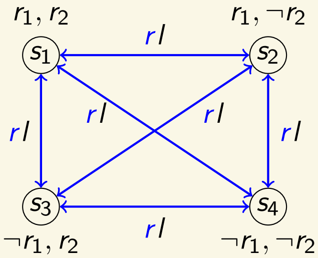
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.

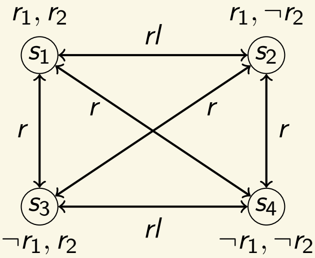




Arrow updates: example part III

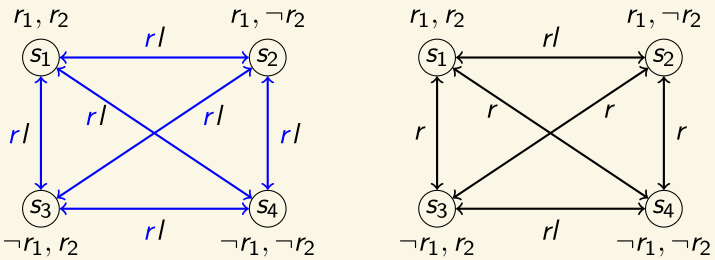
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.





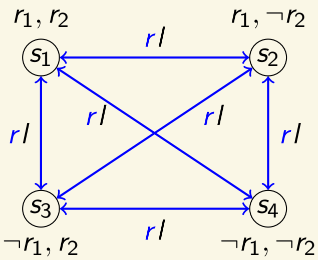
Arrow updates: example part III

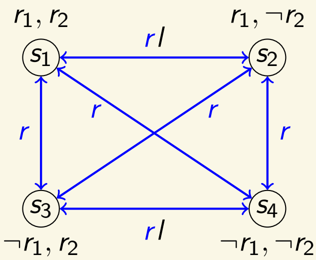
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.



Arrow updates: example part III

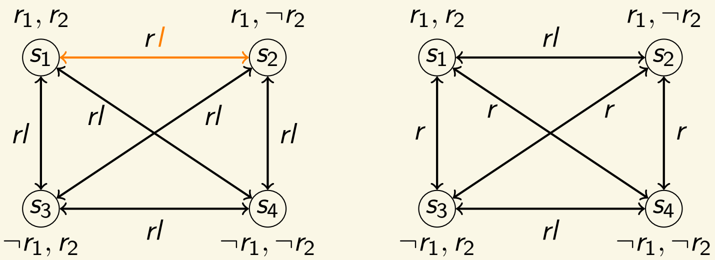
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.





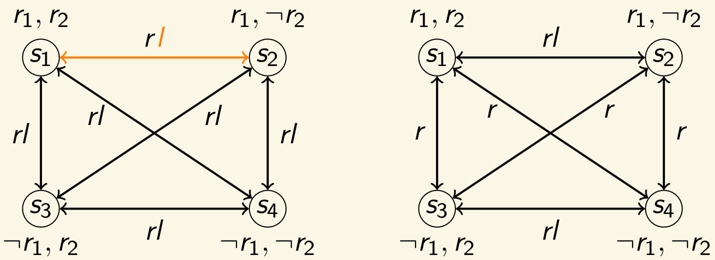
Arrow updates: example part III

- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.



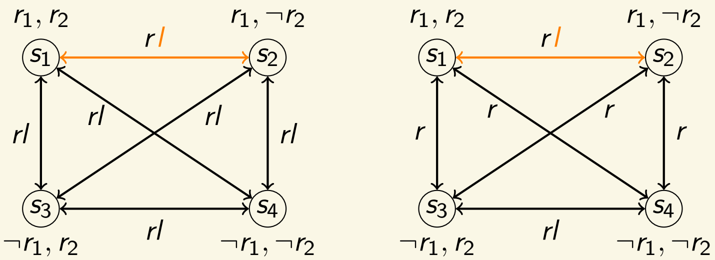
Arrow updates: example part III

- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.



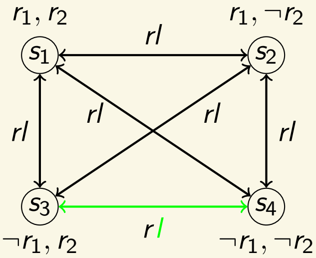
Arrow updates: example part III

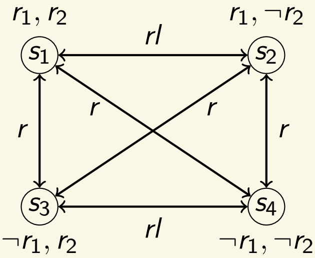
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.



Arrow updates: example part III

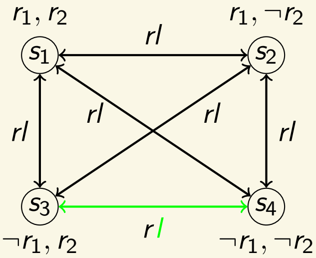
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.

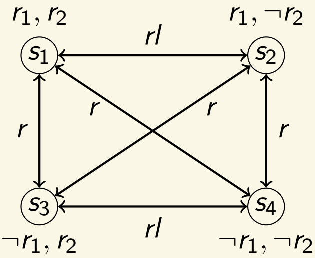




Arrow updates: example part III

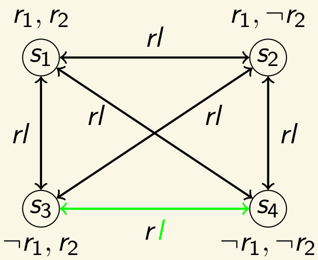
- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.

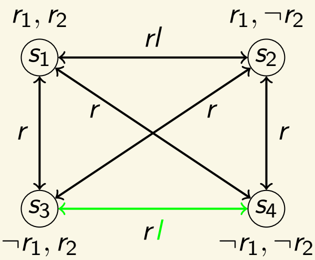




Arrow updates: example part III

- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.





Arrow updates: example part III

- We just established that $U = \{\top \overset{r}{\mapsto} \top, r_1 \overset{l}{\mapsto} r_1, \neg r_1 \overset{l}{\mapsto} \neg r_1\}$.
- We know from before: model on the left should turn into model on the right.
- Let's see why this is so.

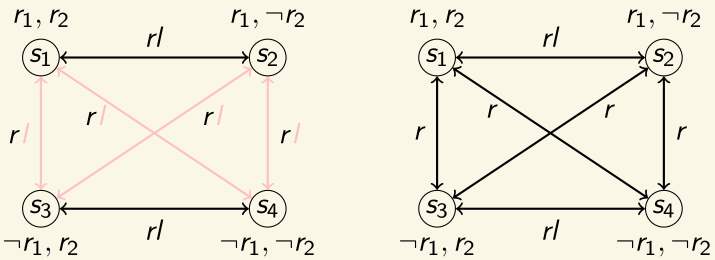


Table of Contents

Axiomatization

- In a moment, we'll discuss axiomatizations for DEL.

Axiomatization

- In a moment, we'll discuss axiomatizations for DEL.
- First, however, brief reminder of axiomatization for EL.

Axiomatization

- In a moment, we'll discuss axiomatizations for DEL.
- First, however, brief reminder of axiomatization for EL.
- Well known proof system **K**:

(Prop)	Any substitution instance of a validity of propositional logic
(K)	$\Box(\mathit{phi} \to \psi) \to (\Box\varphi \to \Box\psi)$
(Necc)	From $\vdash \varphi$, infer $\vdash \Box\varphi$
(MP)	From $\varphi \to \psi$ and φ, infer ψ

Completeness

- Proof in **K** is a finite, numbered list of formulas.
- Each line in proof is justified by (1) being a premise, (2) an axiom of **K** or (3) applying a rule of **K** to earlier line(s).

Completeness

- Proof in **K** is a finite, numbered list of formulas.
- Each line in proof is justified by (1) being a premise, (2) an axiom of **K** or (3) applying a rule of **K** to earlier line(s).
- Notation $\Gamma \vdash \varphi$.

Completeness

- Proof in **K** is a finite, numbered list of formulas.
- Each line in proof is justified by (1) being a premise, (2) an axiom of **K** or (3) applying a rule of **K** to earlier line(s).
- Notation $\Gamma \vdash \varphi$.
- Famously, **K** is sound and strongly complete.

Completeness

- Proof in **K** is a finite, numbered list of formulas.
- Each line in proof is justified by (1) being a premise, (2) an axiom of **K** or (3) applying a rule of **K** to earlier line(s).
- Notation $\Gamma \vdash \varphi$.
- Famously, **K** is sound and strongly complete.
- So, in some sense, all there is to know about basic modal logic.

Predictable

- Public announcements, arrow updates and substitutions change agents' knowledge.

Predictable

- Public announcements, arrow updates and substitutions change agents' knowledge.
- But: the old situation determines the new one.
- So it is unsurprising that whether φ holds in the new situation can be predicted from the old one.

Predictable

- Public announcements, arrow updates and substitutions change agents' knowledge.
- But: the old situation determines the new one.
- So it is unsurprising that whether φ holds in the new situation can be predicted from the old one.
- These predictions can be encoded as axioms.

Axioms for Substitutions

- The axioms for substitutions are the easiest. So we start with those.

Axioms for Substitutions

- The axioms for substitutions are the easiest. So we start with those.
- $[p := \varphi]$ sets value of p to value of φ. Hence $M * [p := \varphi], w \models p$ iff $M, w \models \varphi$.

Axioms for Substitutions

- The axioms for substitutions are the easiest. So we start with those.
- $[p := \varphi]$ sets value of p to value of φ. Hence $M * [p := \varphi], w \models p$ iff $M, w \models \varphi$.
- Result $\models [p := \varphi]p \leftrightarrow \varphi$.

Axioms for Substitutions

- The axioms for substitutions are the easiest. So we start with those.
- $[p := \varphi]$ sets value of p to value of φ. Hence $M * [p := \varphi], w \models p$ iff $M, w \models \varphi$.
- Result $\models [p := \varphi]p \leftrightarrow \varphi$.
- If σ doesn't assign a value to p then $\models [\sigma]p \leftrightarrow p$.

Axioms for Substitutions (II)

- Recall that, in modal logic, $\Box_a \neg\varphi \leftrightarrow \neg\Box_a \varphi$ characterizes *functionality* of accessibility relation.

Axioms for Substitutions (II)

- Recall that, in modal logic, $\Box_a \neg\varphi \leftrightarrow \neg\Box_a \varphi$ characterizes *functionality* of accessibility relation.
- This is because if there is only one successor, then either all successors satisfy φ or no successors satisfy φ.

Axioms for Substitutions (II)

- Recall that, in modal logic, $\Box_a \neg\varphi \leftrightarrow \neg\Box_a\varphi$ characterizes *functionality* of accessibility relation.
- This is because if there is only one successor, then either all successors satisfy φ or no successors satisfy φ.
- The update $[\sigma]$, considered as a model transformer, is also a function.

Axioms for Substitutions (II)

- Recall that, in modal logic, $\Box_a \neg\varphi \leftrightarrow \neg\Box_a\varphi$ characterizes *functionality* of accessibility relation.
- This is because if there is only one successor, then either all successors satisfy φ or no successors satisfy φ.
- The update $[\sigma]$, considered as a model transformer, is also a function.
- Hence: $\models [\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$.

Axioms for Substitutions (II)

- Recall that, in modal logic, $\Box_a \neg\varphi \leftrightarrow \neg\Box_a\varphi$ characterizes *functionality* of accessibility relation.
- This is because if there is only one successor, then either all successors satisfy φ or no successors satisfy φ.
- The update $[\sigma]$, considered as a model transformer, is also a function.
- Hence: $\models [\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$.
- Similarly: $\models [\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$.

Axioms for Substitutions (III)

- Finally: substitutions are public and do not affect distinguishability of worlds.

Axioms for Substitutions (III)

- Finally: substitutions are public and do not affect distinguishability of worlds.
- This implies that $\models [\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$.

Axioms for Substitutions (IV)

Putting it all together:

$[\sigma]p \leftrightarrow \varphi$	where $p := \varphi$ in σ
$[\sigma]p \leftrightarrow p$	where p is not assigned a value in σ
$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$	
$[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$	
$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$	

are sound axioms.

Reduction axioms

$[\sigma]p \leftrightarrow \varphi$	where $p := \varphi$ in σ
$[\sigma]p \leftrightarrow p$	where p is not assigned a value in σ
$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$	
$[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$	
$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$	

Reduction axioms

$[\sigma]p \leftrightarrow \varphi$	where $p := \varphi$ in σ
$[\sigma]p \leftrightarrow p$	where p is not assigned a value in σ
$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$	
$[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$	
$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$	

- Important property: in each axiom right-hand side has less complex formula inside scope of $[\sigma]$.

Reduction axioms

$[\sigma]p \leftrightarrow \varphi$	where $p := \varphi$ in σ
$[\sigma]p \leftrightarrow p$	where p is not assigned a value in σ
$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$	
$[\sigma](\varphi \lor \psi) \leftrightarrow ([\sigma]\varphi \lor [\sigma]\psi)$	
$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$	

- Important property: in each axiom right-hand side has less complex formula inside scope of $[\sigma]$.

Reduction axioms

$[\sigma]p \leftrightarrow \varphi$ where $p := \varphi$ in σ

$[\sigma]p \leftrightarrow p$ where p is not assigned a value in σ

$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$

$[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$

$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$

- Important property: in each axiom right-hand side has less complex formula inside scope of $[\sigma]$.

Reduction axioms

$$[\sigma]p \leftrightarrow \varphi \qquad \text{where } p := \varphi \text{ in } \sigma$$
$$[\sigma]p \leftrightarrow p \qquad \text{where } p \text{ is not assigned a value in } \sigma$$
$$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$$
$$[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$$
$$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$$

- Important property: in each axiom right-hand side has less complex formula inside scope of $[\sigma]$.

Reduction axioms

$[\sigma]p \leftrightarrow \varphi$ where $p := \varphi$ in σ

$[\sigma]p \leftrightarrow p$ where p is not assigned a value in σ

$[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi$

$[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi)$

$[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi$

- Important property: in each axiom right-hand side has less complex formula inside scope of $[\sigma]$.

Reduction axioms

$$
\begin{array}{ll}
[\sigma]p \leftrightarrow \varphi & \text{where } p := \varphi \text{ in } \sigma \\
[\sigma]p \leftrightarrow p & \text{where } p \text{ is not assigned a value in } \sigma \\
[\sigma]\neg\varphi \leftrightarrow \neg[\sigma]\varphi & \\
[\sigma](\varphi \vee \psi) \leftrightarrow ([\sigma]\varphi \vee [\sigma]\psi) & \\
[\sigma]\Box_a\varphi \leftrightarrow \Box_a[\sigma]\varphi &
\end{array}
$$

- Important property: in each axiom right-hand side has less complex formula inside scope of $[\sigma]$.
- Consequence: every formula with $[\sigma]$ is provably equivalent to one without.

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \vee q)]\Box_b \neg p$.

Using reduction axioms

- Example: $[\sigma]\Box_b \neg p$.

Using reduction axioms

- Example: $[\sigma]\Box_b\neg p$.
- $\Box_b[\sigma]\neg p$

Using reduction axioms

- Example: $[\sigma]\Box_b\neg p$.
- $\Box_b[\sigma]\neg p$
- $\Box_b\neg[\sigma]p$

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \vee q)]\Box_b \neg p$.
- $\Box_b [\sigma] \neg p$
- $\Box_b \neg [\sigma] p$

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \vee q)]\Box_b \neg p$.
- $\Box_b [\sigma] \neg p$
- $\Box_b \neg [p := [q := \Box_a \neg p](p \vee q)]p$

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \vee q)]\Box_b \neg p$.
- $\Box_b[\sigma]\neg p$
- $\Box_b \neg [p := [q := \Box_a \neg p](p \vee q)]p$
- $\Box_b \neg [q := \Box_a \neg p](p \vee q)$

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \lor q)]\Box_b \neg p$.
- $\Box_b [\sigma] \neg p$
- $\Box_b \neg [p := [q := \Box_a \neg p](p \lor q)]p$
- $\Box_b \neg [q := \Box_a \neg p](p \lor q)$
- $\Box_b \neg ([q := \Box_a \neg p]p \lor [q := \Box_a \neg p]q)$

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \vee q)]\Box_b \neg p$.
- $\Box_b[\sigma]\neg p$
- $\Box_b \neg [p := [q := \Box_a \neg p](p \vee q)]p$
- $\Box_b \neg [q := \Box_a \neg p](p \vee q)$
- $\Box_b \neg ([q := \Box_a \neg p]p \vee [q := \Box_a \neg p]q)$
- $\Box_b \neg (p \vee [q := \Box_a \neg p]q)$

Using reduction axioms

- Example: $[p := [q := \Box_a \neg p](p \vee q)]\Box_b \neg p$.
- $\Box_b [\sigma] \neg p$
- $\Box_b \neg [p := [q := \Box_a \neg p](p \vee q)] p$
- $\Box_b \neg [q := \Box_a \neg p](p \vee q)$
- $\Box_b \neg ([q := \Box_a \neg p] p \vee [q := \Box_a \neg p] q)$
- $\Box_b \neg (p \vee [q := \Box_a \neg p] q)$
- $\Box_b \neg (p \vee \Box_a \neg p)$

The many uses of reduction axioms

Reduction axioms are nice because:

The many uses of reduction axioms

Reduction axioms are nice because:

1. "Free" completeness: axiomatization for EL $+$ reduction axioms for $[\sigma] =$ axiomatization for EL$+[\sigma]$.

The many uses of reduction axioms

Reduction axioms are nice because:

1. "Free" completeness: axiomatization for EL + reduction axioms for $[\sigma]$ = axiomatization for EL+$[\sigma]$.
2. "Free" expressivity results: EL+$[\sigma]$ formulas are equivalent to EL formulas.

The many uses of reduction axioms

Reduction axioms are nice because:

1. "Free" completeness: axiomatization for EL + reduction axioms for $[\sigma]$ = axiomatization for EL+$[\sigma]$.
2. "Free" expressivity results: EL+$[\sigma]$ formulas are equivalent to EL formulas.
3. "Free" decidability: satisfiability of EL+$[\sigma]$ reduces to satisfiability of EL.

Axioms for public announcements

- We can do the same for public announcements.
- Small complication: $[\varphi]$ is a partial function. To compensate: add a bunch of $\varphi \to \cdots$ conditions.

Axioms for public announcements

- We can do the same for public announcements.
- Small complication: $[\varphi]$ is a partial function. To compensate: add a bunch of $\varphi \to \cdots$ conditions.
- $\models [\varphi]p \leftrightarrow (\varphi \to p)$

Axioms for public announcements

- We can do the same for public announcements.
- Small complication: $[\varphi]$ is a partial function. To compensate: add a bunch of $\varphi \to \cdots$ conditions.
- $\models [\varphi]p \leftrightarrow (\varphi \to p)$
- $\models [\varphi]\neg\psi \leftrightarrow (\varphi \to \neg[\varphi]\psi)$

Axioms for public announcements

- We can do the same for public announcements.
- Small complication: $[\varphi]$ is a partial function. To compensate: add a bunch of $\varphi \rightarrow \cdots$ conditions.
- $\models [\varphi]p \leftrightarrow (\varphi \rightarrow p)$
- $\models [\varphi]\neg\psi \leftrightarrow (\varphi \rightarrow \neg[\varphi]\psi)$
- $\models [\varphi](\psi_1 \vee \psi_2) \leftrightarrow ([\varphi]\psi_1 \vee [\varphi]\psi_2)$

Axioms for public announcements

- We can do the same for public announcements.
- Small complication: $[\varphi]$ is a partial function. To compensate: add a bunch of $\varphi \to \cdots$ conditions.
- $\models [\varphi]p \leftrightarrow (\varphi \to p)$
- $\models [\varphi]\neg\psi \leftrightarrow (\varphi \to \neg[\varphi]\psi)$
- $\models [\varphi](\psi_1 \vee \psi_2) \leftrightarrow ([\varphi]\psi_1 \vee [\varphi]\psi_2)$
- $\models [\varphi]\Box_a\psi \leftrightarrow (\varphi \to \Box_a[\varphi]\psi)$

Axioms for arrow updates

- Most axioms for arrow updates are simpler.

Axioms for arrow updates

- Most axioms for arrow updates are simpler.
- $\models [U]p \leftrightarrow p$
- $\models [U]\neg\varphi \leftrightarrow \neg[U]\varphi$
- $\models [U](\varphi \vee \psi) \leftrightarrow ([U]\varphi \vee [U]\psi)$

Axioms for arrow updates

- Most axioms for arrow updates are simpler.
- $\models [U]p \leftrightarrow p$
- $\models [U]\neg\varphi \leftrightarrow \neg[U]\varphi$
- $\models [U](\varphi \vee \psi) \leftrightarrow ([U]\varphi \vee [U]\psi)$
- Final axioms is more complicated, however.

Axioms for arrow updates

- Most axioms for arrow updates are simpler.
- $\models [U]p \leftrightarrow p$
- $\models [U]\neg\varphi \leftrightarrow \neg[U]\varphi$
- $\models [U](\varphi \vee \psi) \leftrightarrow ([U]\varphi \vee [U]\psi)$
- Final axioms is more complicated, however.
- $\models [U]\Box_a\varphi \leftrightarrow \bigwedge_{(\psi_1,a,\psi_2)\in U}(\psi_1 \rightarrow \Box_a(\psi_2 \rightarrow [U]\varphi))$

Reduction axioms for PAL and AUL

- Again: these are reduction axioms.
- Therefore, “free” completeness, expressivity, decidability.

Reduction axioms for PAL and AUL

- Again: these are reduction axioms.
- Therefore, “free” completeness, expressivity, decidability.
- In particular: note that EL, PAL, AUL and EL$+[\sigma]$ all have the same expressivity.

Reduction axioms for PAL and AUL

- Again: these are reduction axioms.
- Therefore, "free" completeness, expressivity, decidability.
- In particular: note that EL, PAL, AUL and EL+$[\sigma]$ all have the same expressivity.
- This is somewhat surprising: PAL, AUL and EL+$[\sigma]$ *feel* more powerful than EL.

Reduction axioms for PAL and AUL

- Again: these are reduction axioms.
- Therefore, "free" completeness, expressivity, decidability.
- In particular: note that EL, PAL, AUL and EL+$[\sigma]$ all have the same expressivity.
- This is somewhat surprising: PAL, AUL and EL+$[\sigma]$ *feel* more powerful than EL.
- And they are more powerful, in some sense. Just not in expressivity.

Table of Contents

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.
- So why do we bother?

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.
- So why do we bother?
- If $[\sigma], [\varphi], [U]$ don't add expressivity, do they add something fundamentally new?

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.
- So why do we bother?
- If $[\sigma], [\varphi], [U]$ don't add expressivity, do they add something fundamentally new?
- Three reasons.

Comparing the four logics

- In previous section we saw: EL, EL+$[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.
- So why do we bother?
- If $[\sigma], [\varphi], [U]$ don't add expressivity, do they add something fundamentally new?
- Three reasons.
  1. Succinctness. The equivalent formula in EL is typically longer.

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.
- So why do we bother?
- If $[\sigma], [\varphi], [U]$ don't add expressivity, do they add something fundamentally new?
- Three reasons.
  1. Succinctness. The equivalent formula in EL is typically longer.
  2. We can add quantification. (The main point of this course!)

Comparing the four logics

- In previous section we saw: EL, EL$+[\sigma]$, PAL, AUL all have same expressivity.
- I.e., for every formula φ in one language there is an equivalent formula φ' in other language.
- So why do we bother?
- If $[\sigma], [\varphi], [U]$ don't add expressivity, do they add something fundamentally new?
- Three reasons.
  1. Succinctness. The equivalent formula in EL is typically longer.
  2. We can add quantification. (The main point of this course!)
  3. While they have the same expressivity, their *update expressivity* differs.

Update Expressivity

- Expressivity (the normal kind) is about which sets of pointed models can be expressed, i.e., given class X of pointed models, is there a formula φ such that $[\![\varphi]\!] = X$?

Update Expressivity

- Expressivity (the normal kind) is about which sets of pointed models can be expressed, i.e., given class X of pointed models, is there a formula φ such that $[\![\varphi]\!] = X$?
- *Update* expressivity is about which model transformers can be expressed.

Update Expressivity

- Expressivity (the normal kind) is about which sets of pointed models can be expressed, i.e., given class X of pointed models, is there a formula φ such that $[\![\varphi]\!] = X$?
- *Update* expressivity is about which model transformers can be expressed.
- Given a function $f : \mathfrak{M} \to \mathfrak{M}$, is there an update e in the language such that $[\![e]\!] = f$?

A first attempt

First attempt at a definition:

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates.

A first attempt

First attempt at a definition:

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$ if:

A first attempt

First attempt at a definition:

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$ if:

For every $e_1 \in E_1$ there is an $e_2 \in E_2$ s.t. $e_1 = e_2$.

A first attempt

First attempt at a definition:

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$ if:

For every $e_1 \in E_1$ there is an $e_2 \in E_2$ s.t. $e_1 = e_2$.

Problem 1: equality too strong.

A second attempt

Definition

Let $e_1 : \mathfrak{M} \to \mathfrak{M}$ and $e_2 : \mathfrak{M} \to \mathfrak{M}$ be given. We say that e_1 and e_2 are equivalent, denoted $e_1 \sim e_2$ if for all M, w,

Definition

A second attempt

Definition

Let $e_1 : \mathfrak{M} \to \mathfrak{M}$ and $e_2 : \mathfrak{M} \to \mathfrak{M}$ be given. We say that e_1 and e_2 are equivalent, denoted $e_1 \sim e_2$ if for all M, w, the models $e_1(M, w)$ and $e_2(M, w)$ are bisimilar.

Definition

A second attempt

Definition

Let $e_1 : \mathfrak{M} \to \mathfrak{M}$ and $e_2 : \mathfrak{M} \to \mathfrak{M}$ be given. We say that e_1 and e_2 are equivalent, denoted $e_1 \sim e_2$ if for all M, w, the models $e_1(M, w)$ and $e_2(M, w)$ are bisimilar.

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$ if:

A second attempt

Definition

Let $e_1 : \mathfrak{M} \to \mathfrak{M}$ and $e_2 : \mathfrak{M} \to \mathfrak{M}$ be given. We say that e_1 and e_2 are equivalent, denoted $e_1 \sim e_2$ if for all M, w, the models $e_1(M, w)$ and $e_2(M, w)$ are bisimilar.

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$ if:

For every $e_1 \in E_1$ there is an $e_2 \in E_2$ such that $e_1 \sim e_2$.

A second attempt

Definition

Let $e_1 : \mathfrak{M} \to \mathfrak{M}$ and $e_2 : \mathfrak{M} \to \mathfrak{M}$ be given. We say that e_1 and e_2 are equivalent, denoted $e_1 \sim e_2$ if for all M, w, the models $e_1(M, w)$ and $e_2(M, w)$ are bisimilar.

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and E_2 of updates. We say that the *update expressivity* of $\mathcal{L}_1$ is at least as great as that of $\mathcal{L}_2$ if:

For every $e_1 \in E_1$ there is an $e_2 \in E_2$ such that $e_1 \sim e_2$.

Problem 2: public announcements are partial functions, not functions.

Update expressivity: the definition

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, w,

Definition

Update expressivity: the definition

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, w, if $e_1(M, w)$ exists, then $e_2(M, w)$ exists and the two pointed models are bisimilar.

Definition

Update expressivity: the definition

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, w, if $e_1(M, w)$ exists, then $e_2(M, w)$ exists and the two pointed models are bisimilar.

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and 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:

Update expressivity: the definition

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, w, if $e_1(M, w)$ exists, then $e_2(M, w)$ exists and the two pointed models are bisimilar.

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages with associated sets E_1 and 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 $e_1 \in E_1$ there is an $e_2 \in E_2$ such that $e_1 \rightsquigarrow e_2$.

Comparing update expressivity

PAL AUL EL+$[\sigma]$

EL

Comparing update expressivity

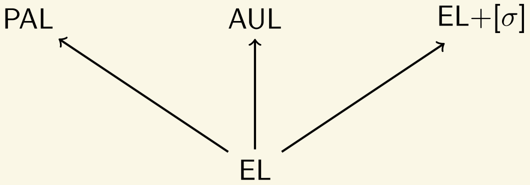
- All three update logics clearly have higher update expressivity than EL.

PAL AUL EL+$[\sigma]$

EL

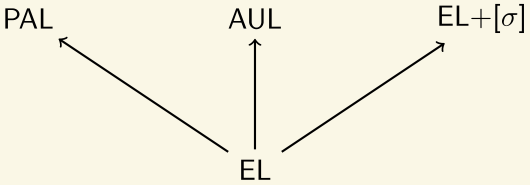
Comparing update expressivity

- All three update logics clearly have higher update expressivity than EL.



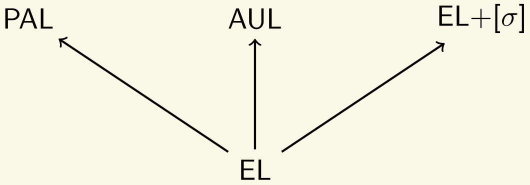
Comparing update expressivity

- All three update logics clearly have higher update expressivity than EL.
- EL+$[\sigma]$ is incomparable with PAL and AUL.



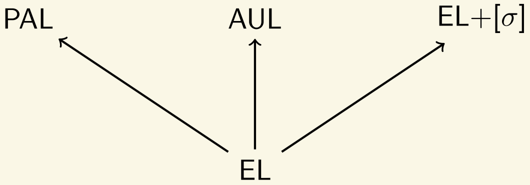
Comparing update expressivity

- All three update logics clearly have higher update expressivity than EL.
- EL+$[\sigma]$ is incomparable with PAL and AUL.
- But: $[\top \xrightarrow{A} \varphi] \rightsquigarrow [\varphi]$, so PAL $\preceq$ AUL.



Comparing update expressivity

- All three update logics clearly have higher update expressivity than EL.
- EL$+[\sigma]$ is incomparable with PAL and AUL.
- But: $[\top \xrightarrow{A} \varphi] \rightsquigarrow [\varphi]$, so PAL $\preceq$ AUL.
- No translation from arrow updates to public announcements. Therefore: PAL $\prec$ AUL.



Comparing update expressivity

- All three update logics clearly have higher update expressivity than EL.
- EL+$[\sigma]$ is incomparable with PAL and AUL.
- But: $[\top \xrightarrow{A} \varphi] \rightsquigarrow [\varphi]$, so PAL $\preceq$ AUL.
- No translation from arrow updates to public announcements. Therefore: PAL $\prec$ AUL.

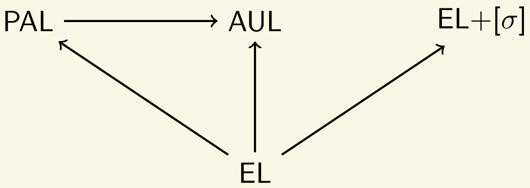


Table of Contents

Summary

Today's overall message:

Summary

Today's overall message:

- Public announcements, arrow updates and substitutions change S, R and V, respectively.

Summary

Today's overall message:

- Public announcements, arrow updates and substitutions change S, R and V, respectively.
- Updates can be seen both as functions $e : \mathfrak{M} \to \mathfrak{M}$ and as linguistic objects.

Summary

Today's overall message:

- Public announcements, arrow updates and substitutions change S, R and V, respectively.
- Updates can be seen both as functions $e : \mathfrak{M} \to \mathfrak{M}$ and as linguistic objects.
- Existence of reduction axioms shows that EL, EL$+[\sigma]$, PAL and AUL have same expressivity and are decidable.

Summary

Today's overall message:

- Public announcements, arrow updates and substitutions change S, R and V, respectively.
- Updates can be seen both as functions $e : \mathfrak{M} \to \mathfrak{M}$ and as linguistic objects.
- Existence of reduction axioms shows that EL, EL+$[\sigma]$, PAL and AUL have same expressivity and are decidable.
- But: the four logics have different update expressivity.