Logics to Describe and Change

Carlos Areces

carlos.areces@gmail.com

FaMAF, Universidad Nacional de Córdoba
Córdoba, Argentina

2011 - Petrópolis - Brasil

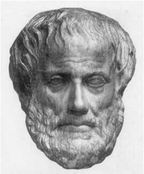
What is Logic? A Personal Perspective

What is Logic? A Personal Perspective

- What is Logic?

What is Logic? A Personal Perspective

- What is Logic?
  - A tool to model how we think ???



What is Logic? A Personal Perspective

- What is Logic?
  - A tool to model how we think ???
  - The fundamental basis of Mathematics ???

What is Logic? A Personal Perspective

- What is Logic?
  - A tool to model how we think ???
  - The fundamental basis of Mathematics ???
  - The origins of Computer Science ???

What is Logic? A Personal Perspective

- What is Logic?
  - A tool to model how we think ???
  - The fundamental basis of Mathematics ???
  - The origins of Computer Science ???
- Perhaps we are asking the wrong question. Let's try with this one:

  What is Logic used for?

What is Logic? A Personal Perspective

- What is Logic?
  - A tool to model how we think ???
  - The fundamental basis of Mathematics ???
  - The origins of Computer Science ???
- Perhaps we are asking the wrong question. Let's try with this one:

  What is Logic used for?

- I believe that a fair and acurate answer is:

  Logics are used to describe.

Next question then: To describe what?

Next question then: To describe what?

- To describe what?

Next question then: To describe what?

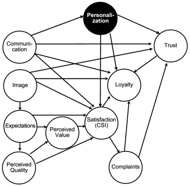
- To describe what? Anything!!!!

Next question then: To describe what?

- To describe what? **Anything!!!!**
- Agreed: With more or less detail.
  But we can use logics to describe (almost?) anything we can dream of.

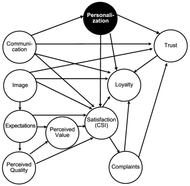
Next question then: To describe what?

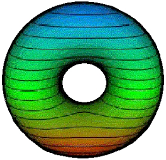
- To describe what? Anything!!!!
- Agreed: With more or less detail.
  But we can use logics to describe (almost?) anything we can dream of.



Next question then: To describe what?

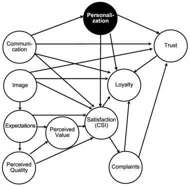
- To describe what? Anything!!!!
- Agreed: With more or less detail.
  But we can use logics to describe (almost?) anything we can dream of.

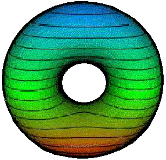


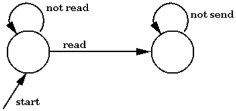


Next question then: To describe what?

- To describe what? Anything!!!!
- Agreed: With more or less detail.
  But we can use logics to describe (almost?) anything we can dream of.

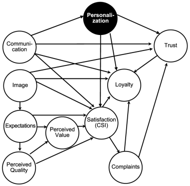


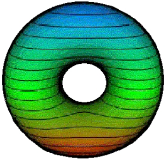


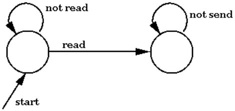


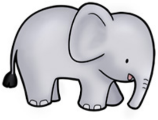
Next question then: To describe what?

- To describe what? Anything!!!!
- Agreed: With more or less detail.
  But we can use logics to describe (almost?) anything we can dream of.









Logics and Drawings

Logics and Drawings

- We say that

Logics are used to describe anything!

Logics and Drawings

- We say that

  Logics are used to describe anything!

- But then, drawings, for example, are also used to describe things.

Logics and Drawings

- We say that

  Logics are used to describe anything!

- But then, drawings, for example, are also used to describe things.
- And they are, in many ways, much better than logics
  - Drawings can be, arguably, more beautiful than logics

Logics and Drawings

- We say that

  Logics are used to describe anything!

- But then, drawings, for example, are also used to describe things.
- And they are, in many ways, much better than logics
  - Drawings can be, arguably, more beautiful than logics
  - Drawings can be done by 6 year olds
  - ...

Logics and Drawings

- We say that

  Logics are used to describe anything!

- But then, drawings, for example, are also used to describe things.
- And they are, in many ways, much better than logics
  - Drawings can be, arguably, more beautiful than logics
  - Drawings can be done by 6 year olds
  - ...
- When/for what are logics better than drawing?

Querying

Querying

- There are at least one aspect in which logics are better than drawings.

We can ask questions to a logic
and, more interestingly, obtain answers

Querying

- There are at least one aspect in which logics are better than drawings.

We can ask questions to a logic
and, more interestingly, obtain answers

- (you can probably try to get answers from a drawing, but it is usually harder than from a logic.)

Querying

- There are at least one aspect in which logics are better than drawings.

We can ask questions to a logic
and, more interestingly, obtain answers

- (you can probably try to get answers from a drawing, but it is usually harder than from a logic.)
- You can ask many different kinds of questions to a logic:
  - Satisfiability / Validity
  - Theoremhood / Axiomatization / Completeness
  - Model equivalence
  - ...

Descriptions, and more. . .

Descriptions, and more...

- We agree then that logic are good at providing descriptions.

Descriptions, and more...

- We agree then that logic are good at providing descriptions.
- Now I want to argue that they can be more than simple descriptions.
- They can actually take an active role on how the model looks like (and I'm not talking about 'leaving out the models we don't want')

Descriptions, and more. . .

- We agree then that logic are good at providing descriptions.
- Now I want to argue that they can be more than simple descriptions.
- They can actually take an active role on how the model looks like (and I'm not talking about 'leaving out the models we don't want')

  Suppose we let formulas change the model so that it actually fits the intended description. . . .

What are Modal Logics?

What are Modal Logics?

Syntax: The languages of modal logics are built with:

- The language of propositional logic: p, q, $\wedge$, $\neg$, etc.
- A collection of modal operators: $\langle r \rangle, [r], \ldots$

What are Modal Logics?

Syntax: The languages of modal logics are built with:

- The language of propositional logic: p, q, $\wedge$, $\neg$, etc.
- A collection of modal operators: $\langle r \rangle, [r]$, ...

Semantics: They are interpreted in terms of first-order relational models, that is:

$$\mathcal{M} = \langle W, \{R_r\}, \{P_r\} \rangle$$

- W is a non-empty set of elements.
- Each R_r is a (binary) relation on W.
- Each P_r is a property on W.

Intuitively speaking, it is a directed graph with labelling.

The Satisfaction Relation (for the basic language)

The Satisfaction Relation (for the basic language)

- Booleans are interpreted in the obvious way. E.g.,

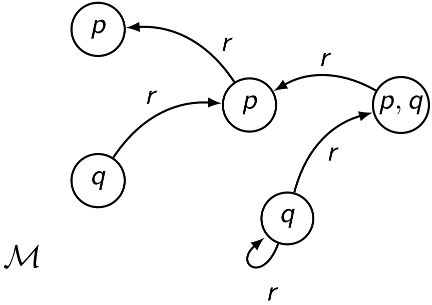
  $\mathcal{M}, w \models \varphi \wedge \varphi'$: φ and φ' holds in w

The Satisfaction Relation (for the basic language)

- Booleans are interpreted in the obvious way. E.g.,
  $\mathcal{M}, w \models \varphi \wedge \varphi'$: φ and φ' holds in w
- $\mathcal{M}, w \models \langle r \rangle \varphi$: φ holds in some R_r-related state of w.
- $\mathcal{M}, w \models [r]\varphi$: φ holds in all R_r-related states of w.

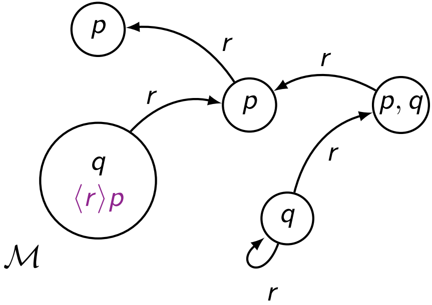
The Satisfaction Relation (for the basic language)

- Booleans are interpreted in the obvious way. E.g.,
  $\mathcal{M}, w \models \varphi \wedge \varphi'$: φ and φ' holds in w
- $\mathcal{M}, w \models \langle r \rangle \varphi$: φ holds in some R_r-related state of w.
- $\mathcal{M}, w \models [r]\varphi$: φ holds in all R_r-related states of w.



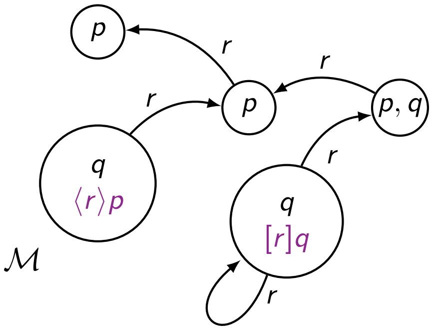
The Satisfaction Relation (for the basic language)

- Booleans are interpreted in the obvious way. E.g.,
  $\mathcal{M}, w \models \varphi \wedge \varphi'$: φ and φ' holds in w
- $\mathcal{M}, w \models \langle r \rangle \varphi$: φ holds in some R_r-related state of w.
- $\mathcal{M}, w \models [r]\varphi$: φ holds in all R_r-related states of w.



The Satisfaction Relation (for the basic language)

- Booleans are interpreted in the obvious way. E.g.,
  $\mathcal{M}, w \models \varphi \wedge \varphi'$: φ and φ' holds in w
- $\mathcal{M}, w \models \langle r \rangle \varphi$: φ holds in some R_r-related state of w.
- $\mathcal{M}, w \models [r]\varphi$: φ holds in all R_r-related states of w.



Standing in the Graph . . .

Standing in the Graph ...

- The last example shows an important characteristic of modal logics: their internal perspective.

Standing in the Graph ...

- The last example shows an important characteristic of modal logics: their internal perspective.
- Modal formulas are evaluated in a node of the graph, and describe the graph as it is seen from that point.

Standing in the Graph . . .

- The last example shows an important characteristic of modal logics: their internal perspective.
- Modal formulas are evaluated in a node of the graph, and describe the graph as it is seen from that point.
- As the Modal Logic book says:

  > *A modal formula is a little automaton standing at some state in a relational structure, and only permitted to explore the structure by making journeys to neighbouring states.*

Standing in the Graph . . .

- The last example shows an important characteristic of modal logics: their internal perspective.
- Modal formulas are evaluated in a node of the graph, and describe the graph as it is seen from that point.
- As the Modal Logic book says:

  > *A modal formula is a little automaton standing at some state in a relational structure, and only permitted to explore the structure by making journeys to neighbouring states.*

- Exploring the structure, yes, but what about changing it?

Standing in the Graph ...

- The last example shows an important characteristic of modal logics: their internal perspective.
- Modal formulas are evaluated in a node of the graph, and describe the graph as it is seen from that point.
- As the Modal Logic book says:

  > *A modal formula is a little automaton standing at some state in a relational structure, and only permitted to explore the structure by making journeys to neighbouring states.*

- Exploring the structure, yes, but what about changing it? Suppose we want to grant our little automaton the additional power to modify the structure during its exploratory trips.

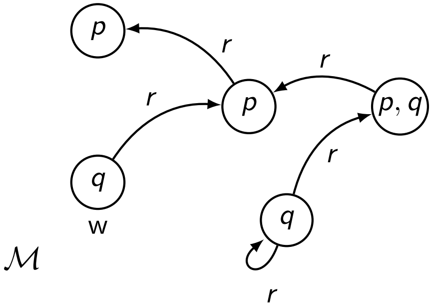
Standing in the Graph ... and Changing it

Standing in the Graph . . . and Changing it

Suppose that we have an operator $\langle \mathtt{s} \rangle$ that swaps the direction of the arrows while travelling through them.

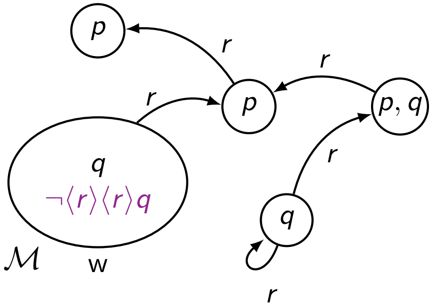
Standing in the Graph . . . and Changing it

Suppose that we have an operator $\langle \mathbf{s} \rangle$ that swaps the direction of the arrows while travelling through them.



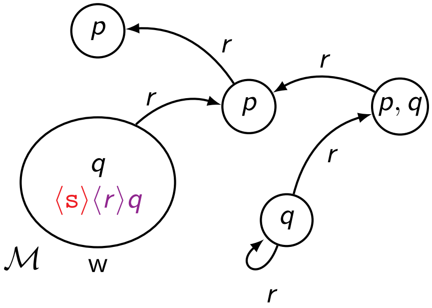
Standing in the Graph . . . and Changing it

Suppose that we have an operator $\langle \mathrm{s} \rangle$ that swaps the direction of the arrows while travelling through them.



Standing in the Graph . . . and Changing it

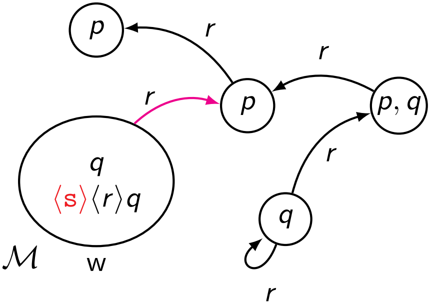
Suppose that we have an operator $\langle \mathrm{s} \rangle$ that swaps the direction of the arrows while travelling through them.



Standing in the Graph . . . and Changing it

Suppose that we have an operator $\langle \mathrm{s} \rangle$ that swaps the direction of the arrows while travelling through them.

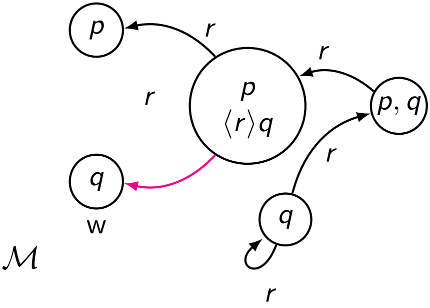
$\mathcal{M}, w \models \langle \mathrm{s} \rangle \langle r \rangle q$ iff



Standing in the Graph . . . and Changing it

Suppose that we have an operator $\langle \mathrm{s} \rangle$ that swaps the direction of the arrows while travelling through them.

$\mathcal{M}, w \models \langle \mathrm{s} \rangle \langle r \rangle q$ iff



Older than Yesterday's Paper

Older than Yesterday's Paper

- Did we just invented something that has never, ever been done before?!!!

Older than Yesterday's Paper

- Did we just invented something that has never, ever been done before?!!!
- Well . . .

Older than Yesterday's Paper

- Did we just invented something that has never, ever been done before?!!!
- Well . . . no.

Older than Yesterday's Paper

- Did we just invented something that has never, ever been done before?!!!
- Well . . . no.
- Actually, already $[r]$ and $\langle r \rangle$ change the model, if we consider the point of evaluation as part of the model.

Older than Yesterday's Paper

- Did we just invented something that has never, ever been done before?!!!
- Well . . . no.
- Actually, already $[r]$ and $\langle r \rangle$ change the model, if we consider the point of evaluation as part of the model.
- Similarly, $\forall$ and $\exists$ change the model, if we consider the assignment as part of the model.

Older than Yesterday's Paper

- Did we just invented something that has never, ever been done before?!!!
- Well . . . no.
- Actually, already $[r]$ and $\langle r \rangle$ change the model, if we consider the point of evaluation as part of the model.
- Similarly, $\forall$ and $\exists$ change the model, if we consider the assignment as part of the model.
- Now we are only allowing more daring changes.

Dynamic Logics

Dynamic Logics

- Even the idea of operators that modify the model in serious ways is not new.

Dynamic Logics

- Even the idea of operators that modify the model in serious ways is not new.
- Languages with this characteristic are sometimes called dynamic.

Dynamic Logics

- Even the idea of operators that modify the model in serious ways is not new.
- Languages with this characteristic are sometimes called dynamic.
- Examples:
  - Dynamic Epistemic Logics [Gerbrandy, van Ditmarsch, etc.] Remember the public announcement operator:

$$\mathcal{M}, w \models [\varphi]\psi \text{ iff } \mathcal{M}|_{[[\varphi]]^{\mathcal{M}}}, w \models \psi$$

Dynamic Logics

- Even the idea of operators that modify the model in serious ways is not new.
- Languages with this characteristic are sometimes called dynamic.
- Examples:
  - Dynamic Epistemic Logics [Gerbrandy, van Ditmarsch, etc.] Remember the public announcement operator:

$$\mathcal{M}, w \models [\varphi]\psi \text{ iff } \mathcal{M}|_{[[\varphi]]^{\mathcal{M}}}, w \models \psi$$

  - Dynamic Predicate Logics used for semantic construction in linguistics [Groenendijk and Stokhof]

Dynamic Logics

- Even the idea of operators that modify the model in serious ways is not new.
- Languages with this characteristic are sometimes called dynamic.
- Examples:
  - Dynamic Epistemic Logics [Gerbrandy, van Ditmarsch, etc.] Remember the public announcement operator:

$$\mathcal{M}, w \models [\varphi]\psi \text{ iff } \mathcal{M}|_{[[\varphi]]^{\mathcal{M}}}, w \models \psi$$

  - Dynamic Predicate Logics used for semantic construction in linguistics [Groenendijk and Stokhof]
  - Real time logics used in verification of real-time systems [Alur and Henzinger, etc.]

Dynamic Logics

- Even the idea of operators that modify the model in serious ways is not new.
- Languages with this characteristic are sometimes called dynamic.
- Examples:
  - Dynamic Epistemic Logics [Gerbrandy, van Ditmarsch, etc.] Remember the public announcement operator:

$$\mathcal{M}, w \models [\varphi]\psi \text{ iff } \mathcal{M}|_{[[\varphi]]^{\mathcal{M}}}, w \models \psi$$

  - Dynamic Predicate Logics used for semantic construction in linguistics [Groenendijk and Stokhof]
  - Real time logics used in verification of real-time systems [Alur and Henzinger, etc.]
- Memory Logics are very simple dynamic logics.

What are Memory Logics?

What are Memory Logics?

- They are logics with operators that can memorise and then retrieve the current point of evaluation.

What are Memory Logics?

- They are logics with operators that can memorise and then retrieve the current point of evaluation.
- Suppose that we extend model $\mathcal{M} = \langle W, \{R_r\}, \{P_r\} \rangle$ with a node memory M: $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}, M \rangle$, $M \subseteq W$.

What are Memory Logics?

- They are logics with operators that can memorise and then retrieve the current point of evaluation.
- Suppose that we extend model $\mathcal{M} = \langle W, \{R_r\}, \{P_r\} \rangle$ with a node memory M: $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}, M \rangle$, $M \subseteq W$.
  Let $\mathcal{M}[w] = \langle W, \{R_r\}, \{P_r\}, M \cup \{w\} \rangle$, define the operators

remember	$\mathcal{M}, w \models ⓡ\varphi$ iff $\mathcal{M}[w], w \models \varphi$.
known	$\mathcal{M}, w \models ⓚ$ iff $w \in M$.

What are Memory Logics?

- They are logics with operators that can memorise and then retrieve the current point of evaluation.
- Suppose that we extend model $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}\rangle$ with a node memory M: $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}, M\rangle$, $M \subseteq W$.
  Let $\mathcal{M}[w] = \langle W, \{R_r\}, \{P_r\}, M \cup \{w\}\rangle$, define the operators

  remember $\mathcal{M}, w \models ⓡ\varphi$ iff $\mathcal{M}[w], w \models \varphi$.

  known $\mathcal{M}, w \models ⓚ$ iff $w \in M$.
- Motivation:
  - It looks interesting...

What are Memory Logics?

- They are logics with operators that can memorise and then retrieve the current point of evaluation.
- Suppose that we extend model $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}\rangle$ with a node memory M: $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}, M\rangle$, $M \subseteq W$.
  Let $\mathcal{M}[w] = \langle W, \{R_r\}, \{P_r\}, M \cup \{w\}\rangle$, define the operators

remember	$\mathcal{M}, w \models ⓡ\varphi$ iff $\mathcal{M}[w], w \models \varphi$.
known	$\mathcal{M}, w \models ⓚ$ iff $w \in M$.

- Motivation:
  - It looks interesting...
  - They seem 'as simple as possible'

What are Memory Logics?

- They are logics with operators that can memorise and then retrieve the current point of evaluation.
- Suppose that we extend model $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}\rangle$ with a node memory M: $\mathcal{M} = \langle W, \{R_r\}, \{P_r\}, M\rangle$, $M \subseteq W$.
  Let $\mathcal{M}[w] = \langle W, \{R_r\}, \{P_r\}, M \cup \{w\}\rangle$, define the operators

  remember $\quad \mathcal{M}, w \models ⓡ\varphi$ iff $\mathcal{M}[w], w \models \varphi$.
  known $\quad \mathcal{M}, w \models ⓚ$ iff $w \in M$.
- Motivation:
  - It looks interesting. . .
  - They seem ‘as simple as possible’
  - (They are ‘weak’ hybrid logics)

What are Memory Logics?

What are Memory Logics?

- The (unary) remember operator ⓡ, marks the current state as being 'already visited', storing this information in M.

What are Memory Logics?

- The (unary) remember operator ⓡ, marks the current state as being 'already visited', storing this information in M.
- The (zeroary) known operator ⓚ, queries M to check if the current state has already been visited.

What are Memory Logics?

- The (unary) remember operator ⓡ, marks the current state as being 'already visited', storing this information in M.
- The (zeroary) known operator ⓚ, queries M to check if the current state has already been visited.
- Consider the formula

$$ⓡ\langle r \rangle ⓚ.$$

What are Memory Logics?

- The (unary) remember operator ⓡ, marks the current state as being 'already visited', storing this information in M.
- The (zeroary) known operator ⓚ, queries M to check if the current state has already been visited.
- Consider the formula

$$ⓡ\langle r\rangle ⓚ.$$

When evaluated in a node w of a model $\langle W, \{R_r\}, \{P_r\}, \emptyset\rangle$, it is true iff $R_r(w, w)$.

What are Memory Logics?

- The (unary) remember operator ⓡ, marks the current state as being 'already visited', storing this information in M.
- The (zeroary) known operator ⓚ, queries M to check if the current state has already been visited.
- Consider the formula

$$ⓡ\langle r \rangle ⓚ.$$

When evaluated in a node w of a model $\langle W, \{R_r\}, \{P_r\}, \emptyset\rangle$, it is true iff $R_r(w, w)$. No equivalent formula exists in the basic modal language (i.e., memory logics are strictly more expressive than the basic modal logic).

What do we Know about ⓡ and ⓚ?

Quite a bit

- Characterization in terms of Bisimulations and EF-games

What do we Know about ⓡ and ⓚ?

Quite a bit

- Characterization in terms of Bisimulations and EF-games
- Expressive Power (e.g., $\mathcal{BML} < \mathcal{ML} < \mathcal{HL}(\downarrow)$)

What do we Know about ⓡ and ⓚ?

Quite a bit

- Characterization in terms of Bisimulations and EF-games
- Expressive Power (e.g., $\mathcal{BML} < \mathcal{ML} < \mathcal{HL}(\downarrow)$)
- Complete Axiomatization (in most cases using @ and nominals)

What do we Know about ⓡ and ⓚ?

Quite a bit

- Characterization in terms of Bisimulations and EF-games
- Expressive Power (e.g., $\mathcal{BML} < \mathcal{ML} < \mathcal{HL}(\downarrow)$)
- Complete Axiomatization (in most cases using @ and nominals)
- Complexity (Undecidable)

What do we Know about ⓡ and ⓚ?

Quite a bit

- Characterization in terms of Bisimulations and EF-games
- Expressive Power (e.g., $\mathcal{BML} < \mathcal{ML} < \mathcal{HL}(\downarrow)$)
- Complete Axiomatization (in most cases using @ and nominals)
- Complexity (Undecidable)
- Tableaux

What do we Know about ⓡ and ⓚ?

Quite a bit

- Characterization in terms of Bisimulations and EF-games
- Expressive Power (e.g., $\mathcal{BML} < \mathcal{ML} < \mathcal{HL}(\downarrow)$)
- Complete Axiomatization (in most cases using @ and nominals)
- Complexity (Undecidable)
- Tableaux
- Interpolation and Beth Properties

Other Memory Operators

Other Memory Operators

Other natural operators can be defined:

Other Memory Operators

Other natural operators can be defined:

- The erase operator ⓔ, that resets the set of known states. For $\mathcal{M}$ any model, let $\mathcal{M}[\emptyset]$ be identical to $\mathcal{M}$ but $M = \emptyset$

$$\text{erase} \quad \mathcal{M}, w \models ⓔ\varphi \text{ iff } \mathcal{M}[\emptyset], w \models \varphi.$$

Other Memory Operators

Other natural operators can be defined:

- The erase operator ⓔ, that resets the set of known states.
  For $\mathcal{M}$ any model, let $\mathcal{M}[\emptyset]$ be identical to $\mathcal{M}$ but $M = \emptyset$

  $$\text{erase} \quad \mathcal{M}, w \models ⓔ\varphi \text{ iff } \mathcal{M}[\emptyset], w \models \varphi.$$

- The forget operator ⓕ, that forgets the current node from the set of known states.
  For $\mathcal{M}$ any model, let $\mathcal{M}[\text{-}w]$ be identical to $\mathcal{M}$ but w has been deleted from M

  $$\text{forget} \quad \mathcal{M}, w \models ⓕ\varphi \text{ iff } \mathcal{M}[\text{-}w], w \models \varphi.$$

Other Memory Operators

Other Memory Operators

- The remember and move $\langle\langle r \rangle\rangle$ operator defined as:

$$\mathcal{M}, w \models \langle\langle r \rangle\rangle \varphi \text{ iff } \exists v. R(w, v) \;\&\; \mathcal{M}[w], v \models \varphi.$$

Other Memory Operators

- The remember and move $\langle\langle r \rangle\rangle$ operator defined as:

$$\mathcal{M}, w \models \langle\langle r \rangle\rangle\varphi \text{ iff } \exists v.R(w, v) \ \& \ \mathcal{M}[w], v \models \varphi.$$

- If we extend $\mathcal{BML}$ with $\langle\langle r \rangle\rangle$ we get
  - Finite model property and decidability (actually PSPACE-completeness)

Other Memory Operators

- The remember and move $\langle\langle r \rangle\rangle$ operator defined as:

$$\mathcal{M}, w \models \langle\langle r \rangle\rangle \varphi \text{ iff } \exists v. R(w, v) \;\&\; \mathcal{M}[w], v \models \varphi.$$

- If we extend $\mathcal{BML}$ with $\langle\langle r \rangle\rangle$ we get
  - Finite model property and decidability (actually PSPACE-completeness)
- But $\mathcal{BML} + \langle\langle r \rangle\rangle + i$ is undecidable

It was Just a Dream

It was Just a Dream

- All the logics we discussed up to know are fragments of first-order logic.

It was Just a Dream

- All the logics we discussed up to know are fragments of first-order logic.
- Formally, there is a translation T from $\mathcal{ML}$ formulas to $\mathcal{FOL}$ formulas with only x as free variable such that

$$\mathcal{M}, w \models_{\mathcal{ML}} \varphi \text{ iff } \mathcal{M}, [x := w] \models_{\mathcal{FOL}} T(\varphi)$$

It was Just a Dream

- All the logics we discussed up to know are fragments of first-order logic.
- Formally, there is a translation T from $\mathcal{ML}$ formulas to $\mathcal{FOL}$ formulas with only x as free variable such that

$$\mathcal{M}, w \models_{\mathcal{ML}} \varphi \text{ iff } \mathcal{M}, [x := w] \models_{\mathcal{FOL}} T(\varphi)$$

- But, as in life, dreams are important!

It was Just a Dream

- All the logics we discussed up to know are fragments of first-order logic.
- Formally, there is a translation T from $\mathcal{ML}$ formulas to $\mathcal{FOL}$ formulas with only x as free variable such that

$$\mathcal{M}, w \models_{\mathcal{ML}} \varphi \text{ iff } \mathcal{M}, [x := w] \models_{\mathcal{FOL}} T(\varphi)$$

- But, as in life, dreams are important! The freedom to modify the structure lead us to new well behaved languages, i.e., new well behaved fragments of $\mathcal{FOL}$.

It was Just a Dream

- All the logics we discussed up to know are fragments of first-order logic.
- Formally, there is a translation T from $\mathcal{ML}$ formulas to $\mathcal{FOL}$ formulas with only x as free variable such that

$$\mathcal{M}, w \models_{\mathcal{ML}} \varphi \text{ iff } \mathcal{M}, [x := w] \models_{\mathcal{FOL}} T(\varphi)$$

- But, as in life, dreams are important! The freedom to modify the structure lead us to new well behaved languages, i.e., new well behaved fragments of $\mathcal{FOL}$.
- And the ways in which we can modify a structure are legion!

Relevant Articles

________, Carreiro, F., Figueira, S. and Mera, S.
Basic Model Theory for Memory Logics.
In *Proceedings of WoLLIC 2011*, 2011

________, Figueira, D., Figueira, S., and Mera, S.
The Expressive Power of Memory Logics.
Review of Symbolic Logic, Cambridge University Press, 2011.

________, Figueira, D., Gorín, D., and Mera, S.
Tableaux and Model Checking for Memory Logics.
In *Automated Reasoning with Analytic Tableaux and Related Methods*, pp. 47–61, Springer, 2009. Proceedings of Tableaux09

________, Figueira, S., and Mera, S.
Completeness results for memory logics.
In *Proceedings of LFCS'09*, the Symposium on Logic Foundations of Computer Science, pp. 16–30, 2009.