Why is modal logic decidable

Petros Potikas

NTUA

9/5/2017

Outline

About modal logic

What is modal logic?

About modal logic

What is modal logic? A *modal* is anything that qualifies the truth of a sentence.

About modal logic

What is modal logic? A *modal* is anything that qualifies the truth of a sentence.

$\Box p, \Diamond p$

About modal logic

What is modal logic? A *modal* is anything that qualifies the truth of a sentence.
$\Box p, \Diamond p$

Historically it begins from Aristotle goes to Leibniz.

About modal logic

What is modal logic? A *modal* is anything that qualifies the truth of a sentence.
$\Box p, \Diamond p$

Historically it begins from Aristotle goes to Leibniz. Continues in 1912 with C.I. Lewis and Kripke in the 60's.

About modal logic

What is modal logic? A *modal* is anything that qualifies the truth of a sentence.
$\Box p, \Diamond p$

Historically it begins from Aristotle goes to Leibniz. Continues in 1912 with C.I. Lewis and Kripke in the 60's.

Applications of ML: artificial intelligence (knowledge representation), program verification, hardware verification, and distributed computing

About modal logic

What is modal logic? A *modal* is anything that qualifies the truth of a sentence.
$\Box p, \Diamond p$

Historically it begins from Aristotle goes to Leibniz. Continues in 1912 with C.I. Lewis and Kripke in the 60's.

Applications of ML: artificial intelligence (knowledge representation), program verification, hardware verification, and distributed computing

Reason: good balance between expressive power and computational complexity

Computational problems

Two computational problems:

1. *Model-checking* problem: is a given formula true at a given state at a given Kripke structure
2. *Validity* problem: is a given formula true in all states of all Kripke structures

Computational problems

- Both problems are decidable.

Computational problems

- Both problems are decidable.
- Model-checking can be solved in linear time, while validity is PSPACE-complete.

Computational problems

- Both problems are decidable.
- Model-checking can be solved in linear time, while validity is PSPACE-complete.
- However, ML is a fragment of first order logic (FO).

Computational problems

- Both problems are decidable.
- Model-checking can be solved in linear time, while validity is PSPACE-complete.
- However, ML is a fragment of first order logic (FO).
- In first order logic, the above problems are computationally hard.

Computational problems

- Both problems are decidable.
- Model-checking can be solved in linear time, while validity is PSPACE-complete.
- However, ML is a fragment of first order logic (FO).
- In first order logic, the above problems are computationally hard.
- Only very restricted fragments of FO are decidable, typically defined in terms of bounded quantifier alternation.

Computational problems

- Both problems are decidable.
- Model-checking can be solved in linear time, while validity is PSPACE-complete.
- However, ML is a fragment of first order logic (FO).
- In first order logic, the above problems are computationally hard.
- Only very restricted fragments of FO are decidable, typically defined in terms of bounded quantifier alternation.
- But in ML we have arbitrary nesting of modalities.

Computational problems

- Both problems are decidable.
- Model-checking can be solved in linear time, while validity is PSPACE-complete.
- However, ML is a fragment of first order logic (FO).
- In first order logic, the above problems are computationally hard.
- Only very restricted fragments of FO are decidable, typically defined in terms of bounded quantifier alternation.
- But in ML we have arbitrary nesting of modalities.
- So, this cannot be captured by bounded quantifier alternation.

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.
- FO^2 is more tractable than full first-order logic.

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.
- FO^2 is more tractable than full first-order logic.
- However, this is not enough, as extensions of ML, as *computation-tree logic* (CTL) while not captured by FO^2

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.
- FO^2 is more tractable than full first-order logic.
- However, this is not enough, as extensions of ML, as *computation-tree logic* (CTL) while not captured by FO^2
- CTL can be viewed as a fragment of 2-variable fixpoint logic (FP^2)

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.
- FO^2 is more tractable than full first-order logic.
- However, this is not enough, as extensions of ML, as *computation-tree logic* (CTL) while not captured by FO^2
- CTL can be viewed as a fragment of 2-variable fixpoint logic (FP^2)
- FP^2 does not enjoy the nice computational properties of FO^2.

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.
- FO^2 is more tractable than full first-order logic.
- However, this is not enough, as extensions of ML, as *computation-tree logic* (CTL) while not captured by FO^2
- CTL can be viewed as a fragment of 2-variable fixpoint logic (FP^2)
- FP^2 does not enjoy the nice computational properties of FO^2.
- Decidability of CTL can be explained by *tree-model property*, which is enjoyed by CTL, but not by FP^2.

Modal logic and first-order logic with two variables

- Taking a closer look at ML, we see that it is a fragments of 2-variable first-order logic FO^2.
- FO^2 is more tractable than full first-order logic.
- However, this is not enough, as extensions of ML, as *computation-tree logic* (CTL) while not captured by FO^2
- CTL can be viewed as a fragment of 2-variable fixpoint logic (FP^2)
- FP^2 does not enjoy the nice computational properties of FO^2.
- Decidability of CTL can be explained by *tree-model property*, which is enjoyed by CTL, but not by FP^2.
- Finally, the tree model property leads to automata-based decision procedures.

Syntax

Definition

(The Basic Modal Language) Let $\mathbb{P} = \{\mathbb{P}_0, \mathbb{P}_1, \mathbb{P}_2, ...\}$ be a set of sentence letters, or atomic propositions. We also include two special propositions $\top$ and $\bot$ meaning 'true' and 'false' respectively. The set of well-formed formulas of modal logic is the smallest set generated by the following grammar: $\mathbb{P}_0, \mathbb{P}_1, \mathbb{P}_2, ... \mid \top \mid \bot \mid \neg A \mid A \vee B \mid A \wedge B \mid A \rightarrow B \mid \Box A \mid \Diamond A$

Examples

Modal formulas include: $\Box\bot, \mathbb{P}_0 \rightarrow \Diamond(\mathbb{P}_1 \wedge \mathbb{P}_2)$.

Truth

- A *Kripke structure* M is a tuple (S, π, R), where S is set of states (or *possible worlds*), $\pi : \mathbb{P} \rightarrow 2^S$, and R a binary relation on S.

Truth

- A *Kripke structure* M is a tuple (S, π, R), where S is set of states (or *possible worlds*), $\pi : \mathbb{P} \rightarrow 2^S$, and R a binary relation on S.
- $(M, s) \models A$, sentence A is true at s in M

Truth

- A *Kripke structure* M is a tuple (S, π, R), where S is set of states (or *possible worlds*), $\pi : \mathbb{P} \to 2^S$, and R a binary relation on S.
- $(M, s) \models A$, sentence A is true at s in M

Truth conditions:

1. $(M, s) \models \mathbb{P}_i$ iff $s \in \pi(\mathbb{P}_i)$
2. $(M, s) \models \top$
3. $(M, s) \not\models \bot$
4. $(M, s) \models \neg A$ iff not $(M, s) \models A$
5. $(M, s) \models A \vee B$ iff either $(M, s) \models A$ or, $(M, s) \models B$,or both
6. $(M, s) \models \Box A$ iff for every t, s.t. $R(s, t), (M, t) \models A$

Truth

- A *Kripke structure* M is a tuple (S, π, R), where S is set of states (or *possible worlds*), $\pi : \mathbb{P} \rightarrow 2^S$, and R a binary relation on S.
- $(M, s) \models A$, sentence A is true at s in M

Truth conditions:

1. $(M, s) \models \mathbb{P}_i$ iff $s \in \pi(\mathbb{P}_i)$
2. $(M, s) \models \top$
3. $(M, s) \not\models \bot$
4. $(M, s) \models \neg A$ iff not $(M, s) \models A$
5. $(M, s) \models A \vee B$ iff either $(M, s) \models A$ or, $(M, s) \models B$,or both
6. $(M, s) \models \Box A$ iff for every t, s.t. $R(s, t), (M, t) \models A$

- A sentence true at every possible world in every model is said to be *valid*, written $\models A$

Model-checking problem

Theorem

There is an algorithm that, given a finite Kripke structure M, a state s of M and a modal formula ϕ, determines whether $(M, s) \models \phi$ in time $O(||M|| \times |\phi|)$.

Model-checking problem

Theorem

There is an algorithm that, given a finite Kripke structure M, a state s of M and a modal formula ϕ, determines whether $(M, s) \models \phi$ in time $O(||M|| \times |\phi|)$.

$||M||$: number of states in S, and number of pairs in R

Model-checking problem

Theorem

There is an algorithm that, given a finite Kripke structure M, a state s of M and a modal formula ϕ, determines whether $(M, s) \models \phi$ in time $O(||M|| \times |\phi|)$.

$||M||$: number of states in S, and number of pairs in R

$|\phi|$: length of ϕ, number of symbols is ϕ

Model-checking problem

Theorem

There is an algorithm that, given a finite Kripke structure M, a state s of M and a modal formula ϕ, determines whether $(M, s) \models \phi$ in time $O(||M|| \times |\phi|)$.

$||M||$: number of states in S, and number of pairs in R

$|\phi|$: length of ϕ, number of symbols is ϕ

Proof.

Let $\phi_1, ..., \phi_m$ be the subformulas of ϕ listed in order of length. Thus $\phi_m = \phi$, and if ϕ_i is a subformulas of ϕ_j, then $i < j$. There are at most $|\phi|$ subformulas, so $m \leq |\phi|$. By induction on k, we can show that we can label each state s with ϕ_j or $\neg\phi_j$, for $j = 1, ..., k$, depending on whether or not ϕ_j is true in s in time $O(k||M||)$. Only interesting case is $\phi_{k+1} = \Box\phi_j$, $j < k + 1$. By induction hypothesis, we have that each state has already been labeled with ϕ_j or $\neg\phi_j$, so we know if node s can be labeled with ϕ_{k+1} or not in time $O(||M|||)$. □

Characterizing the properties of necessity

Set of valid formulas can be viewed as a characterization of the properties of necessity

Characterizing the properties of necessity

Set of valid formulas can be viewed as a characterization of the properties of necessity

Two approaches:

1. *Proof-theoretic*: all properties of necessity can be formally derived from a short list of basic properties
2. *Algorithmic*: we study algorithms that recognize properties of necessity and consider their computational complexity.

Properties of necessity

Some basic properties of necessity:

Theorem

For all formulas ϕ, ψ, and Kripke structures M:

1. *if ϕ is an instance of a propositional tautology, then $M \models \phi$*
2. *if $M \models \phi$ and $M \models \phi \to \psi$, then $M \models \psi$*
3. $M \models (\Box\phi \wedge \Box(\phi \to \psi)) \to \Box\psi$
4. *if $M \models \phi$, then $M \models \Box\phi$*

Characterizing the properties of necessity: Proof-theoretic

Consider the following axiom system $\mathcal{K}$:

- (A1) All tautologies of propositional calculus
- (A2) $(\Box\phi \wedge \Box(\phi \rightarrow \psi)) \rightarrow \Box\psi$ (Distribution axiom)
- (R1) From ϕ and $\phi \rightarrow \psi$ infer ψ (Modus ponens)
- (R2) From ϕ infer $\Box\phi$ (Generalization)

Characterizing the properties of necessity: Proof-theoretic

Consider the following axiom system $\mathcal{K}$:

- (A1) All tautologies of propositional calculus
- (A2) $(\Box\phi \wedge \Box(\phi \rightarrow \psi)) \rightarrow \Box\psi$ (Distribution axiom)
- (R1) From ϕ and $\phi \rightarrow \psi$ infer ψ (Modus ponens)
- (R2) From ϕ infer $\Box\phi$ (Generalization)

Theorem (Kripke '63)

$\mathcal{K}$ is a sound and complete axiom system.

Characterizing the properties of necessity: algorithmically

- The above characterization of the properties of necessity is not constructive.

Characterizing the properties of necessity: algorithmically

- The above characterization of the properties of necessity is not constructive.
- An algorithm that recongizes valid formulas is another characterization.

Characterizing the properties of necessity: algorithmically

- The above characterization of the properties of necessity is not constructive.
- An algorithm that recongizes valid formulas is another characterization.
- First step, if a formula is satisfiable, it is also satisfiable is a finite structure of bounded size (*bounded-model property*).

Characterizing the properties of necessity: algorithmically

- The above characterization of the properties of necessity is not constructive.
- An algorithm that recongizes valid formulas is another characterization.
- First step, if a formula is satisfiable, it is also satisfiable is a finite structure of bounded size (*bounded-model property*).
- Stronger than the *finite-model property*, which asserts that if a formula is satisfiable, then it is satisfiable in a finite structure.

Characterizing the properties of necessity: algorithmically

- The above characterization of the properties of necessity is not constructive.
- An algorithm that recongizes valid formulas is another characterization.
- First step, if a formula is satisfiable, it is also satisfiable is a finite structure of bounded size (*bounded-model property*).
- Stronger than the *finite-model property*, which asserts that if a formula is satisfiable, then it is satisfiable in a finite structure.
- This implies that formula ϕ is valid in *all* Kripke structures iff ϕ is valid in all *finite* Kripke structures.

Characterizing the properties of necessity: algorithmically

- The above characterization of the properties of necessity is not constructive.
- An algorithm that recongizes valid formulas is another characterization.
- First step, if a formula is satisfiable, it is also satisfiable is a finite structure of bounded size (*bounded-model property*).
- Stronger than the *finite-model property*, which asserts that if a formula is satisfiable, then it is satisfiable in a finite structure.
- This implies that formula ϕ is valid in *all* Kripke structures iff ϕ is valid in all *finite* Kripke structures.

Theorem (Fischer,Ladner '79)

If a modal formula ϕ is satisfiable, then ϕ is satisfiable in a Kripke structure with at most $2^{|\phi|}$ states.

Characterizing the properties of necessity: algorithmically

- From the above Theorem we can get an algorithm (not efficient) for testing validity of a formula ϕ: construct all Kripke structures with at most $2^{|\phi|}$ states and check if the formula is true in every state of each of these structures.

Characterizing the properties of necessity: algorithmically

- From the above Theorem we can get an algorithm (not efficient) for testing validity of a formula ϕ: construct all Kripke structures with at most $2^{|\phi|}$ states and check if the formula is true in every state of each of these structures.
- The "inherent difficulty" of the problem is given by the next theorem:

Theorem (Ladner '77)

The validity problem for modal logic is PSPACE-complete.

Modal logic vs. First-Order Logic

- Modal logic can be viewed as a fragment of first-order logic.

Modal logic vs. First-Order Logic

- Modal logic can be viewed as a fragment of first-order logic.
- The states in a Kripke structure correspond to domain elements in a relational structure and modalities correspond to quantifiers.

Modal logic vs. First-Order Logic

- Modal logic can be viewed as a fragment of first-order logic.
- The states in a Kripke structure correspond to domain elements in a relational structure and modalities correspond to quantifiers.
- Given a set $\mathbb{P}$ of propositional constants, let the vocabulary $\mathbb{P}^*$ constist of unary predicate q corresponding to each propositional constant q in $\mathbb{P}$, as well as binary predicate $\mathcal{R}$.

Modal logic vs. First-Order Logic

- Modal logic can be viewed as a fragment of first-order logic.
- The states in a Kripke structure correspond to domain elements in a relational structure and modalities correspond to quantifiers.
- Given a set $\mathbb{P}$ of propositional constants, let the vocabulary $\mathbb{P}^*$ constist of unary predicate q corresponding to each propositional constant q in $\mathbb{P}$, as well as binary predicate $\mathcal{R}$.
- Every Kripke structure M can be viewed as a relational structure M^* over the vocabulary $\mathbb{P}^*$.

Modal logic vs. First-Order Logic

- Modal logic can be viewed as a fragment of first-order logic.
- The states in a Kripke structure correspond to domain elements in a relational structure and modalities correspond to quantifiers.
- Given a set $\mathbb{P}$ of propositional constants, let the vocabulary $\mathbb{P}^*$ constist of unary predicate q corresponding to each propositional constant q in $\mathbb{P}$, as well as binary predicate $\mathcal{R}$.
- Every Kripke structure M can be viewed as a relational structure M^* over the vocabulary $\mathbb{P}^*$.
- Formally, a mapping from a Kriple structure $M = (S, \pi, R)$ to a relational structure M^* over the vocabulary $\mathbb{P}^*$ has:
  1. domain of M^* is S.
  2. for each propositional constant $q \in \mathbb{P}$, the interpretation of q in M^* is the set $\pi(q)$.
  3. the interpretation of the binary predicate $\mathcal{R}$, is the binary relation R.

Translation of Modal logic to First-Order Logic

A translation from modal formulas into first-order formulas over the vocabulary $\mathbb{P}^*$, so that for every modal formula ϕ there is corresponding first-order formula ϕ^* with one free variable (ranging over S):

1. $q^* = q(x)$ for a propositional constant q
2. $(\neg\phi)^* = \neg(\phi^*)$
3. $(\phi \wedge \psi)^* = (\phi^* \wedge \psi^*)$
4. $(\Box\phi)^* = (\forall y(R(x,y) \to \phi^*(x/y)))$, where y is a new variable not appearing in ϕ^* and $\phi^*(x/y)$ is the result of replacing all free occurrences of x in ϕ^* by y

Translation of Modal logic to First-Order Logic

A translation from modal formulas into first-order formulas over the vocabulary $\mathbb{P}^*$, so that for every modal formula ϕ there is corresponding first-order formula ϕ^* with one free variable (ranging over S):

1. $q^* = q(x)$ for a propositional constant q
2. $(\neg\phi)^* = \neg(\phi^*)$
3. $(\phi \wedge \psi)^* = (\phi^* \wedge \psi^*)$
4. $(\Box\phi)^* = (\forall y(R(x,y) \to \phi^*(x/y)))$, where y is a new variable not appearing in ϕ^* and $\phi^*(x/y)$ is the result of replacing all free occurrences of x in ϕ^* by y

Example

$(\Box\Diamond q)^* = \forall y(R(x,y) \to \exists z(R(y,z) \wedge q(z)))$

Theorem (vBenthem '74,'85)

1. $(M, s) \models \phi$ *iff* $(M^*, V) \models \phi^*(x)$, *for each assignment* V *s.t.* $V(x) = s$.
2. ϕ *is a valid modal formula iff* ϕ^* *is a valid first-order formula.*

ϕ^* is true of exactly the domain elements corresponding to states s for which $(M, s) \models \phi$

Translation of Modal logic to First-Order Logic

Is there a paradox?

Translation of Modal logic to First-Order Logic

Is there a paradox?

- Modal logic is essentially a first-order logic.

Translation of Modal logic to First-Order Logic

Is there a paradox?

- Modal logic is essentially a first-order logic.
- Model-checking in first-order logic is PSPACE-complete while in modal logic in linear time.

Translation of Modal logic to First-Order Logic

Is there a paradox?

- Modal logic is essentially a first-order logic.
- Model-checking in first-order logic is PSPACE-complete while in modal logic in linear time.
- Validity is robustly undecidable in first-order logic (decidable only by bounding the alternation of quantifiers), while in modal logic is PSPACE-complete.

Translation of Modal logic to First-Order Logic

Is there a paradox?

- Modal logic is essentially a first-order logic.
- Model-checking in first-order logic is PSPACE-complete while in modal logic in linear time.
- Validity is robustly undecidable in first-order logic (decidable only by bounding the alternation of quantifiers), while in modal logic is PSPACE-complete.
- Carefully examining propositional modal logic, reveals that it is a fragment of *2-variable first-order logic* (FO^2), e.g.
  $\forall x \forall y (R(x,y) \to R(y,x))$ is in FO^2, while
  $\forall x \forall y \forall z (R(x,y) \wedge R(y,z) \to R(x,z))$ is not in FO^2.

Translation of Modal logic to First-Order Logic

Is there a paradox?

- Modal logic is essentially a first-order logic.
- Model-checking in first-order logic is PSPACE-complete while in modal logic in linear time.
- Validity is robustly undecidable in first-order logic (decidable only by bounding the alternation of quantifiers), while in modal logic is PSPACE-complete.
- Carefully examining propositional modal logic, reveals that it is a fragment of *2-variable first-order logic* (FO^2), e.g. $\forall x \forall y (R(x,y) \rightarrow R(y,x))$ is in FO^2, while $\forall x \forall y \forall z (R(x,y) \wedge R(y,z) \rightarrow R(x,z))$ is not in FO^2.
- Two variables suffice to express modal logic formulas, see the above definition, where new variables are introduced only in the last clause:

Translation of Modal logic to First-Order Logic

Is there a paradox?

- Modal logic is essentially a first-order logic.
- Model-checking in first-order logic is PSPACE-complete while in modal logic in linear time.
- Validity is robustly undecidable in first-order logic (decidable only by bounding the alternation of quantifiers), while in modal logic is PSPACE-complete.
- Carefully examining propositional modal logic, reveals that it is a fragment of *2-variable first-order logic* (FO^2), e.g. $\forall x \forall y (R(x,y) \to R(y,x))$ is in FO^2, while $\forall x \forall y \forall z (R(x,y) \wedge R(y,z) \to R(x,z))$ is not in FO^2.
- Two variables suffice to express modal logic formulas, see the above definition, where new variables are introduced only in the last clause:

Example

$(\Box\Box q)^* = \forall y (R(x,y) \to \forall z (R(y,z) \to q(z)))$.

Translation of Modal logic to First-Order Logic

But re-using variables we can avoid introducing new variables. Replace the definition of ϕ^* by definition ϕ^+:

Translation of Modal logic to First-Order Logic

But re-using variables we can avoid introducing new variables. Replace the definition of ϕ^* by definition ϕ^+:

1. $q^+ = q(x)$ for a propositional constant q
2. $(\neg\phi)^+ = \neg(\phi^+)$
3. $(\phi \wedge \psi)^+ = (\phi^* \wedge \psi^+)$
4. $(\Box\phi)^+ = (\forall y(R(x,y) \rightarrow \forall x(x = y \rightarrow \phi^+)))$

Translation of Modal logic to First-Order Logic

But re-using variables we can avoid introducing new variables. Replace the definition of ϕ^* by definition ϕ^+:

1. $q^+ = q(x)$ for a propositional constant q
2. $(\neg\phi)^+ = \neg(\phi^+)$
3. $(\phi \wedge \psi)^+ = (\phi^* \wedge \psi^+)$
4. $(\Box\phi)^+ = (\forall y(R(x,y) \to \forall x(x = y \to \phi^+)))$

Example

$(\Box\Box q)^+ = \forall y(R(x,y) \to \forall x(x = y \to \forall y(R(x,y) \to \forall x(x = y \to q(x))).$

Translation of Modal logic to First-Order Logic

Theorem

1. $(M, s) \models \phi$ *iff* $(M^*, V) \models \phi^+(x)$*, for each assignment* V *s.t.* $V(x) = s$.
2. ϕ *is a valid modal formula iff* ϕ^+ *is a valid* FO^2 *formula.*

Complexity of FO^2

How hard is to evaluate truth of FO^2 formulas?

Complexity of FO^2

How hard is to evaluate truth of FO^2 formulas?

Theorem (Immerman '82, Vardi '95)

There is an algorithm that, given a relational structure M over a domain D, an FO^2-formula $\phi(x, y)$ and an assignment $V : \{x, y\} \to D$, determines whether $(M, V) \models \phi$ in time $O(||M||^2 \times |\phi|)$.

Complexity of FO^2

- Historically, Scott in 1962 showed the first decidability result for FO^2, without equality. The full class FO^2 was considered by Mortimer in 1975, who proved decidability by showing that it has the finite model proporty.

Complexity of FO^2

- Historically, Scott in 1962 showed the first decidability result for FO^2, without equality. The full class FO^2 was considered by Mortimer in 1975, who proved decidability by showing that it has the finite model proporty.
- But Mortimer's proof shows bounded-model property.

Theorem

If an FO^2-formula ϕ is satisfiable, then ϕ is satisfiable in a relational structure with at most $2^{|\phi|)}$ elements.

Complexity of FO^2

- To check the validity of a FO^2 formula ϕ, one has to consider only all structures of exponential size.

Complexity of FO^2

- To check the validity of a FO^2 formula ϕ, one has to consider only all structures of exponential size.
- Further, the translation of modal logic to FO^2 is linear, so we have Theorem 5.

Complexity of FO^2

- To check the validity of a FO^2 formula ϕ, one has to consider only all structures of exponential size.
- Further, the translation of modal logic to FO^2 is linear, so we have Theorem 5.
- Note, however, that the validity problem for FO^2 is hard for co-NEXPTIME (Fürer81) and also complete, while from Theorem 6 modal logic is PSPACE-complete.

Complexity of FO^2

- To check the validity of a FO^2 formula ϕ, one has to consider only all structures of exponential size.
- Further, the translation of modal logic to FO^2 is linear, so we have Theorem 5.
- Note, however, that the validity problem for FO^2 is hard for co-NEXPTIME (Fürer81) and also complete, while from Theorem 6 modal logic is PSPACE-complete.
- The embedding to FO^2 does not give a satisfactory explananation of the tractability of modal logic.

Reflexivity

- In epistemic logic veracity is needed, what is known is true,

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \to \phi$

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \to \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \rightarrow \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity
- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive if the relation R is reflexive. Let M_r be the class of all reflexive Kripke structures.

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \to \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity
- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive if the relation R is reflexive. Let M_r be the class of all reflexive Kripke structures.
- Axiom $\mathcal{T}$: $\Box p \to p$

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \rightarrow \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity
- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive if the relation R is reflexive. Let M_r be the class of all reflexive Kripke structures.
- Axiom $\mathcal{T}$: $\Box p \rightarrow p$

Theorem

$\mathcal{T}$ is sound and complete for M_r.

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \rightarrow \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity
- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive if the relation R is reflexive. Let M_r be the class of all reflexive Kripke structures.
- Axiom $\mathcal{T}$: $\Box p \rightarrow p$

Theorem

$\mathcal{T}$ is sound and complete for M_r.

How hard is validity under the assumption of veracity?

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \rightarrow \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity
- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive if the relation R is reflexive. Let M_r be the class of all reflexive Kripke structures.
- Axiom $\mathcal{T}$: $\Box p \rightarrow p$

Theorem

$\mathcal{T}$ is sound and complete for M_r.

How hard is validity under the assumption of veracity?

Theorem

The validity problem for modal logic in M_r is PSPACE-complete.

Reflexivity

- In epistemic logic veracity is needed, what is known is true, i.e. $\Box\phi \to \phi$
- Logical properties of necessity are related with the properties of the graph, e.g. veracity is reflexivity
- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive if the relation R is reflexive. Let M_r be the class of all reflexive Kripke structures.
- Axiom $\mathcal{T}$: $\Box p \to p$

Theorem

$\mathcal{T}$ is sound and complete for M_r.

How hard is validity under the assumption of veracity?

Theorem

The validity problem for modal logic in M_r is PSPACE-complete.

Theorem

A modal formula ϕ is valid in M_r iff the FO^2 $\forall x(R(x,x) \to \phi^+)$ is valid.

Axiom system S5

What about other properties of necessity?

Axiom system S5

What about other properties of necessity? Consider introspection:

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - “I know what I know”:

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \to \Box\Box p$.

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \to \Box\Box p$.
2. Negative introspection - "I know what I don't know":

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \rightarrow \Box\Box p$.
2. Negative introspection - "I know what I don't know": $\neg\Box p \rightarrow \Box\neg\Box p$.

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \rightarrow \Box\Box p$.
2. Negative introspection - "I know what I don't know": $\neg\Box p \rightarrow \Box\neg\Box p$.

- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive, symmetric, transitive if the relation R is reflexive, symmetric, transitive.

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \to \Box\Box p$.
2. Negative introspection - "I know what I don't know": $\neg\Box p \to \Box\neg\Box p$.

- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive, symmetric, transitive if the relation R is reflexive, symmetric, transitive.
- Let M_{rst} be the class of all reflexive, symmetric and transitive Kripke structures.

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \to \Box\Box p$.
2. Negative introspection - "I know what I don't know": $\neg\Box p \to \Box\neg\Box p$.

- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive, symmetric, transitive if the relation R is reflexive, symmetric, transitive.
- Let M_{rst} be the class of all reflexive, symmetric and transitive Kripke structures.
- Let $S5$ be the axiom system obtained from T by adding the two rules of introspection.

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \rightarrow \Box\Box p$.
2. Negative introspection - "I know what I don't know": $\neg\Box p \rightarrow \Box\neg\Box p$.

- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive, symmetric, transitive if the relation R is reflexive, symmetric, transitive.
- Let M_{rst} be the class of all reflexive, symmetric and transitive Kripke structures.
- Let $S5$ be the axiom system obtained from T by adding the two rules of introspection.

Theorem

1. *$S5$ is sound and complete for M_{rst}.*
2. *The validity problem for $S5$ is NP-complete.*

Axiom system S5

What about other properties of necessity? Consider introspection:

1. Positive introspection - "I know what I know": $\Box p \to \Box\Box p$.
2. Negative introspection - "I know what I don't know": $\neg\Box p \to \Box\neg\Box p$.

- A Kripke structure $M = (S, \pi, R)$ is said to be reflexive, symmetric, transitive if the relation R is reflexive, symmetric, transitive.
- Let M_{rst} be the class of all reflexive, symmetric and transitive Kripke structures.
- Let $S5$ be the axiom system obtained from T by adding the two rules of introspection.

Theorem

1. *$S5$ is sound and complete for M_{rst}.*
2. *The validity problem for $S5$ is NP-complete.*

Symmetry can be expressed by FO^2, $\forall x, y(R(x,y) \to R(y,x))$, while transitivity cannot $\forall x, y, z(R(x,y) \wedge R(y,z) \to R(x,z))$.

About decidability of modal logic

- The validity in a modal logic is typically decidable. It is very hard to find a modal logic, where validity is undecidable.
- The translation to FO^2 provides a partial explanation why modal logic is decidable.