Decidable fragments of first order modal logic

R. Ramanujam

The Institute of Mathematical Sciences, Chennai, India

jam@imsc.res.in

First words . . .

- Thanks, to Tobias Kappé and ILLC for the invitation.

First words . . .

- Thanks, to Tobias Kappé and ILLC for the invitation.
- Please do not wait until the end to ask questions, interruptions are welcome any time.

First words ...

- Thanks, to Tobias Kappé and ILLC for the invitation.
- Please do not wait until the end to ask questions, interruptions are welcome any time.
- The work reported here is joint with Anantha Padmanabha (IRIF, Paris) and Yanjing Wang (PKU, Beijing).

Summary-1

Church - Turing 1936: First order logic is undecidable.

- The classical decision problem: identify the decidable syntactic fragments of first order logic. A successful project of the twentieth century.
- Syntactic restrictions: quantifier prefix classes, restrict number of variables, scope of quantifiers, etc.
- Semantic restrictions: constraints on models by fixing interpretation of predicates; theories of order, arithmetical theories, algebraic theories, combinatorial theories, etc.
- Once we find decidable fragments, we seek to extend them with non-FO-definable constructs maintaining decidability: e.g. fixed-point extensions, set quantification.

Summary-2

Propositional modal logics are extensively used in computer science for specification and verification.

- Many extensions of modal logics are decidable.
- Vardi, 1996: Why are modal logics so robustly decidable ?
- Perhaps because they sit inside the two-variable fragment of First order logic (which is decidable)?
- Andreka, van Benthem, Nemeti: Because they correspond to a guarded fragment of First order logic.

Summary-3

Kripke 1962: First order modal logic (FOML) is undecidable, even with a single monadic predicate, with no equality, constants or function symbols.

- Fischer-Servi *et al*, Segerburg 1978: One-variable fragment is decidable.
- In the last few years: the monodic fragment, some bundled fragments and fragments of Term-modal logics (guarded, two-variable) are decidable.
- The good news: these results indicate that there is plenty out there for those who care to dig!
- Proceed with caution, though: even addition of a few constants can make the big difference.

First order modal logic

A theatre in which numerous philosophical controversies have been played out.

- Every element is dominated by another: A good first order sentence.

First order modal logic

A theatre in which numerous philosophical controversies have been played out.

- Every element is dominated by another: A good first order sentence.
- All processes have terminated: a contingent, but stable proposition.

First order modal logic

A theatre in which numerous philosophical controversies have been played out.

- Every element is dominated by another: A good first order sentence.
- All processes have terminated: a contingent, but stable proposition.
- Every request is eventually granted: modal proposition, interpreted as temporal or reachability.

First order modal logic

A theatre in which numerous philosophical controversies have been played out.

- Every element is dominated by another: A good first order sentence.
- All processes have terminated: a contingent, but stable proposition.
- Every request is eventually granted: modal proposition, interpreted as temporal or reachability.
- Every dominated element can become the dominator:

$$\forall x.[(\exists y.x < y) \supset \Diamond(\forall y.x \geq y)]$$

Propositional modal logic

The extension of propositional logic with a unary operator.

- Syntax:

$$p \in P \mid \neg\alpha \mid \alpha \vee \beta \mid \Box\alpha$$

- $\Box\alpha$ is read as α holds necessarily.
- Its dual, $\Diamond\alpha = \neg\Box\neg\alpha$ is read as α holds possibly.

Possible worlds semantics

Also called Kripke Structures: $M = (W, R, V)$:

- $R \subseteq (W \times W)$, $V : W \to 2^P$.

Possible worlds semantics

Also called Kripke Structures: $M = (W, R, V)$:

- $R \subseteq (W \times W)$, $V : W \to 2^P$.
- $M, w \models p$ if $p \in V(w)$, for $p \in P$.
- $M, w \models \Box\alpha$ if for all w' such that $w\ R\ w'$, $M, w' \models \alpha$.
- It is easily seen that $M, w \models \Diamond\alpha$ if for some w' such that $w\ R\ w'$, $M, w' \models \alpha$.
- α is satisfiable if there exists a model $M = (W, R, V)$ and $w \in W$ such that $M, w \models \alpha$.

Good properties

Has good model theoretic and algorithmic properties.

- A fragment of first order logic.
- Map α to α^* of FOL:

$$\Diamond\alpha \longrightarrow \exists y : (E(x,y) \wedge \alpha^*(y))$$

$$\Box\alpha \longrightarrow \forall y : (E(x,y) \supset \alpha^*(y))$$

- Satisfiability: PSpace-complete.
- Model checking: $O(\mathcal{K} \cdot \alpha)$.

Limitations of modal logic

Modal logic is very weak in terms of expressive power.

- No equality: We cannot say that both an a-transition and b-transition from the current state lead us to the same state.
- Bounded quantification: We cannot say that a property holds in all states.
- New transitions not definable: For instance, we cannot define $E(x, y) = E_a(y, x) \wedge E_b(y, x)$.

More limitations

More on the list of complaints.

- No counting: We cannot say that there is at most one a-transition from the current state (and hence cannot distinguish deterministic systems from nondeterministic ones).
- No recursion: We can look only at a bounded number of transition steps. This is a limitation shared by FOL as well.

And yet, modal logic is interesting, on many counts.

In praise of modal logic

It has interesting model theoretic properties.

- Invariance under bisimulation:

$$(\mathcal{K}, w \models \alpha \wedge (\mathcal{K}, w) \sim (\mathcal{K}', w') \Longrightarrow (\mathcal{K}', w') \models \alpha$$

- In fact, ML is the bisimulation invariant fragment of FOL.
- It has the finite model property.
- It has the tree model property.

Extensions

Numerous extensions of ML, designed to overcome the limitations mentioned, still with similar model theoretic and algorithmic properties.

- $PDL = ML +$ transitive closure.
- $LTL = ML +$ temporal operators on paths.
- $CTL = ML +$ temporal operators on paths $+$ path quantification.
- μ-calculus: encompasses these and others like game logics and description logics.

Robustness

All these extensions have good algorithmic properties.
The following hold for the μ-calculus, which encompasses most modal logics of computation.

- Satisfiability is Exptime-complete.
- Efficient model checking for many subclasses; in general, is in $NP \cap co - NP$.
- Bisimulation invariant fragment of monadic second order logic.

Vardi's question

- Vardi, 1996: Why are modal logics so robustly decidable ?
- The standard translation from ML to FO does not need more than two free variables.
- Traditionally, this has been used as an explanation for why ML has good properties.
- Is this explanation convincing ?

Fixed variable FO

FO^k: relational fragment of FOL with only k free variables.

- "There exists a path of length 17" is in FO^2:

  $$\exists x \exists y (E(x,y) \wedge \exists x (E(x,y) \wedge \exists y (E(x,y) \wedge \ldots \exists y E(x,y)) \ldots))$$

- The satisfiability problem is undecidable for FO^k, for all $k \geq 3$.
- This is true even for most of the prefix classes.

Two variable FO

- Scott 1962: FO^2 without equality can be reduced to the Gödel class and is hence decidable.
- Mortimer 1975: FO^2 has the finite model property, and is decidable.
- Grädel, Kolaitis, Vardi, 1997: FO^2 satisfiability is NExptime complete. (Lower bound essentially from Fürer 1981.)
- FO^2 is not nearly as robustly decidable as modal logic, lacks the tree model property: consider $\forall x \forall y. E(x, y)$.

A closer look

A closer look at the translation from ML to FOL shows not only the use of two variable logic, but also $\exists x.(E_a(x, y) \wedge \ldots)$ and $\forall x.(E_a(x, y) \implies \ldots)$.

- Thus quantifiers are always relativized by atoms in the modal fragment of FOL.
- Each subformula can "speak" only about elements that are 'close together' or guarded.
- Guarded fragment: Quantification is of the form: $\exists x.(\alpha(x, y) \wedge \phi(x, y))$ and $\forall x.(\alpha(x, y) \implies \phi(x, y))$. α is atomic and contains all the free variables in ϕ.

A challenge

- Andréka, van Benthem, Nemeti 1998: The guarded nature of quantification in modal logics is the "real" reason for their good algorithmic and model theoretic properties.
- Results proved since then provide some positive evidence.

Natural directions

All this wisdom suggests similar approaches to First order modal logic.

- We would like to combine the best practices of FO and the elegances of ML.

Natural directions

All this wisdom suggests similar approaches to First order modal logic.

- We would like to combine the best practices of FO and the elegances of ML.
- Unfortunately, FOML seems to combine the worst of the two, even in its simplest versions.

First order logic

Let Var denote the set of variables. A vocabulary is a pair $(C, \mathcal{P})$, where C is a set of constant symbols and $\mathcal{P}$ is a set of predicate symbols with arity. Let $T = Var \cup C$ denote the set of terms.

- Syntax:

$$P^m(t_1, \ldots, t_m) \mid t = t' \mid \neg\alpha \mid \alpha \vee \beta \mid \forall x.\alpha$$

- Model: $M = (D, \iota, \pi)$ where $\pi : Var \to D$, $\iota_c : C \to D$ and ι_P maps predicate symbol P^m to a map $D^m \to \{0, 1\}$.
- $\hat{\pi} : T \to D$: $\hat{\pi} = \iota_c \cup \pi$.
- $M \models P^m(t_1, \ldots, t_m)$ iff $\iota_P(P^m)(\hat{\pi}(t_1), \ldots, \hat{\pi}(t_m)) = 1$.
- $M \models \forall x.\alpha$ if for all $d \in D$, $M_{[x \to d]} \models \alpha$.

First order modal logic

The natural combination of First order and modal logics.

- Syntax:

$$P^m(t_1, \ldots, t_m) \mid t = t' \mid \neg\alpha \mid \alpha \vee \beta \mid \forall x.\alpha \mid \Box\alpha$$

- But the semantics is more complicated now!
- With every world we need to associate a first order structure, and interpret terms as elements of that structure.
- `Statutory warning`: This can get quite chaotic.

Coherence across worlds

Interpretations as well as variable assignments need some coherence.

- Is it reasonable to fix a single domain D for the entire 'universe' of possibilities?

Coherence across worlds

Interpretations as well as variable assignments need some coherence.

- Is it reasonable to fix a single domain D for the entire 'universe' of possibilities?
- Constant domain interpretation, as opposed to Varying domain interpretations: in the latter all quantification is over "current" domain.

Coherence across worlds

Interpretations as well as variable assignments need some coherence.

- Is it reasonable to fix a single domain D for the entire 'universe' of possibilities?
- Constant domain interpretation, as opposed to Varying domain interpretations: in the latter all quantification is over "current" domain.
- But how do you interpret (even) $\Box(P(x) \vee \neg P(x))$, where x is free? Suppose that x evaluates to d in the current world, but d does not exist in an accessible world.
- One solution is to impose a monotonicity condition. If d exists at w and wRw' then d exists at w'.

Simplest semantics

Constant domain interpretations generalize smoothly from modal logics.

- Model $M = (W, D, R, \iota, \rho, \pi)$ with $\iota : C \to D$, $\pi : Var \to D$ and ρ_P maps predicate symbol P^m to a map $(W \times D^m) \to \{0, 1\}$.

Simplest semantics

Constant domain interpretations generalize smoothly from modal logics.

- Model $M = (W, D, R, \iota, \rho, \pi)$ with $\iota : C \to D$, $\pi : Var \to D$ and ρ_P maps predicate symbol P^m to a map $(W \times D^m) \to \{0, 1\}$.
- Is the formula $\forall x.\Box\alpha \supset \Box\forall x.\alpha$ valid? (Barcan formula)

Simplest semantics

Constant domain interpretations generalize smoothly from modal logics.

- Model $M = (W, D, R, \iota, \rho, \pi)$ with $\iota : C \to D$, $\pi : Var \to D$ and ρ_P maps predicate symbol P^m to a map $(W \times D^m) \to \{0, 1\}$.
- Is the formula $\forall x.\Box\alpha \supset \Box\forall x.\alpha$ valid? (Barcan formula)
- Is the formula $\Box\forall x\alpha \supset \forall x.\Box\alpha$ valid?

Simplest semantics

Constant domain interpretations generalize smoothly from modal logics.

- Model $M = (W, D, R, \iota, \rho, \pi)$ with $\iota : C \to D$, $\pi : \mathit{Var} \to D$ and ρ_P maps predicate symbol P^m to a map $(W \times D^m) \to \{0, 1\}$.
- Is the formula $\forall x.\Box\alpha \supset \Box\forall x.\alpha$ valid? (Barcan formula)
- Is the formula $\Box\forall x\alpha \supset \forall x.\Box\alpha$ valid?
- The formula $\forall x.\Box(\exists y.x = y)$ is valid.

Undecidability

Let $FO(Q^2)$ be the set of first order logic formulas (without equality) over the single binary predicate Q^2. Its satisfiability problem is undecidable Gödel 1933.

Undecidability

Let $FO(Q^2)$ be the set of first order logic formulas (without equality) over the single binary predicate Q^2. Its satisfiability problem is undecidable Gödel 1933.

- Kripke 1962 reduces this problem to satisfiability of FOML formulas with unary predicates.

Undecidability

Let $FO(Q^2)$ be the set of first order logic formulas (without equality) over the single binary predicate Q^2. Its satisfiability problem is undecidable Gödel 1933.

- Kripke 1962 reduces this problem to satisfiability of FOML formulas with unary predicates.
- $\tau(Q(x,y)) = \Diamond(P(x) \wedge R(y))$.
- $\tau(\neg\alpha) = \neg(\tau(\alpha))$.
- $\tau(\alpha \vee \beta) = \tau(\alpha) \vee \tau(\beta)$.
- $\tau(\exists x.\alpha) = \exists x.\tau(\alpha))$.
- It is easy to see that α is FO-satisfiable iff $\tau(\alpha)$ is FOML-satisfiable.

The tale of woe

Wolter and Zakharyaschev 2001 lament:

- The monadic fragment of practically all predicate modal logics is undecidable.
- The two variable fragment of practically all predicate modal logics is undecidable, even with constant domain interpretations, without equality and constants.
- This leaves only the inexpressive one variable fragment as decidable.

The monodic fragment

Wolter and Zakharyaschev study the monodic fragment.

- *All undecidability proofs of modal predicate logics exploit formulas of the form 2 (x; y) in which the necessity operator applies to subformulas of more than one free variable; in fact, such formulas play an essential role in the reduction of undecidable problems to those fragments.*

The monodic fragment

Wolter and Zakharyaschev study the monodic fragment.

- *All undecidability proofs of modal predicate logics exploit formulas of the form 2 (x; y) in which the necessity operator applies to subformulas of more than one free variable; in fact, such formulas play an essential role in the reduction of undecidable problems to those fragments.*
- Monodic formulas are those in which only one variable may occur free in the scope of any modality.

The monodic fragment

Wolter and Zakharyaschev study the monodic fragment.

- *All undecidability proofs of modal predicate logics exploit formulas of the form 2 (x; y) in which the necessity operator applies to subformulas of more than one free variable; in fact, such formulas play an essential role in the reduction of undecidable problems to those fragments.*
- Monodic formulas are those in which only one variable may occur free in the scope of any modality.
- They show that if we consider most well-behaved decidable fragments of FO, then their monodic lifting to FOML is decidable.

Modal scope

Monodic formulas look suspiciously like one-variable formulas but they are not; they are more expressive.

- The Barcan formula $\forall x.\Box\alpha \supset \Box\forall x.\alpha$ is in 1-variable fragment which is contained in the monodic fragment.

Modal scope

Monodic formulas look suspiciously like one-variable formulas but they are not; they are more expressive.

- The Barcan formula $\forall x.\Box\alpha \supset \Box\forall x.\alpha$ is in 1-variable fragment which is contained in the monodic fragment.
- $\Diamond(P(x) \wedge \exists y.Q(x, y))$ is monodic but not 1-variable.

Modal scope

Monodic formulas look suspiciously like one-variable formulas but they are not; they are more expressive.

- The Barcan formula $\forall x.\Box\alpha \supset \Box\forall x.\alpha$ is in 1-variable fragment which is contained in the monodic fragment.
- $\Diamond(P(x) \wedge \exists y.Q(x, y))$ is monodic but not 1-variable.
- $\exists x.\forall y.R(x, y)$ is a monodic sentence but not expressible in the 1-variable fragment.

The crucial idea

When we work only with monodic formulas, modal subformulas contain at most one free variable.

- Consider a formula $\Diamond\phi$, where ϕ has no modalities. This is an FO formula with one free variable.

The crucial idea

When we work only with monodic formulas, modal subformulas contain at most one free variable.

- Consider a formula $\Diamond\phi$, where ϕ has no modalities. This is an FO formula with one free variable.
- Build a quasi-model for ϕ with some witness for the free variable. We can build boundedly many such quasi-models with disjoint domains.

The crucial idea

When we work only with monodic formulas, modal subformulas contain at most one free variable.

- Consider a formula $\Diamond\phi$, where ϕ has no modalities. This is an FO formula with one free variable.
- Build a quasi-model for ϕ with some witness for the free variable. We can build boundedly many such quasi-models with disjoint domains.
- Now we can set up an argument by induction on modal depth, building the model level by level from the "leaves" to the root.

The crucial idea

When we work only with monodic formulas, modal subformulas contain at most one free variable.

- Consider a formula $\Diamond\phi$, where ϕ has no modalities. This is an FO formula with one free variable.
- Build a quasi-model for ϕ with some witness for the free variable. We can build boundedly many such quasi-models with disjoint domains.
- Now we can set up an argument by induction on modal depth, building the model level by level from the "leaves" to the root.
- The realised types need to be combined carefully. For instance consider the formula $\exists y.(\Box P(y) \land \Diamond\exists x.\neg P(x))$.

Bundling modalities

We see that in the undecidability proof we used the modality as an additional quantifier. The idea of bundling modalities and quantifiers is to limit this capability.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x \Box \alpha \mid \forall x \Box \alpha$$

Bundling modalities

We see that in the undecidability proof we used the modality as an additional quantifier. The idea of bundling modalities and quantifiers is to limit this capability.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x \Box \alpha \mid \forall x \Box \alpha$$

- It is quite expressive: $\exists x \Box \neg \exists y \Box R(x, y)$: There is a king element such that after any update, no element is sure to dominate it later.

Bundling modalities

We see that in the undecidability proof we used the modality as an additional quantifier. The idea of bundling modalities and quantifiers is to limit this capability.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x \Box \alpha \mid \forall x \Box \alpha$$

- It is quite expressive: $\exists x \Box \neg \exists y \Box R(x, y)$: There is a king element such that after any update, no element is sure to dominate it later.
- $\forall x \Diamond . \exists y \Box R(x, y)$: Every element can be updated in such a way that another can necessarily dominate it.

Bundling modalities

We see that in the undecidability proof we used the modality as an additional quantifier. The idea of bundling modalities and quantifiers is to limit this capability.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x \Box \alpha \mid \forall x \Box \alpha$$

- It is quite expressive: $\exists x \Box \neg \exists y \Box R(x, y)$: There is a king element such that after any update, no element is sure to dominate it later.
- $\forall x \Diamond . \exists y \Box R(x, y)$: Every element can be updated in such a way that another can necessarily dominate it.
- The $\exists x \Box$ fragment was developed by Yanjing Wang in the context of epistemic logic to study Knowing how, and he went on to unify many such modalities.

News on bundling

Once we have bundled modalities, we can freely allow relations of arbitrary arity, and drop variable restrictions.

- The $\exists\Box$ bundle behaves better than the $\forall\Box$ bundle.

News on bundling

Once we have bundled modalities, we can freely allow relations of arbitrary arity, and drop variable restrictions.

- The $\exists\Box$ bundle behaves better than the $\forall\Box$ bundle.
- The latter is undecidable for constant domain interpretations even with only monadic predicates. (The Kripke coding, with some subtlety.)

News on bundling

Once we have bundled modalities, we can freely allow relations of arbitrary arity, and drop variable restrictions.

- The $\exists\Box$ bundle behaves better than the $\forall\Box$ bundle.
- The latter is undecidable for constant domain interpretations even with only monadic predicates. (The Kripke coding, with some subtlety.)
- The former is PSpace-complete for constant domains even allowing arbitrary predicates.

News on bundling

Once we have bundled modalities, we can freely allow relations of arbitrary arity, and drop variable restrictions.

- The $\exists\Box$ bundle behaves better than the $\forall\Box$ bundle.
- The latter is undecidable for constant domain interpretations even with only monadic predicates. (The Kripke coding, with some subtlety.)
- The former is PSpace-complete for constant domains even allowing arbitrary predicates.
- Interestingly the fragment with both bundles is PSpace-complete over varying domain with arbitrary predicates.

Varying domains

We can build a tableau procedure for varying domain semantics.

- Increasing domain semantics enables us to easily add new witnesses as we need.

Varying domains

We can build a tableau procedure for varying domain semantics.

- Increasing domain semantics enables us to easily add new witnesses as we need.
- One complication: we need to add witnesses for existential quantifiers and successor worlds simultaneously, as any decision for one affects the choice of the other.

Varying domains

We can build a tableau procedure for varying domain semantics.

- Increasing domain semantics enables us to easily add new witnesses as we need.
- One complication: we need to add witnesses for existential quantifiers and successor worlds simultaneously, as any decision for one affects the choice of the other.
- We can then show that the $\exists\Box$ bundle cannot distinguish between constant and increasing domains, so we can "guess" sufficiently many witnesses at one go and use them as we need.

Term-modal logics

Introduced by Fitting, Thalmann and Voronkov in 2001.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x.\alpha \mid \Box_x.\alpha$$

Term-modal logics

Introduced by Fitting, Thalmann and Voronkov in 2001.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x.\alpha \mid \Box_x.\alpha$$

- $\exists x.\forall y.(Wit(x) \supset \Box_x Killed(x, Mary))$: All witnesses know who killed Mary.

Term-modal logics

Introduced by Fitting, Thalmann and Voronkov in 2001.

- Consider the syntax:

$$P(\bar{x}) \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x.\alpha \mid \Box_x.\alpha$$

- $\exists x.\forall y.(Wit(x) \supset \Box_x Killed(x, Mary))$: All witnesses know who killed Mary.
- Note that we now have a logic with an unbounded vocabulary: the number of relation symbols can be infinite.
- For us, this study again came up in the context of epistemic logic, to study reasoning in the context of unboundedly many agents (and in a related sense, in games with unboundedly many players).

Undecidability

Since TML contains FO, it is not surprising that it is undecidable.

- In fact even the propositional fragment is undecidable.

Undecidability

Since TML contains FO, it is not surprising that it is undecidable.

- In fact even the propositional fragment is undecidable.
- Padmanabha shows that PTML is as expressive as TML; indeed this holds even for the two-variable fragment.

Translation of TML into FOML

We can translate TML into FOML.

- $\tau(\Diamond_x \alpha) = \Diamond(E(x) \wedge \tau(\alpha))$.

Translation of TML into FOML

We can translate TML into FOML.

- $\tau(\Diamond_x \alpha) = \Diamond(E(x) \wedge \tau(\alpha))$.
- This preserves monodicity and hence many of the earlier results give decidable fragments.

PTML is as hard as TML

PTML is the propositional fragment of TML.

- $\tau(P_i(x_1, \ldots, x_n)) =$
  $\Diamond_{x_1}(\neg q \wedge \Diamond_{x_2}(\ldots \neg q \wedge \Diamond_{x_{n_i}}(\neg q \wedge p_i) \ldots))$.

PTML is as hard as TML

PTML is the propositional fragment of TML.

- $\tau(P_i(x_1, \ldots, x_n)) =$
  $\Diamond_{x_1}(\neg q \wedge \Diamond_{x_2}(\ldots \neg q \wedge \Diamond_{x_{n_i}}(\neg q \wedge p_i) \ldots))$.
- $\tau(\Box_x \phi) = \Box_x(q \implies \tau(\phi))$.
- $\tau(\exists_x \phi) = \exists_x(q \wedge \tau(\phi))$.

PTML is as hard as TML

PTML is the propositional fragment of TML.

- $\tau(P_i(x_1,\ldots,x_n)) =$
  $\Diamond_{x_1}(\neg q \wedge \Diamond_{x_2}(\ldots \neg q \wedge \Diamond_{x_{n_i}}(\neg q \wedge p_i)\ldots))$.
- $\tau(\Box_x\phi) = \Box_x(q \implies \tau(\phi))$.
- $\tau(\exists_x\phi) = \exists_x(q \wedge \tau(\phi))$.
- The translation preserves the number of variables, quantifier rank, and modal depth increases only linearly.

Two variables

Interestingly, the two variable fragment of TML is decidable.

- This again proceeds by constructing a tree model from root to leaf.

Two variables

Interestingly, the two variable fragment of TML is decidable.

- This again proceeds by constructing a tree model from root to leaf.
- It is an induction on modal depth, where at each level, the FO^2 model construction is used.

Two variables

Interestingly, the two variable fragment of TML is decidable.

- This again proceeds by constructing a tree model from root to leaf.
- It is an induction on modal depth, where at each level, the FO^2 model construction is used.
- An analogue of Scott Normal Form is used, and the use of realised types and 'model gluing' is tricky.

The main idea for FO^2

The proof steps involved in showing that the FO^2 fragment has the bounded model property.

- Every sentence $\phi \in FO^2$ has an *equi-satisfiable* sentence in Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).

The main idea for FO^2

The proof steps involved in showing that the FO^2 fragment has the bounded model property.

- Every sentence $\phi \in FO^2$ has an *equi-satisfiable* sentence in Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).
- For a given *FO* structure A and elements c, d in it, the $2 - type(c, d) = (\Gamma_1, \Gamma_2)$ which are the set of *atoms* true in A by mapping the variables x, y to (c, d) and (d, c).

The main idea for FO^2

The proof steps involved in showing that the FO^2 fragment has the bounded model property.

- Every sentence $\phi \in FO^2$ has an *equi-satisfiable* sentence in Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).
- For a given *FO* structure A and elements c, d in it, the $2 - type(c, d) = (\Gamma_1, \Gamma_2)$ which are the set of *atoms* true in A by mapping the variables x, y to (c, d) and (d, c). $1 - type(c)$ is got by mapping x, y to (c, c) .

The main idea for FO^2

The proof steps involved in showing that the FO^2 fragment has the bounded model property.

- Every sentence $\phi \in FO^2$ has an *equi-satisfiable* sentence in Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).
- For a given *FO* structure A and elements c, d in it, the $2 - type(c, d) = (\Gamma_1, \Gamma_2)$ which are the set of *atoms* true in A by mapping the variables x, y to (c, d) and (d, c). $1 - type(c)$ is got by mapping x, y to (c, c) .
- Given ϕ in SNF satisfiable in A, we can build a bounded model based on $1 - type(A)$.

Normal forms

We have normal forms for FO^2 and for modal logic.

- Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).

Normal forms

We have normal forms for FO^2 and for modal logic.

- Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).
- For propositional modal logic we have (a normal form due to Kit Fine): DNF where every clause is of the form $(\bigwedge_i s_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j$ where s_i are literals and α and the β_js are quantifier free.

Normal forms

We have normal forms for FO^2 and for modal logic.

- Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).
- For propositional modal logic we have (a normal form due to Kit Fine): DNF where every clause is of the form $(\bigwedge_i s_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j$ where s_i are literals and α and the β_js are quantifier free.

Normal forms

We have normal forms for FO^2 and for modal logic.

- Scott Normal Form: $\forall x.\forall y.\alpha \wedge \bigwedge_j (\forall x.\exists y.\beta_j)$ where α and the β_j's are quantifier free (by introducing new predicates).
- For propositional modal logic we have (a normal form due to Kit Fine): DNF where every clause is of the form $(\bigwedge_i s_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j$ where s_i are literals and α and the β_js are quantifier free.

We need to combine the two for $PTML^2$.

Fine Scott Normal form for $PTML^2$

Below let z range over $\{x, y\}$.

- For $PTML^2$ we have formulas in DNF where each clause is of the form $\sigma_1 \wedge \sigma_2$ where:
- $\sigma_1 = (\bigwedge_i s_i) \wedge \bigwedge_z (\Box_z \alpha \wedge \bigwedge_j \Diamond_z \beta_j)$
- $\sigma_2 = \bigwedge_z (\forall z.\gamma \wedge \bigwedge_k (\exists z.\delta_k)) \wedge \forall x.\forall y.\phi \wedge \bigwedge_m (\forall x.\exists y.\psi_m)$
- Here α and the β_js are recursively in the normal form, $\gamma, \phi, \delta_k, \psi_\ell$ are all quantifier free and every modal formula occurring in them is recursively in the normal form.

Model construction

Strategy: For a $PTML^2$ formula satisfiable in a tree model, inductively come up with bounded agent models for every subtree of the given tree (based on types), starting from leaves to the root.

- When we add new type based agents to a world at some height, to maintain monotonicity, we need to propagate the newly added agents throughout its descendants.

Model construction

Strategy: For a $PTML^2$ formula satisfiable in a tree model, inductively come up with bounded agent models for every subtree of the given tree (based on types), starting from leaves to the root.

- When we add new type based agents to a world at some height, to maintain monotonicity, we need to propagate the newly added agents throughout its descendants.
- This is achieved by the model extension lemma, which assigns every new agent c to some existing agent $f(c)$ in such a way that c mimics the "type" of $f(c)$.

Model construction

Strategy: For a $PTML^2$ formula satisfiable in a tree model, inductively come up with bounded agent models for every subtree of the given tree (based on types), starting from leaves to the root.

- When we add new type based agents to a world at some height, to maintain monotonicity, we need to propagate the newly added agents throughout its descendants.
- This is achieved by the model extension lemma, which assigns every new agent c to some existing agent $f(c)$ in such a way that c mimics the "type" of $f(c)$. We create one successor subtree of $f(c)$ for each c and add all C to all successor subtrees.

Model construction

Strategy: For a $PTML^2$ formula satisfiable in a tree model, inductively come up with bounded agent models for every subtree of the given tree (based on types), starting from leaves to the root.

- When we add new type based agents to a world at some height, to maintain monotonicity, we need to propagate the newly added agents throughout its descendants.
- This is achieved by the model extension lemma, which assigns every new agent c to some existing agent $f(c)$ in such a way that c mimics the "type" of $f(c)$. We create one successor subtree of $f(c)$ for each c and add all C to all successor subtrees.
- The central idea is that this transformation preserves $PTML^2$ formula satisfiability.

Decision procedure

The model construction outlined proves a bounded agent property.

- We show that ϕ is satisfiable iff it is satisfiable in a model whose domain is of size $\leq 2^{2^{|\phi|}}$.
- So we get a $2 - \mathit{ExpSpace}$ algorithm for satisfiability.

Decision procedure

The model construction outlined proves a bounded agent property.

- We show that ϕ is satisfiable iff it is satisfiable in a model whose domain is of size $\leq 2^{2^{|\phi|}}$.
- So we get a $2 - \mathit{ExpSpace}$ algorithm for satisfiability.
- There is a $\mathit{NExpTime}$ lower bound for FO^2.

Bundled fragments

We spoke of the $\exists x \Box$ and $\forall x \Box$ bundles.

- Over increasing domain, we can have both and the problem is *PSpace* complete.

Bundled fragments

We spoke of the $\exists x \Box$ and $\forall x \Box$ bundles.

- Over increasing domain, we can have both and the problem is *PSpace* complete.
- Over constant domain models, the $\forall x \Box$ fragment is undecidable, but the $\exists x \Box$ fragment is *PSpace* complete.

Bundled fragments

We spoke of the $\exists x\Box$ and $\forall x\Box$ bundles.

- Over increasing domain, we can have both and the problem is *PSpace* complete.
- Over constant domain models, the $\forall x\Box$ fragment is undecidable, but the $\exists x\Box$ fragment is *PSpace* complete.
- There are clearly many more combinations and we need to map the terrain.

Bundled fragments

We spoke of the $\exists x \Box$ and $\forall x \Box$ bundles.

- Over increasing domain, we can have both and the problem is *PSpace* complete.
- Over constant domain models, the $\forall x \Box$ fragment is undecidable, but the $\exists x \Box$ fragment is *PSpace* complete.
- There are clearly many more combinations and we need to map the terrain.

Implicit quantification

A variable-free modal logic.

- $[\forall]\alpha$ asserts α for every x-successor for every x.
- $[\exists]\alpha$ asserts α for every x-successor for some x.
- IQML is exactly the propositional bundled fragment of TML.

Strengthening the decidability results

TML^2 is decidable. Are there decidable extensions?

- Addition of a single constant to the vocabulary makes it undecidable.

Strengthening the decidability results

TML^2 is decidable. Are there decidable extensions?

- Addition of a single constant to the vocabulary makes it undecidable.
- Adding equality, the logic lacks the finite agent property.

Strengthening the decidability results

TML^2 is decidable. Are there decidable extensions?

- Addition of a single constant to the vocabulary makes it undecidable.
- Adding equality, the logic lacks the finite agent property.
- Significant gap between lower bounds and upper bounds.

Strengthening the decidability results

TML^2 is decidable. Are there decidable extensions?

- Addition of a single constant to the vocabulary makes it undecidable.
- Adding equality, the logic lacks the finite agent property.
- Significant gap between lower bounds and upper bounds.
- We have decidability for systems with infinite sets of agents, where they form a regular set.

The door is open

More decidable fragments are known by now.

- The guarded fragment of TML has been shown to be decidable by Orlandelli and Corsi. Shtakser has extended this to set quantification.

The door is open

More decidable fragments are known by now.

- The guarded fragment of TML has been shown to be decidable by Orlandelli and Corsi. Shtakser has extended this to set quantification.
- Many questions remain: equality is intriguing.

The door is open

More decidable fragments are known by now.

- The guarded fragment of TML has been shown to be decidable by Orlandelli and Corsi. Shtakser has extended this to set quantification.
- Many questions remain: equality is intriguing.
- Model classes, correspondence theory: mostly open.

The door is open

More decidable fragments are known by now.

- The guarded fragment of TML has been shown to be decidable by Orlandelli and Corsi. Shtakser has extended this to set quantification.
- Many questions remain: equality is intriguing.
- Model classes, correspondence theory: mostly open.
- Expressiveness of different logics needs to be carefully pinned down.

References

Important papers:

- Melvin Fitting, Lars Thalmann, and Andrei Voronkov. Term-modal logics. Studia Logica, 69(1):133-169, 2001.
- Mikhail Rybakov and Dmitry Shkatov. Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter. Studia Logica, pages 1-23, 2017.
- Yanjing Wang. A New Modal Framework for Epistemic Logic. TARK 2017: 515-534.
- Frank Wolter and Michael Zakharyaschev. Decidable fragments of first-order modal logics. The Journal of Symbolic Logic, 66(3):1415- 1438, 2001.

Our papers

- Anantha Padmanabha, R. Ramanujam, and Yanjing Wang. Bundled fragments of first-order modal logic: (un)decidability. FSTTCS 2018, December 11-13, 2018, Ahmedabad, India, pages 43:1-43:20, 2018.
- Anantha Padmanabha and R. Ramanujam. The Monodic Fragment of Propositional Term Modal Logic. Studia Logica, Volume 107(3), Pages 533 - 557, 2019.
- Anantha Padmanabha and R. Ramanujam. Two variable fragment of Term Modal Logic. 44th International Symposium on Mathematical Foundations of Computer Science (MFCS), 2019.
- Anantha Padmanabha and R. Ramanujam. Propositional Modal Logic with Implicit Modal Quantification. Logic and Its Applications - 8th Indian Conference, (ICLA), Lecture Notes in Computer Science, Volume 11600, 2019.