Satisfiability problem for Term Modal Logic

Anantha Padmanabha

joint work with R. Ramanujam

Institute of Mathematical Sciences, HBNI, Chennai
19 July 2018

Formal Methods Update Meeting 2018
BITS, Goa Campus

1. Introduction
2. Term Modal logic
3. Undecidability results
4. Decidable fragments

Modal logics

- Modal logics are extensively studied and applied in various fields like verification, epistemic logic etc.

Modal logics

- Modal logics are extensively studied and applied in various fields like verification, epistemic logic etc.
- We have a fixed set of modalities $Ag = \{1 \cdots n\}$ and $\Box_i$ and $\Diamond_i$ modalities for every $i \in Ag$.

Modal logics

- Modal logics are extensively studied and applied in various fields like verification, epistemic logic etc.
- We have a fixed set of modalities $Ag = \{1 \cdots n\}$ and $\Box_i$ and $\Diamond_i$ modalities for every $i \in Ag$.
- Modalities are also referred as agents.

Modal logics

- Modal logics are extensively studied and applied in various fields like verification, epistemic logic etc.
- We have a fixed set of modalities $Ag = \{1 \cdots n\}$ and $\Box_i$ and $\Diamond_i$ modalities for every $i \in Ag$.
- Modalities are also referred as agents.

Syntax:

$Ag = \{1 \cdots n\}$ is a non-empty fixed set of finite agents. P is a countable set of propositions.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_i \varphi$$

where $p \in P$ and $i \in Ag$.

Modal logic

Semantics:

$M = (W, R_1 \cdots R_n, V)$ is a structure where

- W is a non-empty set of worlds
- $R_i \subseteq W \times W$
- $V : W \to 2^P$.

For any $w \in W$ and a formula φ, $M, w \models \varphi$ is defined inductively as follows:

$M, w \models p$	iff	$p \in V(w)$
$M, w \models \neg\varphi_1$	iff	$M, w \not\models \varphi_1$
$M, w \models \varphi_1 \wedge \varphi_2$	iff	$M, w \models \varphi_1$ and $M, w \models \varphi_2$
$M, w \models \Box_i \varphi_1$	iff	for every $w' \in W$ if $(w, w') \in R_i$ then $M, w' \models \varphi_1$.

Fixed agent set

- In the classical modal logic case (and first order model logic), the collection of agents is fixed.

Fixed agent set

- In the classical modal logic case (and first order model logic), the collection of agents is fixed.
- Every agent has a unique name and these names are common knowledge among the agents.

Fixed agent set

- In the classical modal logic case (and first order model logic), the collection of agents is fixed.
- Every agent has a unique name and these names are common knowledge among the agents.
- Are these assumptions reasonable?

Fixed agent set

- In situations like client-server systems, the clients do not know how many other clients are present.

Fixed agent set

- In situations like client-server systems, the clients do not know how many other clients are present.
- The number of processes cannot be bounded a priori in a system where processes can fork new processes.

Fixed agent set

- In situations like client-server systems, the clients do not know how many other clients are present.
- The number of processes cannot be bounded a priori in a system where processes can fork new processes.
- In epistemic settings, can we have a logic where agency is in the scope of knowledge?

Fixed agent set

- In situations like client-server systems, the clients do not know how many other clients are present.
- The number of processes cannot be bounded a priori in a system where processes can fork new processes.
- In epistemic settings, can we have a logic where agency is in the scope of knowledge?
  - Everyone who knows p, knows that someone knows q.

TML

- Term Modal Logic (TML) was introduced by Fitting, Thalmann and Voronkov[2001].

TML

- Term Modal Logic (TML) was introduced by Fitting, Thalmann and Voronkov[2001].
- In TML, modalities are indexed by terms and these terms can be quantified.
  $\exists x(\Box_x \forall y \Diamond_y (P(x, y)))$

TML syntax

Given Var (variables) and **P** (predicates), the syntax of TML is defined as follows:

$$\varphi ::= P\overline{x} \mid x \approx y \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \exists x\varphi \mid \Box_x\varphi$$

where $x \in$ Var, $P \in \mathbf{P}$.

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds
- D is a non-empty set of agents

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds
- D is a non-empty set of agents
- $R \subseteq (W \times D \times W)$

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds
- D is a non-empty set of agents
- $R \subseteq (W \times D \times W)$
- $\delta : W \to 2^D$ assigns to each $w \in W$ a *non-empty* set of "live agents"

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds
- D is a non-empty set of agents
- $R \subseteq (W \times D \times W)$
- $\delta : W \to 2^D$ assigns to each $w \in W$ a *non-empty* set of "live agents" s.t. $(w, d, v) \in R$ implies $d \in \delta(w) \subseteq \delta(v)$ for any $w, v \in W$

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds
- D is a non-empty set of agents
- $R \subseteq (W \times D \times W)$
- $\delta : W \to 2^D$ assigns to each $w \in W$ a *non-empty* set of "live agents" s.t. $(w, d, v) \in R$ implies $d \in \delta(w) \subseteq \delta(v)$ for any $w, v \in W$
- $\rho : (W \times \mathbf{P}) \to \bigcup_{n \in \omega} 2^{D^n}$ such that ρ assigns to each n-ary predicate on each world an n-ary relation on D.

TML

Semantics

An (increasing agent) model M for TML is a tuple (W, D, δ, R, ρ) where,

- W is a non-empty set of worlds
- D is a non-empty set of agents
- $R \subseteq (W \times D \times W)$
- $\delta : W \to 2^D$ assigns to each $w \in W$ a *non-empty* set of "live agents" s.t. $(w, d, v) \in R$ implies $d \in \delta(w) \subseteq \delta(v)$ for any $w, v \in W$
- $\rho : (W \times \mathbf{P}) \to \bigcup_{n \in \omega} 2^{D^n}$ such that ρ assigns to each n-ary predicate on each world an n-ary relation on D.

We need interpretation for variables $\sigma : \mathrm{Var} \to D$.

TML

Semantics

Given $M = (W, D, \delta, R, \rho)$, $w \in W$, and an assignment σ that is relevant at w, define $M, w, \sigma \vDash \varphi$ inductively as follows:

$$
\begin{array}{lcl}
M, w, \sigma \vDash P(x_1 \cdots x_n) & \Leftrightarrow & (\sigma(x_1), \cdots, \sigma(x_n)) \in \rho(P, w) \\
M, w, \sigma \vDash x \approx y & \Leftrightarrow & \sigma(x) = \sigma(y) \\
M, w, \sigma \vDash \neg\varphi & \Leftrightarrow & M, w, \sigma \nvDash \varphi \\
M, w, \sigma \vDash (\varphi \wedge \psi) & \Leftrightarrow & M, w, \sigma \vDash \varphi \text{ and } M, w, \sigma \vDash \psi \\
M, w, \sigma \vDash \exists x \varphi & \Leftrightarrow & \text{there is some } d \in \delta(w) \text{ such} \\
 & & M, w, \sigma[x \mapsto d] \vDash \varphi \\
M, w, \sigma \vDash \Box_x \varphi & \Leftrightarrow & M, v, \sigma \vDash \varphi \text{ for all } v \text{ s.t.} \\
 & & (w, \sigma(x), v) \in R
\end{array}
$$

Term-modal logic

Examples

- Everyone who knows p, knows that someone knows q
  $\forall x.\Box_x(p \Rightarrow \exists y \Box_y q)$.

Term-modal logic

Examples

- Everyone who knows p, knows that someone knows q
  $\forall x.\Box_x(p \Rightarrow \exists y \Box_y q)$.
- For every process, there exists another process such that there is one execution the first process after which any possible execution of the second process, property p holds.
  $\forall x \exists y.\Diamond_x \Box_y p$.

Satistiability problem

- Satisfiability problem asks Given a formula $\varphi \in$ TML is there a model M and a σ such that $M, w, \sigma \models \varphi$.

Satistiability problem

- Satisfiability problem asks Given a formula $\varphi \in$ TML is there a model M and a σ such that $M, w, \sigma \models \varphi$.
- Since FO corresponds to the modal free fragment of TML, satisfiability problem is undecidable.

Satistiability problem

- Satisfiability problem asks Given a formula $\varphi \in$ TML is there a model M and a σ such that $M, w, \sigma \models \varphi$.
- Since FO corresponds to the modal free fragment of TML, satisfiability problem is undecidable.

Theorem (PR 2017)

Satisfiability problem for TML *is undecidable even if atoms are restricted to propositions.*

Satistiability problem

- Satisfiability problem asks Given a formula $\varphi \in$ TML is there a model M and a σ such that $M, w, \sigma \models \varphi$.
- Since FO corresponds to the modal free fragment of TML, satisfiability problem is undecidable.

Theorem (PR 2017)

Satisfiability problem for TML *is undecidable even if atoms are restricted to propositions.*

- Can be strengthen the undecidability result?
- Are there any interesting decidable fragments?

TML over $(\top, \bot)$

Theorem (PR)

Satisfiability problem for TML *when atoms are restricted to* $(\top, \bot)$ *is undecidable.*

TML over $(\top, \bot)$

Theorem (PR)

Satisfiability problem for TML *when atoms are restricted to* $(\top, \bot)$ *is undecidable.*

Proof sketch.

Reduction from FO(R) satisfiatibility problem.

TML over $(\top, \bot)$

Theorem (PR)

Satisfiability problem for TML *when atoms are restricted to* $(\top, \bot)$ *is undecidable.*

Proof sketch.

Reduction from FO(R) satisfiatibility problem.

- $\alpha_{R(x,y)} \rightsquigarrow \Diamond_x \Diamond_y \top$
- $\alpha_{\neg\varphi} \rightsquigarrow \neg\alpha_\varphi$
- $\alpha_{\varphi\wedge\psi} \rightsquigarrow \alpha_\varphi \wedge \alpha_\psi$
- $\alpha_{\exists x\ \varphi} \rightsquigarrow \exists x\ \alpha_\varphi$

TML over $(\top, \bot)$

Theorem (PR)

Satisfiability problem for TML *when atoms are restricted to* $(\top, \bot)$ *is undecidable.*

Proof sketch.

Reduction from FO(R) satisfiatibility problem.

- $\alpha_{R(x,y)} \rightsquigarrow \Diamond_x \Diamond_y \top$
- $\alpha_{\neg\varphi} \rightsquigarrow \neg\alpha_\varphi$
- $\alpha_{\varphi\wedge\psi} \rightsquigarrow \alpha_\varphi \wedge \alpha_\psi$
- $\alpha_{\exists x\ \varphi} \rightsquigarrow \exists x\ \alpha_\varphi$

Any $\varphi \in$ FO(R) is satisfiable iff α_φ is satisfiable.

TML over $(\top, \bot)$

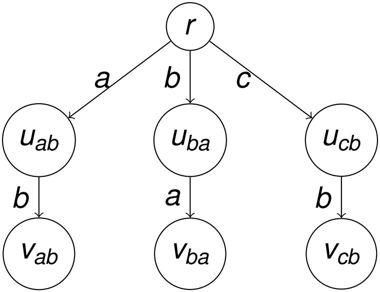

Figure: Model corresponding to the FO structure (D, I) where $D = \{a, b, c\}$ and $I = \{(a, b),\ (b, a),\ (c, b)\}$.

Relationship between modal depth and arity of predicates

Mod. depth	Predicates	Status	Remark
0	$\mathbf{P}^0, \mathbf{P}^1$	D	Follows from FO
0	R	UD	Same as FO(R)
1	$\mathbf{P}^0$	D	Fragment of Monodic TML
1	P	UD	Encode $R(x, y)$ as $\Diamond_x P(y)$
≥ 2	$(\top, \bot)$	UD	Encode $R(x, y)$ as $\Diamond_x \Diamond_y \top$

TML over $\approx$

Theorem (PR)

For TML *over* $\approx$, *the* FinSat, UnSat *and* InfAx *are mutually recursively inseparable.*

TML over $\approx$

Theorem (PR)

For TML *over* $\approx$, *the* FinSat, UnSat *and* InfAx *are mutually recursively inseparable.*

Proof sketch

- Reduction from tiling problem.

TML over $\approx$

Theorem (PR)

For TML *over* $\approx$, *the* FinSat, UnSat *and* InfAx *are mutually recursively inseparable.*

Proof sketch

- Reduction from tiling problem.
- For any tiling instance T, we come up with a formula φ_T such that
  - $\varphi_T \in$ FinSat iff T has some periodic tiling
  - $\varphi_T \in$ *UnSat* iff T has no tiling.
  - $\varphi_T \in$ InfAx iff T has only aperiodic tiling

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.
- A tiling instance is given by $T = (X, t_0)$ where X is a finite set of tile types and $t_0 \in X$
- Given T, $S : \mathbb{N} \times \mathbb{N} \to X$ is a tiling if

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.
- A tiling instance is given by $T = (X, t_0)$ where X is a finite set of tile types and $t_0 \in X$
- Given T, $S : \mathbb{N} \times \mathbb{N} \to X$ is a tiling if $S(0,0) = t_0$

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.
- A tiling instance is given by $T = (X, t_0)$ where X is a finite set of tile types and $t_0 \in X$
- Given T, $S : \mathbb{N} \times \mathbb{N} \to X$ is a tiling if $S(0,0) = t_0$ and for all $i, j \in \mathbb{N}$, if $S(i,j) = t$ and $S(i+1, j) = t'$ then $r_t = \ell_{t'}$

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.
- A tiling instance is given by $T = (X, t_0)$ where X is a finite set of tile types and $t_0 \in X$
- Given T, $S : \mathbb{N} \times \mathbb{N} \to X$ is a tiling if $S(0,0) = t_0$ and for all $i, j \in \mathbb{N}$, if $S(i,j) = t$ and $S(i+1, j) = t'$ then $r_t = \ell_{t'}$ and similarly, if $S(i,j) = t$ and $S(i, j+1) = t'$ then $u_t = d_{t'}$.

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.
- A tiling instance is given by $T = (X, t_0)$ where X is a finite set of tile types and $t_0 \in X$
- Given T, $S : \mathbb{N} \times \mathbb{N} \to X$ is a tiling if $S(0,0) = t_0$ and for all $i, j \in \mathbb{N}$, if $S(i,j) = t$ and $S(i+1, j) = t'$ then $r_t = \ell_{t'}$ and similarly, if $S(i,j) = t$ and $S(i, j+1) = t'$ then $u_t = d_{t'}$.

Before encoding the tiling instance, we need to encode a grid structure which is independent of the tiling instance.

TML over $\approx$

Tiling encoding

- A tile is given by $t = (u_t, d_t, r_t, \ell_t)$ where each component is one of the finite set of colours C.
- A tiling instance is given by $T = (X, t_0)$ where X is a finite set of tile types and $t_0 \in X$
- Given T, $S : \mathbb{N} \times \mathbb{N} \to X$ is a tiling if $S(0,0) = t_0$ and for all $i, j \in \mathbb{N}$, if $S(i,j) = t$ and $S(i+1,j) = t'$ then $r_t = \ell_{t'}$ and similarly, if $S(i,j) = t$ and $S(i,j+1) = t'$ then $u_t = d_{t'}$.

Before encoding the tiling instance, we need to encode a grid structure which is independent of the tiling instance. Before grid we need to enforce $\mathbb{N}$.

TML over $\approx$

Idea: Encode $x < y$ as $\Diamond_x \Diamond_y \top$.

TML over $\approx$

Idea: Encode $x < y$ as $\Diamond_x \Diamond_y \top$.

φ_0	$:=$	$\exists x\ zero(x)$	there is a min. element.
φ_{ir}	$:=$	$\forall x \neg \Diamond_x \Diamond_x \top$	$c \not< c$ (irreflexive)
φ_{tot}	$:=$	$\forall x \forall y (x \not\approx y \Rightarrow \Diamond_x \Diamond_y \top \vee \Diamond_y \Diamond_x \top)$	for all $c \neq d$ either $c < d$ or $d < c$ (total)
φ_{dis}	$:=$	$\forall x \big(last(x) \vee \exists y\ succ(x, y) \big)$	for all c, either c is last or has a successor
φ_{trans}	$:=$	$\forall x \forall y \forall z (\Diamond_x \Diamond_y \top \wedge \Diamond_y \Diamond_z \top) \Rightarrow (\Diamond_x \Diamond_z \top)$	$c < d$ and $d < e$ implies $c < e$.

TML over $\approx$

Idea: Encode $x < y$ as $\Diamond_x \Diamond_y \top$.

φ_0	$:=$	$\exists x\ zero(x)$	there is a min. element.
φ_{ir}	$:=$	$\forall x \neg \Diamond_x \Diamond_x \top$	$c \not< c$ (irreflexive)
φ_{tot}	$:=$	$\forall x \forall y (x \not\approx y \Rightarrow \Diamond_x \Diamond_y \top \vee \Diamond_y \Diamond_x \top)$	for all $c \neq d$ either $c < d$ or $d < c$ (total)
φ_{dis}	$:=$	$\forall x \big(last(x) \vee \exists y\ succ(x, y)\big)$	for all c, either c is last or has a successor
φ_{trans}	$:=$	$\forall x \forall y \forall z (\Diamond_x \Diamond_y \top \wedge \Diamond_y \Diamond_z \top) \Rightarrow (\Diamond_x \Diamond_z \top)$	$c < d$ and $d < e$ implies $c < e$.
where,			
$zero(x)$	$:=$	$\forall y\ \neg \Diamond_y \Diamond_x \top$	for all $c, c \not< I(x)$
$last(x)$	$:=$	$\forall y \neg \Diamond_x \Diamond_y \top$	forall c, $I(x) \not< c$
$succ(x, y)$	$:=$	$\big(last(y) \wedge zero(x)\big) \vee \big(\Diamond_x \Diamond_y \top \wedge \forall z (\Diamond_z \Diamond_y \top \Rightarrow (x \approx z \vee \Diamond_z \Diamond_x \top)\big)$	$(I(y) = max\ ;\ I(x) = min)$ or $I(x) < I(y)$ and for all c if $c < I(y)$ then $x = c$ or $c < I(x)$.

TML with $\approx$

Define $\text{Ord} = \{\varphi_0, \varphi_{ir}, \varphi_{tot}, \varphi_{dis}\}$ and $\hat{\varphi} = \bigwedge_{\varphi \in \text{Ord}} \varphi$.

TML with $\approx$

Define $\text{Ord} = \{\varphi_0, \varphi_{ir}, \varphi_{tot}, \varphi_{dis}\}$ and $\hat{\varphi} = \bigwedge_{\varphi \in \text{Ord}} \varphi$.

Lemma

The following statements hold for the formula $\hat{\varphi}$:

1. *For any $\mathbb{N}' \subseteq \mathbb{N}$ (either finite or infinite) which is an initial fragment of $\mathbb{N}$, there is some $M = (W, \mathbb{N}', \delta, R)$ and $w \in W$ where $\mathbb{N}' = \gamma(w)$ such that $M, w \models \hat{\varphi}$.*

TML with $\approx$

Define Ord $= \{\varphi_0, \varphi_{ir}, \varphi_{tot}, \varphi_{dis}\}$ and $\hat{\varphi} = \bigwedge_{\varphi \in \text{Ord}} \varphi$.

Lemma

The following statements hold for the formula $\hat{\varphi}$:

1. *For any $\mathbb{N}' \subseteq \mathbb{N}$ (either finite or infinite) which is an initial fragment of $\mathbb{N}$, there is some $M = (W, \mathbb{N}', \delta, R)$ and $w \in W$ where $\mathbb{N}' = \gamma(w)$ such that $M, w \models \hat{\varphi}$.*
2. *For any model $M = (W, D, \delta, R)$ if $M, w \models \hat{\varphi}$ then there some initial fragment of $\mathbb{N}$(say $\mathbb{N}'$) and a function $f : \mathbb{N}' \to \delta(w)$ where for all $i, j \in \mathbb{N}'$, we have $i < j$ iff $M, w \models \Diamond_{f(i)} \Diamond_{f(j)} \top$.*

TML with $\approx$

Define Ord $= \{\varphi_0, \varphi_{ir}, \varphi_{tot}, \varphi_{dis}\}$ and $\hat{\varphi} = \bigwedge_{\varphi \in \text{Ord}} \varphi$.

Lemma

The following statements hold for the formula $\hat{\varphi}$:

1. *For any $\mathbb{N}' \subseteq \mathbb{N}$ (either finite or infinite) which is an initial fragment of $\mathbb{N}$, there is some $M = (W, \mathbb{N}', \delta, R)$ and $w \in W$ where $\mathbb{N}' = \gamma(w)$ such that $M, w \models \hat{\varphi}$.*
2. *For any model $M = (W, D, \delta, R)$ if $M, w \models \hat{\varphi}$ then there some initial fragment of $\mathbb{N}$(say $\mathbb{N}'$) and a function $f : \mathbb{N}' \to \delta(w)$ where for all $i, j \in \mathbb{N}'$, we have $i < j$ iff $M, w \models \Diamond_{f(i)} \Diamond_{f(j)} \top$.*

Hence *w.l.o.g* for any $M, w \models \hat{\varphi}$ we can assume that there is some initial fragment $\mathbb{N}'$ of $\mathbb{N}$ such that $\delta(w) = \mathbb{N}'$ and for all $i, j \in \mathbb{N}',\ i < j$ iff $M, w \models \Diamond_i \Diamond_j \top$.

TML with $\approx$

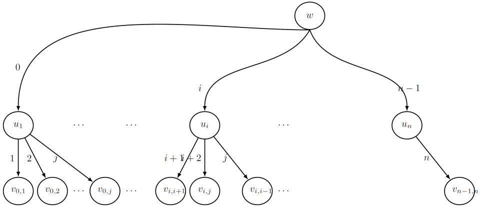

Figure: A model for Ord when $\mathbb{N}' = [1 \cdots n]$ is finite.

TML over $\approx$

- Encode every tile t_i as a path of length i, given by

$$p_i ::= \bigwedge_{j<i} (\forall z \Box)^j (\exists z \Diamond \top) \wedge (\forall z \Box)^i (\forall z \Box_z \bot)$$

φ_{tile}	$:=$	$\forall z_1 \forall z_2 \forall x \forall y \, \Box_{z_1} \Box_{z_2} ((\Diamond_x \Diamond_y \top) \wedge (\Box_x \Box_y \bigvee_{t_i \in X} p_i))$
φ_{init}	$:=$	$\forall x \; zero(x) \Rightarrow \forall z_1 \forall z_2 (\Box_{z_1} \Box_{z_2} \Diamond_x \Diamond_x p_0)$
φ_{hor}	$:=$	$\forall x \forall y \forall z \; succ(x,y) \Rightarrow$ $(\forall z_1 \forall z_2 \Box_{z_1} \Box_{z_2} (\bigvee_{r_{t_i} = \ell_{t_j}} (\Box_x \Box_z (p_i) \wedge \Box_y \Box_z (p_j)))$
φ_{ver}	$:=$	$\forall x \forall y \forall z \; succ(x,y) \Rightarrow$ $\forall z_1 \forall z_2 (\Box_{z_1} \Box_{z_2} (\bigvee_{u_{t_i} = d_{t_j}} (\Box_z \Box_x (p_i) \wedge \Box_z \Box_y (p_j)))$

TML with $\approx$

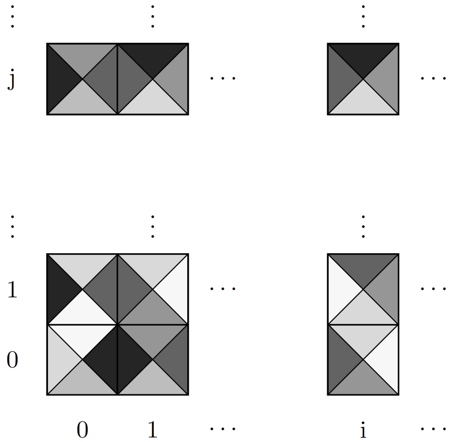

Figure: Tiling instance.

TML with $\approx$

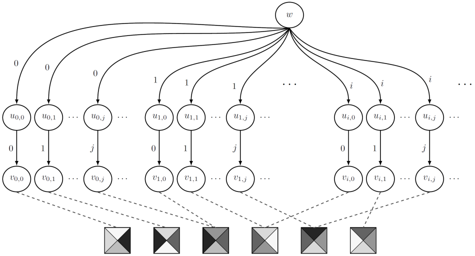

Figure: A model for the tiling problem.

Decidable fragments

Literature

- Orlandelli and Crosi consider two cases of decidable fragments:
    - Atoms are propositions and quantifier occurrence is restricted to the form: $\exists x \Box_x \alpha$ (and $\forall x \Diamond_x \alpha$ dually)
    - Atoms are propositions and quantifiers appear in a guarded form: $\forall x (P(x) \Rightarrow \Box_x \alpha)$ and $\exists x (P(x) \wedge \Box_x \alpha)$ (and their duals).

Decidable fragments

Literature

- Orlandelli and Crosi consider two cases of decidable fragments:
  - Atoms are propositions and quantifier occurrence is restricted to the form: $\exists x \Box_x \alpha$ (and $\forall x \Diamond_x \alpha$ dually)
  - Atoms are propositions and quantifiers appear in a guarded form: $\forall x(P(x) \Rightarrow \Box_x \alpha)$ and $\exists x(P(x) \wedge \Box_x \alpha)$ (and their duals).
- Shtakser considers a more general guarded fragment (propositional atoms) of the form $\forall X(P(X) \Rightarrow \Box_X \alpha)$ and $\exists X(P(X) \wedge \Box_X \alpha)$ where X is quantified over subsets of agents and P is interpreted appropriately.

Decidable fragments

Literature

- Orlandelli and Crosi consider two cases of decidable fragments:
  - Atoms are propositions and quantifier occurrence is restricted to the form: $\exists x \Box_x \alpha$ (and $\forall x \Diamond_x \alpha$ dually)
  - Atoms are propositions and quantifiers appear in a guarded form: $\forall x(P(x) \Rightarrow \Box_x \alpha)$ and $\exists x(P(x) \wedge \Box_x \alpha)$ (and their duals).
- Shtakser considers a more general guarded fragment (propositional atoms) of the form $\forall X(P(X) \Rightarrow \Box_X \alpha)$ and $\exists X(P(X) \wedge \Box_X \alpha)$ where X is quantified over subsets of agents and P is interpreted appropriately.

Semantically motivated fragments, from their interest in the epistemic logic. (ex: All eye-witnesses know who killed Mary)

Monodic fragment

Monodic fragment is where there is only one free variable inside the scope of modality.

Monodic fragment

Monodic fragment is where there is only one free variable inside the scope of modality. φ is in monodic fragment if for any subformula $\Box_x \alpha$ we have $\mathsf{Fv}(\alpha) \subseteq \{x\}$.

Monodic fragment

Monodic fragment is where there is only one free variable inside the scope of modality. φ is in monodic fragment if for any subformula $\Box_x \alpha$ we have $\mathsf{Fv}(\alpha) \subseteq \{x\}$.

Theorem (PR 2017)

Satisfiability problem for monodic fragment is decidable.

Monodic fragment

Monodic fragment is where there is only one free variable inside the scope of modality. φ is in monodic fragment if for any subformula $\Box_x \alpha$ we have $\mathrm{Fv}(\alpha) \subseteq \{x\}$.

Theorem (PR 2017)

Satisfiability problem for monodic fragment is decidable.

Proof sketch.

- Given a formula φ, define $type(w, d) = \{\Delta_x \alpha(x) \mid M, w \models \Delta_a \alpha(a)\}$ where $\Delta \in \{\Box, \Diamond\}$ for every $\Delta_x \alpha(x) \in SF(\varphi)$.

Monodic fragment

Monodic fragment is where there is only one free variable inside the scope of modality. φ is in monodic fragment if for any subformula $\Box_x \alpha$ we have $\mathrm{Fv}(\alpha) \subseteq \{x\}$.

Theorem (PR 2017)

Satisfiability problem for monodic fragment is decidable.

Proof sketch.

- Given a formula φ, define $type(w, d) = \{\Delta_x \alpha(x) \mid M, w \models \Delta_a \alpha(a)\}$ where $\Delta \in \{\Box, \Diamond\}$ for every $\Delta_x \alpha(x) \in SF(\varphi)$.
- The types are finite (exponentially many). Now define equivalence on worlds based which have same set of types. (double exponential).

Monodic fragment

Monodic fragment is where there is only one free variable inside the scope of modality. φ is in monodic fragment if for any subformula $\Box_x \alpha$ we have $\text{Fv}(\alpha) \subseteq \{x\}$.

Theorem (PR 2017)

Satisfiability problem for monodic fragment is decidable.

Proof sketch.

- Given a formula φ, define $type(w, d) = \{\Delta_x \alpha(x) \mid M, w \models \Delta_a \alpha(a)\}$ where $\Delta \in \{\Box, \Diamond\}$ for every $\Delta_x \alpha(x) \in SF(\varphi)$.
- The types are finite (exponentially many). Now define equivalence on worlds based which have same set of types. (double exponential).
- This gives a non deterministic double exponential time algorithm.

TML on 2-variables

- Note that undecidability results need 3 variables. What happens to 2 variable TML?

TML on 2-variables

- Note that undecidability results need 3 variables. What happens to 2 variable TML?
- Note that 2 variable fragment of FO(FO^2) is decidable.

TML on 2-variables

- Note that undecidability results need 3 variables. What happens to 2 variable TML?
- Note that 2 variable fragment of FO(FO^2) is decidable.
- However, Gradel and Otto showed that most of the natural extensions of FO^2(like 2 var lfp, *t*ransitive closure etc) except for the counting quantifiers are undecidable.

TML on 2-variables

- Note that undecidability results need 3 variables. What happens to 2 variable TML?
- Note that 2 variable fragment of FO(FO^2) is decidable.
- However, Gradel and Otto showed that most of the natural extensions of FO^2(like 2 var lfp, *t*ransitive closure etc) except for the counting quantifiers are undecidable.
- On the other hand Kontchakov et.al prove that first order modal logic over 2 variables in undecidable.

TML on 2-variables

- Note that undecidability results need 3 variables. What happens to 2 variable TML?
- Note that 2 variable fragment of FO(FO^2) is decidable.
- However, Gradel and Otto showed that most of the natural extensions of FO^2(like 2 var lfp, transitive closure etc) except for the counting quantifiers are undecidable.
- On the other hand Kontchakov et.al prove that first order modal logic over 2 variables in undecidable.
  In the proof, they use the formula of the form $P(x, y) \equiv \Box P(x, y)$. Now this is not expressible in TML^2 since $\Box$ has to be indexed either by x or y. We use this property of TML crucially to get the decidability.

Towards TML^2 decidability

- Kit Fine introduces a normal form for modal logic which is a DNF where each clause is of the form: $(\bigwedge_i (l_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j)$ where α, β_j are recursively in the normal form.

Towards TML^2 decidability

- Kit Fine introduces a normal form for modal logic which is a DNF where each clause is of the form: $(\bigwedge_i (l_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j)$ where α, β_j are recursively in the normal form.
- Scott has a formal form for FO^2 which is given by: $\forall x \forall y \varphi \wedge \bigwedge_i \forall x \exists y \beta_i$

Towards TML^2 decidability

- Kit Fine introduces a normal form for modal logic which is a DNF where each clause is of the form: $(\bigwedge_i (l_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j)$ where α, β_j are recursively in the normal form.
- Scott has a formal form for FO^2 which is given by: $\forall x \forall y \varphi \wedge \bigwedge_i \forall x \exists y \beta_i$
- We can combine there two to get a normal form for TML^2 which is a DNF where each clause is of the form: $\bigwedge_{i \leq a} s_i \wedge \bigwedge_{z \in \{x,y\}} (\Box_z \alpha \wedge \bigwedge_{j \leq m_z} \Diamond_z \beta_j) \wedge \bigwedge_{z \in \{x,y\}} (\forall z \gamma \wedge \bigwedge_{k \leq n_z} \exists z\ \delta_k) \wedge \forall x \forall y\ \varphi \wedge \bigwedge_{l \leq b} \forall x \exists y \psi_l$ where s_i are literals and all α, β_j are recursively in the normal form and $\gamma, \delta_k, \varphi, \psi_l$ do not have quantifiers at the outermost level.

Lemma

Every TML^2 *formula has an equi-satisfiable formula in the normal form.*

Lemma

Every TML^2 *formula has an equi-satisfiable formula in the normal form.*

Theorem

Every formula in TML^2 *normal form has small model property.*

Lemma

Every TML^2 *formula has an equi-satisfiable formula in the normal form.*

Theorem

Every formula in TML^2 *normal form has small model property.*

Inductive kings and courts argument. We believe that we have the proof, but going through it to check for bugs.

Conclusion

Summary

- We looked at some tight undecidability results for TML.
- On the positive side, we have some decidable fragments like monodic, 2-variable fragment.

Conclusion

Summary

- We looked at some tight undecidability results for TML.
- On the positive side, we have some decidable fragments like monodic, 2-variable fragment.

Future Work

- Looking at different class of frames.

Conclusion

Summary

- We looked at some tight undecidability results for TML.
- On the positive side, we have some decidable fragments like monodic, 2-variable fragment.

Future Work

- Looking at different class of frames.
- Fragments based on other parameters like quantifier alternation etc.

Conclusion

Summary

- We looked at some tight undecidability results for TML.
- On the positive side, we have some decidable fragments like monodic, 2-variable fragment.

Future Work

- Looking at different class of frames.
- Fragments based on other parameters like quantifier alternation etc.
- Model checking and verification.