Propositional Term Modal Logic

By

Anantha Padmanabha MS

MATH10201305001

The Institute of Mathematical Sciences, Chennai

A thesis submitted to the

Board of Studies in Mathematical Sciences

In partial fulfillment of requirements

for the Degree of

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE

December, 2019

Homi Bhabha National Institute

Recommendations of the Viva Voce Committee

As members of the Viva Voce Committee, we certify that we have read the dissertation prepared by Anantha Padmanabha MS entitled "Propositional Term Modal Logic" and recommend that it may be accepted as fulfilling the thesis requirement for the award of Degree of Doctor of Philosophy.

Kamal Lodaya — Date: 6/12/19
Chairman - Kamal Lodaya

R Raman — Date: 6 Dec 2019
Guide/Convenor - R Ramanujam

[signature] — Date: 6 Dec 2019
Examiner - Sreejith AV

Suresh — Date: December 6, 2019
Member 1 - SP Suresh

Sujata Ghosh — Date: 6.12.2019
Member 2 - Sujata Ghosh

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to HBNI.

I hereby certify that I have read this thesis prepared under my direction and recommend that it may be accepted as fulfilling the thesis requirement.

Date: 6 Dec 2019

Place: Chennai

R Raman

R Ramanujam (Guide)

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the Library to be made available to borrowers under rules of the HBNI.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the Competent Authority of HBNI when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Anantha Padmanabha MS

DECLARATION

I hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree / diploma at this or any other Institution / University.

Anantha Padmanabha MS

LIST OF PUBLICATIONS ARISING FROM THE THESIS

Journal

1. The Monodic Fragment of Propositional Term Modal Logic;
Anantha Padmanabha and R. Ramanujam;
Studia Logica, 2019, Volume 107(3), Pages 533 – 557.

Conferences

1. Two variable fragment of Term Modal Logic;
Anantha Padmanabha and R. Ramanujam; in 44th International Symposium on Mathematical Foundations of Computer Science (MFCS), Leibniz International Proceedings in Informatics, 2019, Volume 138, Pages 30 : 1 – 30 : 14.

2. Propositional Modal Logic with Implicit Modal Quantification;
Anantha Padmanabha and R. Ramanujam; in 8th Indian Conference on Logic and Its Applications (ICLA), Lecture Notes in Computer Science, 2019, Volume 11600, Pages 6 - 17.

3. Model Checking a Logic over Systems with Regular Sets of Processes;
Anantha Padmanabha and R. Ramanujam; in Developmental Aspects of Intelligent Adaptive Systems (Innovations in Software Engineering), 2017, CEUR Workshop Proceedings, Volume 1819, Pages 5 : 1 – 5 : 5.

Anantha Padmanabha MS

Dedicated to my teachers. . .

ACKNOWLEDGEMENTS

I thank my PhD supervisor Ramanujam for his infinite patience. His ideas and insights during discussions have helped me better understand the subject and the problem at hand. He was always there for me during the *low* phases of my PhD and made sure I remained grounded during the *high* phases. I thank him for his constant support and motivation.

I thank Kamal, Sujata, Suresh, Yanjing, Hans, Meena, Arvind, Saket, Vikram, Venkatesh, CRS, Amri, Madhavan, Kumar, Srivatsan, Sreejith, Praveen, Aishwarya and all other faculty members at IMSc and CMI from whom I have learnt a lot, academically and otherwise. I thank them for introducing me to the amazing world of *theoretical computer science*. I also thank the numerous researchers and experts worldwide whose text books, papers and lectures have helped me appreciate the beauty of the subject.

I have been fortunate to have inspiring teachers throughout my life. Thanks to my Ajji, Baby mam, Manorama mam, Kousalya mam, Uma mam, Saleena mam, Mathad sir, Suresh sir, Antony sir, Kumar sir, Ramkrishna sir, KNG sir, CNS sir, Roysten sir, Agnes mam, GK sir, NRP sir and many others who have always been guiding me and motivating me. It is because of them that I am writing this thesis.

My parents have supported me with all the decisions that I have made. I thank them for giving me the freedom to pursue what I want. It is from my dad that I learnt the joy of getting lost in working on something that one is passionate about. My mother taught me that faith can indeed move mountains and sometimes it is better to be patient and have faith than to try and control the situation. Thanks to Pavan for being my friend, intellectual companion and an inspiration.

I thank the administrative staff at IMSc who have been kind and helpful. Special thanks to the house keeping and maintenance staff for making the stay at IMSc as comfortable as possible. They are so dedicated that even when the whole city was under water during the 2015 floods, they ensured that we had access to Canteen and Electricity (and internet!). I thank all my friends from IMSc for a pleasant and memorable life in Chennai. Also, thanks to my friends from Bhadravathi and Mysuru who have always believed in me and supported me. I thank my relatives and cousins for all the love that they have shown towards me.

Finally, everybody who I have crossed path with, has taught me a new perspective towards life. I am deeply indebted to every one of them.

Contents

List of Figures

SUMMARY

Modal logic has been ubiquitously used in many fields of computer science including verification, epistemic logic etc. Typically we have two modal operators $\Box$ and $\Diamond$ which in a broad sense refers to *necessity* and *possibility* respectively. For instance, $\Box_i \alpha$ in an epistemic setting means that “Reasoner i knows that alpha". Similarly, $\Diamond_i \alpha$ in the context of a system of processes is interpreted as “Process i can possibly change the system configuration to a state where α holds". These reasoners or process index are referred to as agents in general.

Classically, the number of agents is assumed to be fixed and finite. But in many settings like multi-process systems / client-server systems / systems with unboundedly many reasoners, we cannot fix the agent set beforehand. The active agents change not only from one model to the other but also from one state to the other in the same model. For instance, in multi-process systems, when the system configuration changes, some processes may be terminated and some new ones may be created.

Term modal logic introduced by Fitting et.al is suitable to study such settings, where we can state properties like $\exists x \forall y\ \Box_x \Box_y \alpha$ which in the epistemic setting translates to “there exists some agent who knows that everybody knows that α".

In this thesis we will explore three main aspects for term modal logic:
(1) Satisfiability problem (2) Bisimulation and (3) Model checking problem.

Satisfiability problem Surprisingly restriction to propositional fragment is of no help. In fact we prove that TML satisfiability problem is undecidable even when $(\top, \bot)$ are used as atoms. Using reductions of tiling problems, we strengthen the result further that the FinSat, UnSat and InfAx are mutually recursively inseparable for TML with atoms restricted only to $e$quality.

These undecidability results motivate us to identify decidable fragments. In this thesis, we identify some decidable fragments of term modal logic: the monodic fragment, the bundled fragment and the two variable fragment.

Bisimulation characterizes modal logics model theoretically. We introduce bisimulation for propositional term modal logic, and prove that it preserves elementary equivalence and the converse holds over image finite models.

Further, we discuss van-benthem type invariance theorem for the variable free fragment called the *implicitly quantified modal logic*. We also tailor the bisimulation to different fragments of TML and use this to compare their expressiveness.

Model Checking When we consider the model checking problem for term modal logic, it is clear that it reduces to classical model checking of First order logic when models are finite, and only complexity issues are interesting. We present these, considering the variants where the model is fixed, or the formula is fixed, or when both are inputs.

When the model is infinite, we need a finite representation to provide input to the algorithm. We consider models where agents are specified as *regular expressions*. These specifications are motivated by consideration of how process identifiers are created in dynamic systems of processes. For such specification, we show that model checking is decidable.

Chapter 1

Introduction

> ... after 13 rounds of nail biting contest, Alice, Bob and Charles were declared as the best logicians in the city. Being friends, the three of them decided to go to a bar and celebrate. In the bar, the bartender welcomed them and asked 'Do all of you want beer?' Alice said, 'I don't know' and so did Bob. Now everybody looked at Charles, who replied with a smile, 'yes'.

How did Charles figure out the answer?[1] This can be explained using *formal logic*. Formal logic, among many other things, is a tool to model various systems and reason about them. It was developed in the late nineteenth and early twentieth century with notable contributions from Frege, Boole, Russell, Tarski, Hilbert and many others.

The above example concerns the knowledge of the participants involved in the situation. Hence, the *logic* that can be used here should be able to refer to the knowledge of Alice, Bob and Charles and reason about it. Similarly, if there was some other scenario which could have been modelled as a graph, then we would need a logic that would allow us to talk about the properties of a graph.

[1] Refer the book on Reasoning about Knowledge [FHMV04] for a detailed formalization.

Thus, depending on the situation that is to be modelled, we have different logics. In general, the syntax part of the logic defines what are all the properties that can be expressed in the logic. The underlying scenario is then represented as an abstract mathematical model. Finally, the semantics of the logic defines the notion of how to check if the given property is true in the model or not.

For the example at hand, the logic suitable to model the setting is called Multi-agent epistemic logic [FHMV04]. The syntax of this logic allows us to talk about properties like : *Alice wants beer, Bob knows that Alice wants beer, Alice knows that Bob does not know that Charles wants beer* etc. The system is then formally represented as an epistemic Kripke-structure and the semantics describes how to check if a property that is expressed in the syntax is true or false.[2]

Initially, logicians were interested in reasoning about mathematical structures like groups, finite fields, graphs, linear orders and so on. For this purpose, Propositional logic (PL) and First-order logic (FO) were ideal candidates.

1.1 Propositional logic

Propositional logic (PL) is a logic of *true/false*. A *proposition* is any 'property' that is either true or false in the situation under consideration. For example, in the context of graphs, in PL we talk of propositions like *graph is connected, graph has an odd cycle* etc which is either *true or false* for any given graph.

Suppose $\mathcal{P}^0 = \{p_0, p_1, p_2, \ldots\}$ is some set of propositions over graphs then with respect to the set $\mathcal{P}^0$, any graph $\mathcal{G}$ can be described as a sequence $\rho_{\mathcal{G}} = (v_0, v_1 \ldots)$ over $\{\mathtt{T}, \mathtt{F}\}$ where $v_i = \mathtt{T}$ if and only if the property p_i is *True* for the graph $\mathcal{G}$.

[2] In the example we need to analyze why the property *Charles knows that (Alice wants beer* and *Bob wants beer* and *Charles wants beer*) is *true* after Charles knows that [Alice does not know that (Alice wants beer and Bob wants beer and Charles wants beer) AND (Bob does not know that (Alice wants beer and Bob wants beer and Charles wants beer)].

Equivalently, $\rho_{\mathcal{G}}$ can be represented as a subset of $\mathcal{P}^0$ where $\rho_{\mathcal{G}} = \{p_i \mid$ property p_i is true in $\mathcal{G}\}$.[3]

In propositional logic over $\mathcal{P}^0$, every proposition $p \in \mathcal{P}^0$ is a property that can be asserted about the underlying model. PL also allows us to assert boolean combinations of these inductively constructed properties. Thus we can assert $\neg\alpha$ (negation of property α) ; $\alpha \wedge \beta$ (property α and property β); $\alpha \vee \beta$ (property α or property β); $\alpha \rightarrow \beta$ (if property α then property β). These are called as *formulas* of PL over $\mathcal{P}^0$. As a norm in the literature, the syntax is formally defined by the grammar:

$$\alpha := p_i \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha$$

where $p_i \in \mathcal{P}^0$.

Thus, any PL formula (property) α over $\mathcal{P}^0$ can be associated with a parse tree that generates α using the above grammar.

Note that every property α is either true or false in any model $\mathcal{G}$. Suppose $\mathcal{G}$ is described as the corresponding $\rho_{\mathcal{G}} \subseteq \mathcal{P}^0$ then there is an inductive way to evaluate whether the property corresponding to α is True or False over $\mathcal{G}$ just by looking at $\rho_{\mathcal{G}}$ and parse tree structure of α.

We denote $\rho_{\mathcal{G}} \models \alpha$ to indicate that the property corresponding to α is True in $\mathcal{G}$. This can be evaluated inductively as follows:

$\rho_{\mathcal{G}} \models p_i$	$\Leftrightarrow$	$p_i \in \rho_{\mathcal{G}}$
$\rho_{\mathcal{G}} \models \neg\alpha$	$\Leftrightarrow$	$\rho_{\mathcal{G}} \not\models \alpha$
$\rho_{\mathcal{G}} \models (\alpha \wedge \beta)$	$\Leftrightarrow$	$\rho_{\mathcal{G}} \models \alpha$ and $\rho_{\mathcal{G}} \models \beta$
$\rho_{\mathcal{G}} \models (\alpha \vee \beta)$	$\Leftrightarrow$	$\rho_{\mathcal{G}} \models \alpha$ or $\rho_{\mathcal{G}} \models \beta$
$\rho_{\mathcal{G}} \models (\alpha \rightarrow \beta)$	$\Leftrightarrow$	$\rho_{\mathcal{G}} \not\models \alpha$ or $\rho_{\mathcal{G}} \models \beta$

[3]If we were talking about groups, then $\mathcal{P}^0$ could have been a set of properties over groups and any group $\mathcal{G} = (A, e, *)$ could have been described as $\rho_{\mathcal{G}} \subseteq \mathcal{P}^0$ where $p_i \in \rho_{\mathcal{G}}$ if and only if the property p_i is *True* for the group $\mathcal{G}$.

For a given $\rho_{\mathcal{G}}$ and a PL formula α, checking whether $\rho_{\mathcal{G}} \models \alpha$ is called the *model checking problem* and this can be done in linear time [EFT13] for PL.

For any logic, one typical computational problem is to check for *satisfiability*, which asks: given a formula φ, is there some model in which φ is true? This is one of the central algorithmic problems for any logic, since it helps us to check if some *desirable property* can ever be achieved or not. This is called the *satisfiability problem*. Note that this problem is in NP [AB09] for PL since a non-determinitic algorithm can guess $\rho_{\mathcal{G}}$ and verify that $\rho_{\mathcal{G}} \models \alpha$. In fact, satisfiability problem for PL is NP-complete [EFT13].

A closely related problem is to ask whether the formula α is true in all models and if it is so, then α is called a *validity*. It is easy to see a formula α is a validity iff $\neg\alpha$ is not satisfiable. Thus, checking if a given PL formula is valid is co-NP-complete [EFT13, AB09].

Propositional logic has very limited expressive power in the sense that it can describe only properties that are either true or false. It does not give access to the underlying structure itself. For instance we might want to talk about edges between vertices in the graph or describe some special properties of the group operator. Such requirements lead us to First order logic.

1.2 First order logic

In the previous section, we represented a graph $\mathcal{G}$ as a sequence over $\{\mathtt{T}, \mathtt{F}\}$ with respect to the set of propositions $\mathcal{P}^0$. Another useful way of representing a graph is to define $\mathcal{G} = (V, E)$ where V is the set of vertices of the graph and $E \subseteq (V \times V)$ is the edge set.[4]

[4]In the context of groups, any group can be represented as $\mathcal{G} = (\mathcal{D}, 0, f)$ where $\mathcal{D}$ is the underlying set on which the group operator acts, $0 \in \mathcal{D}$ is the identity and $f : (\mathcal{D} \times \mathcal{D}) \mapsto \mathcal{D}$ describes the action of $*$ operator.

In such specification, the underlying set (vertices in the case of graphs) is called the domain and the corresponding vocabulary (edges in graphs) is interpreted as a relation over the domain. With such representation, first order logic allows us to access the domain of the given structure using *terms* and use *predicates* to talk about the relation between the domain elements. Further, the syntax of first order logic allows us to quantify over the terms.

For instance, when we consider graphs, there is only one predicate E which represents the edges and this predicate has arity 2. We can talk about formulas like $\forall y\ (x \neq y) \rightarrow E(x, y)$ which asserts *there is an edge to all other vertices from* x. Now in any graph $\mathcal{G} = (V, E)$ there could be some vertex $v \in V$ for which the property is true when x is assumed to be the vertex v and some other vertex $u \in V$ for which the property is false when x is assumed to be the vertex u. Thus, first order logic is more expressive compared to PL since we can *mark* the domain elements of the structure using variables and verify properties over these *marked domain elements*. The variables x, y etc used in the formula are called *terms* and these are interpreted as domain elements. The quantifier $\exists x\ \alpha$ means there is some domain element which on marking with x, makes α true. Similarly, $\forall x\ \alpha$ means that the property α holds no matter which domain element is marked by x.

Thus, FO allows us to talk about the structures in more detail compared to PL. The model checking problem for FO which is to evaluate if a given FO formula is true in a given finite structure is in PTIME (in the size of the input structure) [EFT13]. However, the satisfiability problem for FO, which is to decide whether the given FO formula is true in some structure or not, turns out to be undecidable [EFT13, AB09]. Thus, FO gives more expressive power compared to PL but the computational problems are hard to solve.

1.3 Modal Logic

Some propositions like *Alice is wearing a red shirt* is 'contingent' whereas properties like *2+2=4* seem to be true always (necessary truth). *Modal logic* was initially conceived to study the notion of contingent and necessary truth [Bal14, Ben10b]. The basic modality $\Box\alpha$ asserts *it is necessary that* α and the modality $\Diamond\alpha$ is intended to mean *it is possible that* α. The logic was initially developed by Lewis, Carnap, Hintikka and many others [Bal14].

To model situations where necessary and contingent truths are under consideration, Kripke introduced the *possible world semantics*. Any proposition that is contingent (for instance *Alice is wearing red shirt*) could be either true or false. However, the necessary propositions (for instance 2+2=4) are always true. Thus, in the above case there are at least two *possible worlds*:[5]

- Alice is wearing red shirt and 2+2=4.
- Alice is not wearing red shirt and 2+2=4.

Formally, given a set of propositions $\mathcal{P}^0 = \{p_0, \ldots\}$, a *world* w is associated with $\rho(w) \subseteq \mathcal{P}^0$ such that $p_i \in \rho(w)$ iff property p_i is *true* at the world w. A Kripke structure is given by $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \rho)$ where $\mathcal{W}$ denotes the set of all *possible worlds* and $\rho : \mathcal{W} \mapsto 2^{\mathcal{P}^0}$ where $\rho(w)$ is as described above, for every $w \in \mathcal{W}$. The relation $\mathcal{R} \subseteq (\mathcal{W} \times \mathcal{W})$ encodes the *accessibility relation*. The key idea is that the contingent propositions might be either *true or false* in different accessible worlds but the necessary propositions are true at all accessible worlds [HC96, BdRV01].

The modal operators can be interpreted meaningfully in various other contexts. For instance, Gödel used $\Box\alpha$ to mean *it is provable that* α. Tarski used $\Box\alpha$ to

[5] Note that there might be many more possible worlds depending on other factors: for instance, if Alice's hair is black or not etc. In the above example we have restricted the scenario to consider only two properties as important: red shirt of Alice and 2+2=4.

describe the *topological interior of a set defined by* α. With the works of Prior, Hintikka, Stalnaker and others, modal logic found application in various fields of philosophy and depending on the application these systems were called *Epistemic logic, Deontic logic, Temporal logic, Doxastic logic, Dynamic logic* etc [HC96, BdRV01] which were obtained by enforcing various conditions on the accessibility relation $\mathcal{R}$. For instance, $\Box\alpha$ in epistemic logic is interpreted as *reasoner knows that* α [vDvdHK07, FHMV04] in which $\mathcal{R}$ is restricted to be an equivalence relation. Aumann and others used modal logic to study the notion of knowledge of players in games [OR94].

With the development of theoretical computer science, the usefulness of modal logics increased manyfold [BdRV01]. Notable among them are logics of programs like propositional dynamic logic [FL79] and temporal logics used in the synthesis and verification [Pnu77, AHK02] problems. Since then, modal logic has been ubiquitously used to model various systems that arise in the context of theoretical computer science.

1.3.1 Modality as agency

In epistemic logic, $\Box\alpha$ is intended to mean that *the reasoner knows* α [FHMV04]. However, if there are more than one reasoners involved, then we need to specify which reasoner knows α. Thus, indexing the modal operator with the relevant reasoner seems natural. We have $\Box_{alice}\alpha$ to mean that *Alice knows that* α. In general, we use natural numbers to index the modalities and these indices are assumed to represent the reasoners/agents present in the system. This logic is called *multi-modal logics* where the modal operators are indexed. A typical formula in this logic would be of the form $\Box_1\Box_2\alpha$ which is intended to mean *reasoner* 1 *knows that reasoner* 2 *knows that* α.

Suppose there are n agents, the set of indices is given by $\mathsf{Ag} = \{1, \ldots, n\}$ where every index $i \in \mathsf{Ag}$ corresponds to a particular agent present in the system. In multi-modal logics the modal operators are of the form $\Box_i \alpha$ or $\Diamond_i \alpha$ where $i \in \mathsf{Ag}$.

Having indexed modality is useful in many applications. For instance, the modality $\Box_i$ can be interpreted as *knowledge or belief of reasoner* i [vDvdHK07, FHMV04]; *player* i *move in a game* [OR94]; *configuration update by process* i *in a system* [Pnu77, AHK02] and many more.

1.3.2 Beyond finitely many agents

Note that in multi-modal logic, the underlying agent set Ag is assumed to be fixed and finite. It is fixed beforehand depending on the number of agents involved in the system under consideration. However, there are numerous situations where the agent set cannot be fixed beforehand.

Example 1.1. *Consider the following examples.*

- *Suppose we want to model some social network like Facebook and study how knowledge about a particular incident spreads in the network. In this case, we cannot have a limit on the number of users in the network or have an upper bound on the the number of friends of a particular user. Moreover, the users do not know all the other people in the entire network which in other words means that the names of the users is not common knowledge [GH93, WS18].*

- *Consider a server-client system where the server handles the requests from the clients. If we want to model this, again, we cannot bound the number of clients beforehand. Moreover, the set of clients keeps changing dynamically. Once a server has handled a client's request that client becomes deactivated but during this time there may be new clients who have issued requests.*

- *In a system of processes that can spawn new processes, the number of processes active at any point of time is unbounded. Also, every time the system configuration changes, some processes may be terminated and some new processes may be created.*

A natural question arises in such contexts: how can we model systems where the number of agents cannot be fixed a priori? Such requirements also arise in the context of large games where the number of players is unbounded [PR14].

This leads us to consider modal logics where the modal index set not only varies from structure to structure but also from one possible world to the other. We want the logic used in such context to be able to index the unbounded number of agents.

1.4 Term Modal Logic

Unboundedness of the set of agents implies our inability to name them syntactically but we can refer to such agents by their properties. Quantification serves this purpose well and we may express properties like *there exists an agent x such that* $\Box_x\alpha$ or *for all agent x if x satisfies property φ then* $\Box_x\alpha$.

Term Modal Logic (TML), introduced by Fitting, Thalmann and Voronkov is a natural candidate logic to study unboundedly many agents [FTV01]. TML is built on a generic first order logic where the modalities are indexed by terms which can be quantified over. For instance, we can assert: *all agents who know that it is raining also know that the ground is wet* as $\forall x\ \Box_x(raining) \rightarrow \Box_x(wet\_ground)$. In fact TML can also express agent properties as predicates which allows us to assert properties of the flavour: *All eye-witnesses know who killed Mary* as:
$\exists y \forall x (Wit(x) \rightarrow \Box_x\ killed(y, Mary))$.

1.5 First order modal logic

Term modal logic is closely related to First order modal logic (FOML) [FM99]. Note that propositional modal logic talks about contingent and necessary truth of propositions. On similar lines, FOML allows us to talk about contingent and necessary truths about first order structures. First order multi-modal logic is well suited to study epistemic logics of knowing-how, knowing-why, knowing-what, and so on [Wan17]. For instance, $\exists x\ \Box_i P(x)$ may mean that the reasoner i knows the value of x satisfying P.

Again, multi-modal FOML has fixed and finite agent set $\mathsf{Ag} = \{1, \dots, n\}$ from which the modalities are indexed. A typical multi-modal FOML formula is of the form $\forall x\ \big(\Box_i(P(x) \vee \Diamond_j\ (\exists y\ R(x,y))\big)$ where $i, j \in \mathsf{Ag}$.

Note that term modal logic is different from first order modal logic. The crucial difference is that variables refer to domain elements in FOML and predicates refer to properties of them whereas in TML the predicates refer to properties of agents and variables refer to agents. Moreover, in TML quantified variables act as indices for modalities whereas in multi-modal FOML, modality is still indexed by a fixed finite index set Ag. However, TML can be embedded into FOML and we will discuss this in detail in Chapter 5.

1.6 Thesis Contribution

In this thesis, we investigate three central problems for *term modal logic*: *satisfiability problem, expressivity* and *model checking problem.*

For most of the thesis, we restrict the atoms to propositions which we call *Propositional term modal logic* (PTML). We make this choice since the technical difficulties of quantification over modalities are already manifested in this simpler language.

Clearly, the satisfiability problem for TML is undecidable since formulas of FO are also TML formulas. As we will see, restricting the atoms to propositions is not enough to get decidability. In fact, we will prove that the problem remains undecidable even when the atoms are restricted to just $(\top, \bot)$ or *equality*. This naturally motivates us to identify some decidable fragments. There have been some attempts in the literature to identify decidable fragments of TML [OC17, Sht18, PR19a, PR19c]. In this thesis we prove that the *monodic, bundled* and the *two variable fragments* of TML are decidable.

Bisimulation characterizes modal logics model theoretically. We define the notion of bisimulation appropriately for PTML, and prove that it preserves PTML formulas and that the converse holds over image finite models. The notion of bisimulation for PTML is along the same lines as that of the bisimulation for first order modal logic [Ben10a] with obvious adjustments to suit term modal logic. Since FOML also has predicates, the bisimulation needs to incorporate the notion of *Partial isomorphism* associated with FO. On the other hand, PTML has only propositions as atoms with quantified modalities and this leads to a slightly different way to define the notion of bisimulation which we discuss in detail.

From algorithmic perspective, we analyse the computational complexity issues for deciding whether the given two models are bisimilar or not. Further, we consider a variable free fragment of PTML called the *implicitly quantified modal logic* (IQML) with two modalities which, in an epistemic setting translates to *everybody knows* and *somebody knows*. We give a complete axiom system for IQML and refine the notion to bisimulation which matches the IQML fragment of PTML. We also show that there is a translation of IQML to an appropriate 2-sorted FO and for which 'van Benthem type' characterization theorem holds. We tailor bisimulation for various fragments of PTML which will help us compare their expressive power.

Model checking problem for TML has a lot of potential applications including distributed systems, dynamic network of processes etc. When we consider the model checking problem for TML over finite structures, it is clear that the problem reduces to classical first order model checking, and only complexity issues are interesting which is discussed in the thesis.

On the other hand, when the model is infinite, we first need a finite representation to specify it. Motivated by creation of process id's in dynamic network of processes, we suggest one such finite representation where the agents are specified as *regular expressions* and prove that with such specification, model checking problem for PTML is decidable.

Chapter 2

Preliminaries

A *decision problem* is one where we have to decide *yes/no* for any given input. An algorithm to solve a decision problem is called a *recursive procedure* if it halts on all the inputs and outputs *yes* iff the input is a *yes* instance. Similarly, the algorithm is called *recursively enumerable procedure* if it is guaranteed to halt on all inputs which are *yes* instances and outputs correct answer whenever it terminates. A decision problem is said to be *decidable* if there is a *recursive procedure* to compute it (for more details, refer the books [AB09, Sip06]).

Most of the decidability results discussed in the thesis are supplemented with an analysis of the complexity class to which the problem belongs to. We assume that the reader is familiar with the standard complexity classes like PTIME, NP, PSPACE etc [AB09].

Let $\mathbb{N}$ denote the set of all natural numbers $\{0, 1, 2, \ldots\}$ and $\emptyset$ denote empty set. We use 2^X to denote the power set of X.

Let $\mathcal{P} = \{\mathcal{P}^0, \mathcal{P}^1, \ldots\}$ be a collection of predicates where each $\mathcal{P}^n$ is a countable set of predicates of arity n. Every $p \in \mathcal{P}^0$ is called a *proposition* and every $P \in \mathcal{P}^1$ is called a *unary predicate*.

We use the same collection of predicates $\mathcal{P}$ in the vocabulary for all the logics under consideration that use predicates (first order logic, term modal logic and first order modal logic). We use the same set of propositions $\mathcal{P}^0$ for propositional multi-modal logic (ML^n) and propositional term modal logic (PTML). Also, we use the same set of variables ($\mathcal{V}$) for all the logics that use *terms*.[1]

2.1 Propositional multi-modal logic

As discussed earlier, propositional multi-modal logic has a fixed and finite agent set $\mathsf{Ag} = \{1, \dots, n\}$. The logic is built on propositional logic (PL) by adding indexed modal operators of the form $\Box_i \alpha$ and $\Diamond_i \alpha$ where $i \in \mathsf{Ag}$.

Definition 2.1 (ML^n syntax)**.** *Let $\mathcal{P}^0$ be a countable set of propositions. The syntax of propositional multi-modal logic with agent set $\mathsf{Ag} = \{1, \dots n\}$ is given by:*

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_i\ \varphi$$

where $p \in \mathcal{P}^0$ and $i \in \mathsf{Ag}$.

The boolean connectives $\vee$ (or) and $\to$ (implication) are defined in the standard way where $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \to \psi := (\neg\varphi \vee \psi)$. The dual of $\Diamond_i$ modality is given by $\Box_i \varphi := \neg\Diamond_i(\neg\varphi)$. We denote ML^n to be the set of all formulas with $\mathsf{Ag} = \{1, \dots, n\}$. If Ag is a singleton set then we have ML.

The modal depth of a formula α is inductively defined where for all $p \in \mathcal{P}^0$, $\mathsf{md}(p) = 0$; $\mathsf{md}(\neg\alpha) = \mathsf{md}(\alpha)$; $\mathsf{md}(\alpha \wedge \beta) = \max(\mathsf{md}(\alpha), \mathsf{md}(\beta))$ and $\mathsf{md}(\Diamond_i \alpha) = \mathsf{md}(\alpha) + 1$.

The set of subformulas of a formula α is defined inductively where $\mathsf{SF}(p) = \{p\}$; $\mathsf{SF}(\neg\alpha) = \{\neg\alpha\} \cup \mathsf{SF}(\alpha)$; $\mathsf{SF}(\alpha \wedge \beta) = \{\alpha \wedge \beta\} \cup \mathsf{SF}(\varphi)$ and $\mathsf{SF}(\Diamond_i \alpha) = \{\Diamond_i \alpha\} \cup \mathsf{SF}(\alpha)$.

[1] This will help us translate the formulas from one logic to the other easily.

Definition 2.2 (ML^n structure and semantics). *A Kripke model for ML^n is a tuple $\mathcal{M} = (\mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n, \rho)$ where $\mathcal{W}$ is a non-empty countable set called* worlds*; for every $i \in \mathsf{Ag}$ the* accessibility relation *is given by $\mathcal{R}_i \subseteq (\mathcal{W} \times \mathcal{W})$ and $\rho : \mathcal{W} \to 2^{\mathcal{P}^0}$ is the* valuation.

For any $w \in \mathcal{W}$ and a formula $\varphi \in \mathsf{ML}^n$ define $\mathcal{M}, w \models \varphi$ inductively where:

$$
\begin{array}{lcl}
\mathcal{M}, w \models p & \Leftrightarrow & p \in \rho(w) \\
\mathcal{M}, w \models \neg\varphi & \Leftrightarrow & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models (\varphi \wedge \psi) & \Leftrightarrow & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Diamond_i\, \varphi & \Leftrightarrow & \text{there is some } u \in \mathcal{W} \text{ such that} \\
 & & (w, u) \in \mathcal{R}_i \text{ and } \mathcal{M}, u \models \varphi
\end{array}
$$

A formula φ is *satisfiable* if there is some model $\mathcal{M}$ and some world $w \in \mathcal{W}$ such that $\mathcal{M}, w \models \varphi$. The length of a formula φ (denoted by $|\varphi|$) is the number of nodes in the parse tree of φ from which the formula is generated.

Theorem 2.3 ([BdRV01]). *For all $n \geq 1$, the satisfiability problem for ML^n is* PSPACE*-complete.*

2.2 First order logic

We recall first order logic over relational vocabulary without constants and without function symbols.

Definition 2.4 (FO syntax). *Given a collection of predicates $\mathcal{P} = (\mathcal{P}^0, \mathcal{P}^1, \ldots)$ where each $\mathcal{P}^k$ is a countable set of predicates of arity k, and given a countable set of variables $\mathcal{V}$ the syntax of first order logic is given by:*

$$\alpha \quad := \quad P(x_1, \ldots, x_n) \mid x = y \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists x\ \alpha$$

where $P \in \mathcal{P}^n$ and $x, y, x_1, \ldots, x_n \in \mathcal{V}$.

The boolean connectives $\vee$ and $\rightarrow$ are defined in the standard way. The dual of $\exists$ quantifier is given by $\forall x\ \varphi := \neg\exists x\ (\neg\varphi)$. We denote the set of all first order logic formulas by $\mathsf{FO}_=$ and denote FO to be the set of all formulas that do not contain equality. We denote FO^k (similarly $\mathsf{FO}^k_=$) to be the set of all $\mathsf{FO}(\mathsf{FO}_=)$ formulas that mention at most k-variables.

Given a formula α, the set of free variables of α is defined inductively where for all $P \in \mathcal{P}^n$ we have $\mathsf{FV}(P(x_1, \ldots, x_n)) = \{x_1, \ldots, x_n\}$; $\mathsf{FV}(\neg\alpha) = \mathsf{FV}(\alpha)$; $\mathsf{FV}(\alpha \wedge \beta) = \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta)$ and $\mathsf{FV}(\exists x\ \alpha) = \mathsf{FV}(\alpha) \setminus \{x\}$. We say α is a *sentence* if $\mathsf{FV}(\alpha) = \emptyset$.

The quantifier rank of a formula α (denoted by $\mathsf{qr}(\alpha)$) is inductively defined where for all $P \in \mathcal{P}^n$ we have $\mathsf{qr}(P(x_1, \ldots, x_n)) = 0$; $\mathsf{qr}(\neg\alpha) = \mathsf{qr}(\alpha)$; $\mathsf{qr}(\alpha \wedge \beta) = \max(\mathsf{qr}(\alpha), \mathsf{qr}(\beta))$ and $\mathsf{qr}(\exists x\ \alpha) = \mathsf{qr}(\alpha) + 1$.

Definition 2.5 (FO structure)**.** *An FO structure is given by $\mathfrak{A} = (\mathcal{D}, \rho)$ where $\mathcal{D}$ is a non-empty, countable set called* domain *and $\rho : \mathcal{P} \mapsto \bigcup_i 2^{\mathcal{D}^i}$ is the interpretation of the predicates such that for all $n \geq 1$ and all $P \in \mathcal{P}^n$ we have $\rho(P) \subseteq 2^{\mathcal{D}^n}$ and for all $p \in \mathcal{P}^0$ we have $\rho(p) \in \{\mathcal{D}, \emptyset\}$.*

We assume that a proposition $p \in \mathcal{P}^0$ is true in an FO structure if $\rho(p) = \mathcal{D}$. We use the same convention for first order modal logic and term modal logic also. We do not mention this technicality again, but it is implicitly assumed.

To interpret the variables, we have a function $\sigma : \mathcal{V} \mapsto \mathcal{D}$. For any interpretation function σ and $d \in \mathcal{D}$, let $\sigma_{[x \mapsto d]}$ denote the variant of σ where for all $y \neq x$ we have $\sigma_{[x \mapsto d]}(y) = \sigma(y)$ and $\sigma_{[x \mapsto d]}(x) = d$.

Definition 2.6 (FO semantics)**.** *Given a formula $\varphi \in \mathsf{FO}_=$ and an FO structure $\mathfrak{A} = (\mathcal{D}, \rho)$ and an interpretation of variables $\sigma : \mathcal{V} \mapsto \mathcal{D}$, define $\mathfrak{A}, \sigma \models \varphi$ inductively as follows:*[2]

[2] For all propositions $p \in \mathcal{P}^0$, we say $\mathfrak{A}, \sigma \models p \Leftrightarrow \rho(p) = \mathcal{D}$. We use the same convention for first order modal logic and term modal logic.

$\mathfrak{A}, \sigma \models P(x_1, \ldots, x_n)$	$\Leftrightarrow$	$(\sigma(x_1), \ldots \sigma(x_n) \in \rho(P)$
$\mathfrak{A}, \sigma \models x = y$	$\Leftrightarrow$	$\sigma(x) = \sigma(y)$
$\mathfrak{A}, \sigma \models \neg\varphi$	$\Leftrightarrow$	$\mathfrak{A}, \sigma \not\models \varphi$
$\mathfrak{A}, \sigma \models (\varphi \wedge \psi)$	$\Leftrightarrow$	$\mathfrak{A}, \sigma \models \varphi$ *and* $\mathfrak{A}, \sigma \models \psi$
$\mathfrak{A}, \sigma \models \exists x\ \varphi$	$\Leftrightarrow$	*there is some* $d \in \mathcal{D}$ *such that* $\mathfrak{A}, \sigma_{[x \mapsto d]} \models \varphi$

A formula φ is *satisfiable* if there is some FO structure $\mathfrak{A}$ and some interpretation σ such that $\mathfrak{A}, \sigma \models \varphi$. The set of subformulas of a formula φ is defined as usual where $\mathsf{SF}(P(x_1, \ldots, x_n)) = \{P(x_1, \ldots, x_n)\}$ and $\mathsf{SF}(\exists x\ \varphi) = \{\exists x\ \varphi\} \cup \mathsf{SF}(\varphi)$.

Theorem 2.7. *We recall some classical theorems about first order logic which are relevant for the thesis.*

1. *[Göd33] Let $Q \in \mathcal{P}^2$ be some arbitrary binary predicate and let $\mathsf{FO}^3(Q)$ denote the 3-variable fragment of FO in which the only predicate occurring is Q. The satisfiability problem for $\mathsf{FO}^3(Q)$ is undecidable.*

2. *[EFT13] The satisfiability problem for $\mathsf{FO}_=$ restricted to propositions and unary predicates ($\mathcal{P}^0 \cup \mathcal{P}^1$) is* NEXPTIME*-complete.*

3. *[GKV97] The satisfiability problem for $\mathsf{FO}^2_=$ is* NEXPTIME*-complete.*

2.3 First order multi-modal logic

Note that *propositional multi-modal logic* (ML^n) is built by adding modal operators to propositional logic. Similarly, first order multi-modal logic is built by adding modal operators to first order logic.

Definition 2.8 (multi-modal FOML syntax)**.** *Given a collection of predicates $\mathcal{P} = (\mathcal{P}^0, \mathcal{P}^1, \ldots)$ where each $\mathcal{P}^m$ is a countable set of predicates of arity m, and given a countable set of variables $\mathcal{V}$ and a fixed finite set of agents $\mathsf{Ag} = \{1, \ldots, n\}$, the syntax of first order multi-modal logic over $\mathcal{P}$ is given by:*

$$\alpha := P(x_1, \ldots, x_k) \mid x = y \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists x\ \alpha \mid \Diamond_i\ \alpha$$

where $P \in \mathcal{P}^n$ and $x, x_1, \ldots, x_k \in \mathcal{V}$ and $i \in \mathsf{Ag}$.

The boolean connectives $\vee$ and $\rightarrow$ are defined in the standard way. The dual of $\exists$ quantifier is given by $\forall x\ \varphi := \neg\exists x\ (\neg\varphi)$ and the dual of $\Diamond_i$ modality is given by $\Box_i\alpha := \neg\Diamond_i\neg\alpha$. We denote the set of all first order multi-modal logic formulas by $\mathsf{FOML}^n_=$ (with agent set $\mathsf{Ag} = \{1, \ldots n\}$) and denote FOML^n to be the set of all formulas that do not contain equality. If the agent set Ag is singleton, then we have FOML and $\mathsf{FOML}_=$. Note that the propositional fragment of FOML^n corresponds to ML^n.

Given a formula $\alpha \in \mathsf{FOML}^n_=$, the set of free variables of α is defined inductively as in first order logic where $\mathsf{FV}(\Diamond_i\alpha) = \mathsf{FV}(\alpha)$. The quantifier rank of α (denoted by $\mathsf{qr}(\alpha)$) is also defined as in first order logic where $\mathsf{qr}(\Diamond_i\alpha) = \mathsf{qr}(\alpha)$. The modal depth of a formula α is inductively defined in the standard way as in ML^n where $\mathsf{md}(P(x_1, \ldots, x_n)) = 0$ for all $P \in \mathcal{P}^n$ and $\mathsf{md}(\exists x\ \alpha) = \mathsf{md}(\alpha)$.

The set of all subformulas of a formula α is also defined inductively in the standard way as in first order logic where $\mathsf{SF}(\Diamond_i\alpha) = \{\Diamond_i\alpha\} \cup \mathsf{SF}(\alpha)$.

To define the structures, recall that in the models for ML^n, every world w is associated with a set of propositions that are true at w given by the valuation function $\rho(w) \subseteq \mathcal{P}^0$. In the similar spirit, for FOML^n, every world is to be associated with an FO-structure. Roughly, we want the valuation function to be of the form $\chi : \mathcal{W} \mapsto \Gamma$ where Γ is a collection of FO-structures.

This can be elegantly defined as an FOML^n Kripke structure given by the tuple $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \mathcal{R}_1, \ldots \mathcal{R}_n, \delta, \rho)$ where $\mathcal{W}$ is the non-empty set of worlds and $\mathcal{D}$ is the potential domain set. The function $\delta : \mathcal{W} \mapsto 2^{\mathcal{D}}$ defines the *local domain set* at every world and the valuation function $\rho : (\mathcal{W} \times \mathcal{P}) \mapsto \bigcup_i 2^{\mathcal{D}^i}$ gives the interpretation for the predicates at every world over the corresponding local domain. The accessibility relations $\mathcal{R}_1, \ldots \mathcal{R}_n$ continues to serve the same purpose as in ML^n.

When we associate every world with a first order structure, we encounter a key technical difficulty in evaluating the formulas involving free variables. For instance, when we try to evaluate the formula $\forall x \Box_i(P(x))$ at a world w, then for every $d \in \delta(w)$ and every successor $(w, u) \in \mathcal{R}_i$ we need to evaluate $P(d)$ at u. But for this d has to be present at the world u (or at least we should know how to evaluate $P(d)$ at u).

One way out of this technicality is that we can put a restriction that the FOML^n models to have *increasing domain property* where we impose a monotonicity condition on the function δ with respect to $\mathcal{R}$. Then, for the above example d is always present at u and hence we can evaluate $P(d)$ as u. We will use this monotonicity assumption in the thesis. We will encounter a similar problem for term modal logic also, where we will explain this in more detail. Even there we will assume the monotonicity condition.

There are other ways to deal with this problem of evaluating formulas involving free variables. For instance, we can consider $\delta(w)$ to be an inner domain for every world, but then the valuation function ρ is defined for the entire potential domain set $\mathcal{D}$ (which is called the outer domain). Thus we can always evaluate the predicates involving free variables. In this case, we insist that the witness for existential formulas come from inner domain. We will not consider this approach in this thesis, but use the monotonicity restriction.

Definition 2.9 (FOML^n structure). *An increasing domain model for FOML^n is a tuple $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}_1, \ldots \mathcal{R}_n, \rho)$ where, $\mathcal{W}$ is a non-empty countable set called* worlds*; $\mathcal{D}$ is a non-empty countable set called* domain*; for every $i \in \mathsf{Ag}$, the* accessibility relation *is given by $\mathcal{R}_i \subseteq (\mathcal{W} \times \mathcal{W})$. The map $\delta : \mathcal{W} \mapsto 2^{\mathcal{D}}$ assigns to each $w \in \mathcal{W}$ a non-empty local domain set such that whenever $(w, v) \in \mathcal{R}_i$ we have $\delta(w) \subseteq \delta(v)$ and $\rho : (\mathcal{W} \times \mathcal{P}) \mapsto \bigcup_n 2^{\mathcal{D}^n}$ is the* valuation function *where for all $n \geq 1$ and $P \in \mathcal{P}^n$ we have $\rho(w, P) \subseteq [\delta(w)]^n$ and for $p \in \mathcal{P}^0$ we have $\rho(w, p) \in \{\delta(w), \emptyset\}$.*

Since FOML^n is built on first order logic, to interpret free variables, we need a variable assignment $\sigma : \mathcal{V} \mapsto \mathcal{D}$. Call σ *relevant* at $w \in \mathcal{W}$ if $\sigma(x) \in \delta(w)$ for all $x \in \mathcal{V}$. Given a model $\mathcal{M}$, for any interpretation σ and $d \in \mathcal{D}$, we denote $\sigma_{[x \mapsto d]}$ to be the variant of σ where for all $y \neq x$, $\sigma_{[x \mapsto d]}(y) = \sigma(y)$ and $\sigma_{[x \mapsto d]}(x) = d$.

Definition 2.10 (FOML semantics). *Given an FOML^n model $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ and $w \in \mathcal{W}$, and σ relevant at w, for all $\mathsf{FOML}^n_=$ formula φ define $\mathcal{M}, w, \sigma \models \varphi$ inductively as follows:*

$\mathcal{M}, w, \sigma \models P(x_1, \ldots, x_n)$	$\Leftrightarrow$	$(\sigma(x_1), \ldots, \sigma(x_n)) \in \rho(w, P)$
$\mathcal{M}, w, \sigma \models x = y$	$\Leftrightarrow$	$\sigma(x) = \sigma(y)$
$\mathcal{M}, w, \sigma \models \neg\varphi$	$\Leftrightarrow$	$\mathcal{M}, w, \sigma \not\models \varphi$
$\mathcal{M}, w, \sigma \models (\varphi \wedge \psi)$	$\Leftrightarrow$	$\mathcal{M}, w, \sigma \models \varphi$ *and* $\mathcal{M}, w, \sigma \models \psi$
$\mathcal{M}, w, \sigma \models \exists x\ \varphi$	$\Leftrightarrow$	*there is some* $d \in \delta(w)$ *such that* $\mathcal{M}, w, \sigma_{[x \mapsto d]} \models \varphi$
$\mathcal{M}, w, \sigma \models \Diamond_i\ \varphi$	$\Leftrightarrow$	*there is some* $u \in \mathcal{W}$ *such that* $(w, u) \in \mathcal{R}_i$ *and* $\mathcal{M}, u, \sigma \models \varphi$

A formula φ is *satisfiable* if there is some FOML^n structure $\mathcal{M}$ and $w \in \mathcal{W}$ and some interpretation σ relevant at w such that $\mathcal{M}, w, \sigma \models \varphi$. A formula φ is *valid* if $\neg\varphi$ is not satisfiable.

Let P be a unary predicate and Q be a binary predicate. Consider an FOML model (with single accessibility relation) $\mathcal{M}$ where:

$$\mathcal{W} = \{u, v\} \qquad \mathcal{D} = \{a, b, c\} \qquad \mathcal{R} = \{(u, v)\}$$

$$\delta(u) = \{a, b\} \text{ and } \delta(v) = \{a, b, c\}$$

$$\rho(u, P) = \{a\} \text{ and } \rho(v, P) = \{a, c\}$$

$$\rho(u, Q) = \emptyset \text{ and } \rho(v, Q) = \{(a, c)\}$$

In this model, $\mathcal{M}, u \models \forall x \left(P(x) \to \Box (\exists y\, Q(x, y))\right)$ whereas $\mathcal{M}, u \not\models \forall x \Diamond\, P(x)$.

Note that the modal free part of FOML (with single agent) corresponds to FO and hence satisfiability problem for FOML is undecidable. In fact the two variable fragment of FOML over propositions and unary predicates is already undecidable.

Theorem 2.11 (Rybakov and Shkatov[RS17]). *Satisfiability problem for FOML (with single agent, without equality) for 2-variable fragment restricted to propositions and unary predicates ($\mathcal{P}^0 \cup \mathcal{P}^1$) is undecidable.*

Towards finding decidable fragments, one promising direction is to restrict to *monodic formulas*. An FOML formula φ is *monodic* if every modal subformula of φ of the form $\Delta\psi$ has $|\mathsf{FV}(\psi)| \leq 1$ where $\Delta \in \{\Box, \Diamond\}$. For instance, the formula $\forall x \exists y\, (\Box P(x) \to \Diamond \neg Q(y))$ is a monodic formula whereas $\forall x \exists y\, \Box(P(x) \land \neg Q(y))$ is not a monodic formula.

Theorem 2.12 (Wolter and Zakharyaschev [WZ01]). *Satisfiability problem for monodic FOML formulas over unary predicates is decidable.*

2.4 Countably many modalities

The first step towards considering unboundedly many agents is to hard-wire the names to come from a countably infinite set, instead of a finite set. Thus we can have a variant of propositional modal logic where natural numbers index the modalities.

Definition 2.13. *Let $\mathcal{P}^0$ be a countable set of propositions. The syntax of propositional modal logic with countably many agents (ML^ω) is given by:*

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_i\, \varphi$$

where $p \in \mathcal{P}^0$ and $i \in \mathbb{N}$.

The model is the standard Kripke model as in ML^n but the accessibility relation spans the set of natural numbers ($\mathbb{N}$). The semantics of this logic is standard except that the relation $\mathcal{R}$ in the Kripke model is labelled by natural numbers.

Definition 2.14. *A model for ML^ω is a tuple $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \rho)$ where $\mathcal{W}$ is a non-empty set of worlds, $\mathcal{R} \subseteq (\mathcal{W} \times \mathbb{N} \times \mathcal{W})$ and $\rho : \mathcal{W} \to 2^{\mathcal{P}}$.*

For any $w \in \mathcal{W}$ and a formula $\varphi \in \mathsf{ML}^\omega$ define $\mathcal{M}, w \models \varphi$ where:

$\mathcal{M}, w \models p$	$\Leftrightarrow$	$p \in \rho(w)$
$\mathcal{M}, w \models \neg\varphi$	$\Leftrightarrow$	$\mathcal{M}, w \not\models \varphi$
$\mathcal{M}, w \models (\varphi \wedge \psi)$	$\Leftrightarrow$	$\mathcal{M}, w \models \varphi$ *and* $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models \Diamond_i\, \varphi$	$\Leftrightarrow$	*there is some* $u \in \mathcal{W}$ *such that*
		$(w, i, u) \in \mathcal{R}$ *and* $\mathcal{M}, u \models \varphi$

The notions of a formula $\alpha \in \mathsf{ML}^\omega$ is *satisfiable/valid* is standard. Similarly, the modal depth ($\mathsf{md}(\alpha)$), set of subformulas of ($\mathsf{SF}(\alpha)$) and the length of α ($|\alpha|$) are also defined along standard lines.

For $i \in \mathbb{N}$, let $\mathcal{R}_i = \{(u, v) \mid (u, i, v) \in \mathcal{R}\}$. It is easily seen that this logic has a *bounded agent property*. For any $\mathbb{N}' \subseteq \mathbb{N}$ we say that a model $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \rho)$ is an $\mathbb{N}'$-*agent model* if for all $i \notin \mathbb{N}'$ we have $\mathcal{R}_i = \emptyset$.

Proposition 2.15. *For all formula $\varphi \in \mathsf{ML}^\omega$, φ is satisfiable iff φ is satisfiable in an $\mathbb{N}'$-agent model such that $|\mathbb{N}'| \leq |\varphi|$.*

Proof. It is enough to prove ($\Rightarrow$). Let $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \rho)$ and $w \in \mathcal{W}$ such that $\mathcal{M}, w \models \varphi$. Let $\mathbb{N}' = \{i \mid i \text{ occurs in } \varphi\}$ and define $\mathcal{M}' = (\mathcal{W}, \mathcal{R}', \rho)$ where $\mathcal{R}' = \{(u, i, v) \mid i \in \mathbb{N}' \text{ and } (u, i, v) \in \mathcal{R}\}$. Since the formula can mention at most $|\varphi|$ many indices, we have $|\mathbb{N}'| \leq |\varphi|$ and by construction, $\mathcal{M}'$ is an $\mathbb{N}'$-agent model. Thus it is enough to show that $\mathcal{M}', w \models \varphi$. For this we prove a stronger claim.

Claim. For all $u \in \mathcal{W}$ and for all $\psi \in \mathsf{SF}(\varphi)$ we have $\mathcal{M}, u \models \psi$ iff $\mathcal{M}', u \models \psi$.

The proof is by induction on the structure of ψ. In the base case ψ is a proposition of the form $p \in \mathcal{P}^0$. Hence $\mathcal{M}, u \models p$ iff $u \in \rho(u)$ iff $\mathcal{M}', u \models p$.

For the case $\neg\psi$, we have $\mathcal{M}, u \models \neg\psi$ iff $\mathcal{M}, u \not\models \psi$ iff (by induction hypothesis) $\mathcal{M}', u \not\models \psi$ iff $\mathcal{M}', u \models \neg\psi$.

For the case $\psi \wedge \psi'$, we have $\mathcal{M}, u \models \psi \wedge \psi'$ iff $\mathcal{M}, u \models \psi$ and $\mathcal{M}, u \models \psi'$ iff (by induction hypothesis) $\mathcal{M}', u \models \psi$ and $\mathcal{M}', u \models \psi'$ iff $\mathcal{M}', u \models \psi \wedge \psi'$.

For the case $\Diamond_i\psi$, first note that $i \in \mathbb{N}'$. Thus we have $\mathcal{M}, u \models \Diamond_i\psi$ iff there is some $(u, i, v) \in \mathcal{R}$ and $\mathcal{M}, v \models \psi$ iff (by construction) $(u, i, v) \in \mathcal{R}'$ and (by induction hypothesis) $\mathcal{M}', v \models \psi$ iff $\mathcal{M}', u \models \Diamond_i\psi$.

Since $\varphi \in \mathsf{SF}(\varphi)$, by claim we have $\mathcal{M}', w \models \varphi$. □

The theorem implies that no formula in this logic can force unboundedly many agents. Once a formula φ is specified, the number of agents required is known and thus the satisfiability problem reduces to the case of propositional multi-modal logic (ML^n) for some $n \leq |\varphi|$. Thus, from Theorem 2.3 it follows that:

Theorem 2.16. *Satisfiability problem for ML^ω is* PSPACE*-complete.*

In the logic ML^ω, the syntax ignores the essential problem that arises when we work with unboundedly many agents, namely that the identity of an agent is not known when we are specifying the system. We would then want to access the modal index via quantification. This consideration naturally motivates term modal logic.

Chapter 3

Term modal logic

Term Modal Logic (TML) was introduced by Fitting, Thalmann and Voronkov [FTV01] in which the modalities are indexed by terms which can be quantified over. We first introduce the language of term modal logic without constant symbols in the vocabulary. Further, we discuss some conventions and notations for various fragments like propositional fragment, formulas with(out) equality etc and discuss extending the language with constants. We do not consider function symbols in the language throughout.

3.1 TML syntax

Definition 3.1 (TML syntax). *Given a collection of predicates* $\mathcal{P} = (\mathcal{P}^0, \mathcal{P}^1, \dots)$ *where each* $\mathcal{P}^n$ *is a countable set of predicates of arity* n*, and given a countable set of variables* $\mathcal{V}$ *the syntax of term modal logic is defined by*

$$\varphi \ := \ P(x_1, \dots, x_n) \mid x = y \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \exists x \ \varphi \mid \Diamond_x \ \varphi$$

where $P \in \mathcal{P}^n$ *and* $x, y, x_1, \dots, x_n \in \mathcal{V}$.

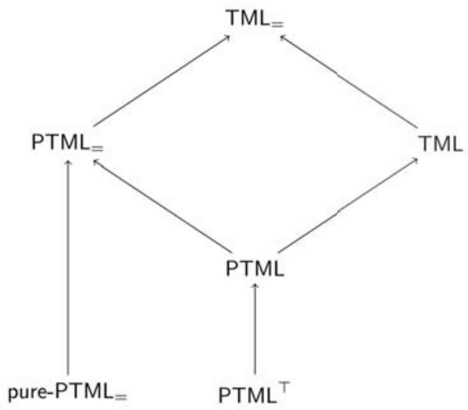


Figure 3.1: Some of the syntactic fragments of term modal logic considered in the thesis. The arrow indicates the inclusion relation.

The boolean operators $\vee$ and $\to$ and $\forall$ quantifier are defined in the standard way. The dual of $\Diamond_x$ modality is given by $\Box_x\varphi := \neg\Diamond_x\neg\varphi$. Define bi-implication $\varphi \Leftrightarrow \psi := (\varphi \to \psi) \wedge (\psi \to \varphi)$. Let $\top \in \mathcal{P}^0$ be a distinguished proposition and $\bot := \neg\top$.

We denote the set of all term modal logic formulas by $\mathsf{TML}_=$ and TML denotes the set of all formulas without equality. Also, we denote propositional fragment of term modal logic by PTML ($\mathsf{PTML}_=$) which is the set of all term modal logic formulas which mention only propositions $\mathcal{P}^0$ (and equality). Further, we denote $\mathsf{pure\text{-}PTML}_=$ to be the set of all formulas that does not mention any predicates or propositions (only equality is allowed) and $\mathsf{PTML}^\top$ to be the set of all formulas that mention only $\top$ and $\bot$ as atomic propositions (no equality). Figure 3.1 describes the relation between these syntactic fragments.

For any formula φ, the notion of free variables is defined in the standard way as in first order logic where $\mathsf{FV}(\Diamond_x\, \varphi) = \{x\} \cup \mathsf{FV}(\varphi)$. We say that φ is a *sentence* if $\mathsf{FV}(\varphi) = \emptyset$. Similarly, $\mathsf{qr}(\varphi)$ is the quantifier rank of φ which is also standard where $\mathsf{qr}(\Diamond_x\varphi) = \mathsf{qr}(\varphi)$. The modal depth of a formula (denoted by $\mathsf{md}(\varphi)$) is again standard, where $\mathsf{md}(\Diamond_x\, \varphi) = \mathsf{md}(\varphi) + 1$.

The subformulas of a given formula φ, denoted by $\mathsf{SF}(\varphi)$ is also defined in the natural way where $\mathsf{SF}(\Diamond_x \varphi) = \{\Diamond_x \varphi\} \cup \mathsf{SF}(\varphi)$. Let the negation closure of $\mathsf{SF}(\varphi)$ be given by $\overline{\mathsf{SF}}(\varphi) = \mathsf{SF}(\varphi) \cup \{\neg\psi \mid \psi \in \mathsf{SF}(\varphi)\}$

Given a $\mathsf{TML}_=$ formula φ and variables $x, y \in \mathcal{V}$, we write $\varphi[y/x]$ for the formula obtained by replacing every occurrence of x by y in φ. If $\mathsf{FV}(\varphi) \subseteq \{x_1, \ldots, x_n\}$ then we sometimes make it explicit by writing φ as $\varphi(x_1, \ldots, x_n)$. The length of a formula φ (denoted by $|\varphi|$) is simply the number of nodes in the parse tree that generates the formula φ.

3.2 TML semantics

Recall that term modal logic is motivated from the settings where the agent set cannot be fixed beforehand. Thus, every model comes with its own set of potential agents. Formally, in the model description, along with a set of worlds $\mathcal{W}$ we also have a potential agent set $\mathcal{D}$. Now, accessibility relation has to be labelled by the potential agent set $\mathcal{D}$ and is given by $\mathcal{R} \subseteq (\mathcal{W} \times \mathcal{D} \times \mathcal{W})$. We also allow the set of *relevant agents* to change from world to world and this agent dynamics is captured by a function ($\delta : \mathcal{W} \mapsto 2^{\mathcal{D}}$ below) that specifies, at any world w, the set of agents *live* (or meaningful) at w. The condition that whenever $(u, d, v) \in \mathcal{R}$, we have that $d \in \delta(u)$ ensures only an agent live at u can consider v accessible. Finally, the function $\rho : (\mathcal{W} \times \mathcal{P}) \mapsto \bigcup_{n \in \omega} 2^{\mathcal{D}^n}$ gives the valuation for predicates at every world over the local agent set.

We impose *monotonicity* condition on the δ function with respect to the accessibility relation: whenever $(u, d, v) \in \mathcal{R}$, we have that $\delta(u) \subseteq \delta(v)$. This restriction is on the same lines of that we have for first order modal logic (Refer Page 19), which is to handle the interpretation of free variables. Hence, the models are called *increasing agent* models.

Definition 3.2 (TML structure)**.** *An increasing agent model for term modal logic is a tuple* $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ *where,* $\mathcal{W}$ *is a non-empty countable set called* worlds*;* $\mathcal{D}$ *is a non-empty countable set called* agents*;* $\mathcal{R} \subseteq (\mathcal{W} \times \mathcal{D} \times \mathcal{W})$ *is the* accessibility relation *and* $\delta : \mathcal{W} \mapsto 2^{\mathcal{D}}$ *is called* live agent function *which maps every* $w \in \mathcal{W}$ *to a non-empty subset of* $\mathcal{D}$ *s.t if* $(w, d, v) \in \mathcal{R}$ *then* $d \in \delta(w) \subseteq \delta(v)$*; and the valuation function is given by* $\rho : (\mathcal{W} \times \mathcal{P}) \mapsto \bigcup_n 2^{\mathcal{D}^n}$ *where for all* $n \geq 1$ *and* $P \in \mathcal{P}^n$ *we have* $\rho(w, P) \subseteq [\delta(w)]^n$ *and for all* $p \in \mathcal{P}^0$ *we have* $\rho(w, p) \in \{\delta(w), \emptyset\}$*.*

For a given model $\mathcal{M}$, we use $\mathcal{W}^{\mathcal{M}}, \mathcal{D}^{\mathcal{M}}, \delta^{\mathcal{M}}, \mathcal{R}^{\mathcal{M}}, \rho^{\mathcal{M}}$ to refer to the corresponding components. We drop the superscript when $\mathcal{M}$ is clear from the context. We sometimes write $(w, d, u) \in \mathcal{R}$ as $w \xrightarrow{d} u$ and write $\delta(w)$ as $\mathcal{D}_w$. If $c \in \delta(w)$ then we say *c is live at w*. A *constant agent* model is one where $\mathcal{D}_w = \mathcal{D}$ for all $w \in \mathcal{W}$. A model $\mathcal{M}$ is said to be a *finite model* if both $\mathcal{W}$ and $\mathcal{D}$ are finite.

To interpret free variables, we need a variable assignment $\sigma : \mathcal{V} \mapsto \mathcal{D}$. We say that σ *relevant* at $w \in \mathcal{W}$ if $\sigma(x) \in \delta(w)$ for all $x \in \mathcal{V}$. For all $x \in \mathcal{V}$ and $d \in \mathcal{D}$, we denote $\sigma_{[x \mapsto d]}$ to be the variant of σ where for all $y \neq x$, $\sigma_{[x \mapsto d]}(y) = \sigma(y)$ and $\sigma_{[x \mapsto d]}(x) = d$.

Observation 3.3. *In any TML model $\mathcal{M}$ and any $w \in \mathcal{W}$ and for any interpretation σ, the following holds:*

1. *if σ is relevant at w then for all $d \in \mathcal{D}_w$ and for all $(w, d, u) \in \mathcal{R}$, σ is relevant at u.*

2. *if σ is relevant at w then for any $d \in \mathcal{D}_w$ the variant $\sigma_{[x \mapsto d]}$ is relevant at w.*

The first observation follows from the monotonicity property of δ and the second observation follows from the definition of $\sigma_{[x \mapsto d]}$. Also note that in a constant agent model, every assignment σ is relevant at all the worlds.

Definition 3.4 (TML semantics). *Given a TML model $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ and $w \in \mathcal{W}$, and σ relevant at w, for all $\varphi \in \mathsf{TML}_=$, define $\mathcal{M}, w, \sigma \models \varphi$ inductively where:*

$\mathcal{M}, w, \sigma \models P(x_1, \ldots, x_n)$	$\Leftrightarrow$	$(\sigma(x_1), \ldots, \sigma(x_n)) \in \rho(w, P)$
$\mathcal{M}, w, \sigma \models x = y$	$\Leftrightarrow$	$\sigma(x) = \sigma(y)$
$\mathcal{M}, w, \sigma \models \neg\varphi$	$\Leftrightarrow$	$\mathcal{M}, w, \sigma \not\models \varphi$
$\mathcal{M}, w, \sigma \models (\varphi \wedge \psi)$	$\Leftrightarrow$	$\mathcal{M}, w, \sigma \models \varphi$ *and* $\mathcal{M}, w, \sigma \models \psi$
$\mathcal{M}, w, \sigma \models \exists x\ \varphi$	$\Leftrightarrow$	*there is some* $d \in \delta(w)$ *such that* $\mathcal{M}, w, \sigma_{[x \mapsto d]} \models \varphi$
$\mathcal{M}, w, \sigma \models \Diamond_x\ \varphi$	$\Leftrightarrow$	*there is some* $u \in \mathcal{W}$ *such that* $(w, \sigma(x), u) \in \mathcal{R}$ *and* $\mathcal{M}, u, \sigma \models \varphi$

Note that from Obs. 3.3, it is clear that all inductive definitions only deal with relevant interpretations. We often abuse notation and say 'for all w and for all interpretations σ', when we mean 'for all w and for all interpretations σ relevant at w' (and we will ensure that relevant σ are used in proofs).

For the distinguished proposition $\top \in \mathcal{P}^0$, for all models $\mathcal{M}$ and for all $w \in \mathcal{W}^{\mathcal{M}}$ we have $\mathcal{M}, w \models \top$. In general, when considering the truth of φ in a model, it suffices to consider $\sigma : \mathsf{FV}(\varphi) \mapsto \mathcal{D}$, assignment restricted to the variables occurring free in φ. When $\mathsf{FV}(\varphi) \subseteq \{x_1, \ldots, x_n\}$ and $\overline{d} \in [\mathcal{D}_w]^n$ is a vector of length n over $\mathcal{D}_w$, we sometimes write $\mathcal{M}, w \models \varphi[\overline{d}]$ to denote $\mathcal{M}, w, \sigma \models \varphi(\overline{x})$ where for all $i \leq n$, $\sigma(x_i) = d_i$. When φ is a sentence, we simply write $\mathcal{M}, w \models \varphi$. A formula φ is *satisfiable* if there is some model $\mathcal{M}$ and some world $w \in \mathcal{W}$ and some interpretation σ relevant at w such that $\mathcal{M}, w, \sigma \models \varphi$. A formula φ is *valid* if $\neg\varphi$ is not satisfiable. Two formulas φ and ψ are *equivalent* if for all model $\mathcal{M}$ and all $w \in \mathcal{W}$ and all σ relevant at w we have $\mathcal{M}, w, \sigma \models \varphi$ iff $\mathcal{M}, w, \sigma \models \psi$.

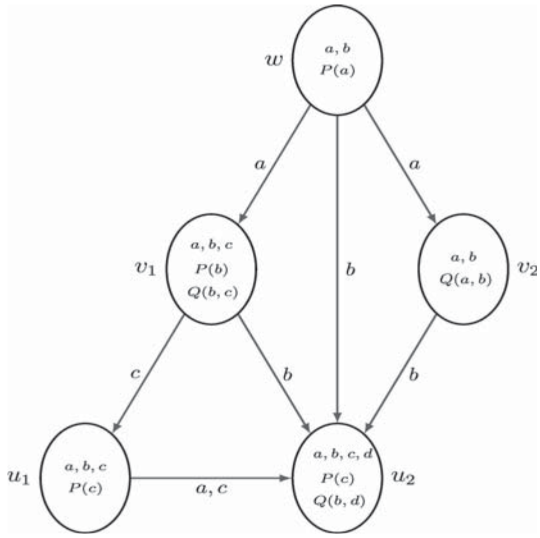


Figure 3.2: Illustration of a term modal logic model

3.3 Example

Consider the increasing model described in the Figure 3.2 given by $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ where:

- $\mathcal{W} = \{w, v_1, v_2, u_1, u_2\}$ and $\mathcal{D} = \{a, b, c, d\}$
- $\delta(w) = \delta(v_2) = \{a, b\};\ \delta(v_1) = \delta(u_1) = \{a, b, c\};\ \delta(u_2) = \{a, b, c, d\}$
- $\mathcal{R} = \{(w, a, v_1), (w, a, v_2), (w, b, u_2)\} \cup \{(v_1, c, u_1), (v_1, b, u_2), \}\cup$
  $\{(v_2, b, u_2)\} \cup \{(u_1, a, u_2), (u_1, c, u_2)\}$
- P is a unary predicate and Q is a binary predicate. The ρ function is defined as follows:

	w	v_1	v_2	u_1	u_2
P	$\{a\}$	$\{b\}$	$\emptyset$	$\{c\}$	$\{c\}$
Q	$\emptyset$	$\{(b, c)\}$	$\{(a, b)\}$	$\emptyset$	$\{(b, c)\}$

Now consider the formula $\forall x \Diamond_x\ (\exists y\ Q(x, y))$. We can verify that $\mathcal{M}, w \models \forall x \Diamond_x\ (\exists y\ Q(x, y))$. This is because when x is assigned a we have $\mathcal{M}, v_2 \models \exists y\ Q(a, y)$ and when x is assigned b we have $\mathcal{M}, u_2 \models \exists y\ Q(b, y)$.

3.4 Monotone condition of δ

Recall that in the TML models, we have imposed monotonicity condition on δ with respect to $\mathcal{R}$, which states that whenever $(w, d, u) \in \mathcal{R}$ we have $\delta(w) \subseteq \delta(u)$.

Suppose this restriction was not there, then assume for now that in the model described in Fig. 3.2 we had $\delta(v_1) = \{a, c\}$ and hence it violates $\delta(w) \subseteq \delta(v_1)$ since $b \in \delta(w)$ but $b \notin \delta(v_1)$. Now consider a formula $\forall x \forall y \Box_x P(y)$. Suppose we want to evaluate whether $\mathcal{M}, w \models \forall x \forall y \Box_x P(y)$. By semantics, in particular, we need to verify that $\mathcal{M}, w \models \Box_a P(b)$. For this, we need to verify that $\mathcal{M}, v_1 \models P(b)$. But if $b \notin \delta(v_1)$ then $\mathcal{M}, v_1 \models P(b)$ is not well defined. This problem can be avoided if we ensure that the agents assigned to free variables are always alive at the successor worlds. The monotonicity condition achieves exactly this. The monotone condition can be interpreted as the restriction where new agents are allowed to be born but agents do not die.

Note that there are other ways to deal with this problem without imposing the monotonicity condition. Analogous to the first order modal logic, we can consider $\delta(w)$ to be the inner agent set for every world. But then, the valuation function ρ is defined for the entire potential agent set $\mathcal{D}$, thereby solving the problem of evaluating the predicates involving free variables. In this case, the witness for existential formulas come from inner agent set.

We will not consider this approach in this thesis, but use the monotonicity restriction.

3.5 Negation normal form

A literal is a predicate (proposition) or its negation. A formula φ is said to be in negation normal form (NNF) if negation appears only in literals occurring in φ.

If we consider $\vee, \Box_x$ and $\forall$ operators explicitly (and not as derived operators), then we can push the negations to atoms and get an equivalent NNF for every TML formula.

Observation 3.5. *For all TML formula φ, there is a formula ψ which is in negation normal form such that φ and ψ are equivalent.*

We can use the following validities and push the negation to the literals to get the required negation normal form.

$$
\begin{array}{ccccccc}
\neg(\alpha \wedge \beta) & \Leftrightarrow & (\neg\alpha \vee \neg\beta) & \text{and} & \neg(\alpha \vee \beta) & \Leftrightarrow & (\neg\alpha \wedge \neg\beta) \\
\neg\Diamond_x\alpha & \Leftrightarrow & \Box_x\neg\alpha & \text{and} & \neg\Box_x\alpha & \Leftrightarrow & \Diamond_x\neg\alpha \\
\neg\exists x\ \alpha & \Leftrightarrow & \forall x\neg\alpha & \text{and} & \neg\forall x\ \alpha & \Leftrightarrow & \exists x\neg\alpha
\end{array}
$$

3.6 Propositional Term modal logic

In this thesis we consider a special case of TML, where the atoms are restricted to propositions (eg. $\forall x\Box_x p \to \exists y\Diamond_y\Box_x(\neg p)$). Note that the variables still occur as the index of modalities. We call this fragment *Propositional term modal logic* (PTML). Analogously we have $\mathsf{PTML}_=$ which is the propositional fragment of $\mathsf{TML}_=$. The formulas that contain only equality (no propositions) as atoms are denoted by pure-$\mathsf{PTML}_=$. These fragments help us to forget about the complications due to predicates and focus on the effect of quantification over modal indices in isolation.

Recall that in the specification of TML models, for all propositions $p \in \mathcal{P}^0$ we have the convention that $\delta(w,p) \in \{\delta(w), \emptyset\}$ and $\mathcal{M}, w \models p$ if $\rho(w,p) = \delta(w)$. Given a TML model $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ if we are working with PTML (or $\mathsf{PTML}_=$), we can equivalently specify $\mathcal{M}$ as $\mathcal{M}' = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho')$ the valuation function ρ' is simply a map $\rho : \mathcal{W} \mapsto 2^{\mathcal{P}^0}$ where $\mathcal{P}^0$ is the set of propositions such that $\rho'(w) = \{p \mid \rho(w,p) = \mathcal{D}\}$. With respect to PTML (or (pure)-$\mathsf{PTML}_=$), both $\mathcal{M}$ and $\mathcal{M}'$ are the equivalent.

Thus, whenever we consider PTML or $\mathsf{PTML}_=$, without loss of generality we assume that the models have the valuation function of the form $\rho : \mathcal{W} \mapsto 2^{\mathcal{P}^0}$.

Now we illustrate that the fragment $\mathsf{PTML}_=$ can already define some non-trivial properties. Consider the formula $\alpha := \forall x \exists y \left(\Diamond_x \Diamond_y \top \wedge \forall z \left((\Diamond_x \Diamond_z \top) \to (y = z) \right) \right)$. Let the inner formula $\beta := \Diamond_x \Diamond_y \top \wedge \forall z \left((\Diamond_x \Diamond_z \top) \to (y = z) \right)$.

Proposition 3.6. *For any PTML model $\mathcal{M}$, suppose $\mathcal{M}, w \models \alpha$ then let*
$\mathrm{G} = \{(c,d) \mid \mathcal{M}, w, [x \mapsto c, y \mapsto d] \models \beta\}$. *Then* G *defines a function.*

Proof. To verify this, first note that since $\mathcal{M}, w \models \forall x \exists y\ \beta$, for every $c \in \delta(w)$ there is at least one $d \in \delta(w)$ such that $\mathcal{M}, w, [x \mapsto c, y \mapsto d] \models \beta$. Now we need to prove that for all $c, d, d' \in \delta(w)$ if $\{(c,d),\ (c,d')\} \subseteq \mathrm{G}$ then $d = d'$. To see this, assume $\{(c,d),\ (c,d')\} \subseteq \mathrm{G}$.

Now since $(c,d) \in \mathrm{G}$ we have $\mathcal{M}, w, [x \mapsto c, y \mapsto d] \models \forall z\ (\Diamond_x \Diamond_z \top) \to (y = z)$ which implies $\mathcal{M}, w, [x \mapsto c, y \mapsto d, z \mapsto d'] \models (\Diamond_x \Diamond_z \top) \to (y = z)$. Again by assumption since $(c,d') \in \mathrm{G}$ we have $\mathcal{M}, w, [x \mapsto c, z \mapsto d'] \models \Diamond_x \Diamond_z \top$ and hence $\mathcal{M}, w, [x \overset{c}{\mapsto}, y \mapsto d, z \mapsto d'] \models (y = z)$ which implies $d = d'$. $\square$

Thus, PTML which mentions only equality and $\top$ can already define functions. We can also state non-trivial properties of such functions. For instance, $\alpha \wedge \forall x \forall y \forall z \big((\Diamond_x \Diamond_z \top \wedge \Diamond_y \Diamond_z \top) \to (x = y) \big)$ ensures that the induced function is a bijection. We analyze the fragments $\mathsf{PTML}^\top$ and $\mathsf{pure\text{-}PTML}_=$ in detail in the next chapter and in fact show that both of them are undecidable.

3.7 Adding Constants to the vocabulary

Let $\mathbf{C}$ be a countable set of constants. To extend term modal logic by adding constants, in the syntax we have *terms* which are either constants or variables.

These terms can appear as components of some predicates or as indices of modalities or in the form $t = t'$ where t, t' are terms. The model specification now comes with an interpretation for the constants in the vocabulary. Note that the interpretation for constants can be specified in two ways:

1. One global interpretation function for all the constants.

2. Every world has a local interpretation function for the constants.

These two approaches are referred to as rigid constant interpretation and non-rigid constant interpretation respectively. Both these approaches have been considered in the literature. In [FTV01], where term modal logic is introduced, Fitting, Thalmann and Voronkov consider rigid constant interpretation which does not change from world to world. On the other hand, term modal logic has been studied in dynamic epistemic setting by Kooi [Koo07] where the interpretation of constants vary from world to world.

In fact we can consider other variants. For instance, Wang and Seligman ([WS18]) study a restricted version of term modal logic where we have assignments in place of quantifiers. In this variant, we have formulas of the form $[x := b]K_x(\alpha)$ where b is a constant, whose interpretation as an agent will be assigned to x. This fragment is motivated from modelling knowledge in a system of agents where the identity of the agents are not common knowledge.

However, if there are no constants in the vocabulary, then this technicality of rigid and non-rigid interpretation goes away. In this thesis, all the results are stated for the vocabulary without constants and at the end of each chapter, we briefly discuss how the results of that chapter extends when we have constants in the vocabulary.

Chapter 4

Satisfiability problem: Undecidability

Note that the set of all modal free formulas of term modal logic is the same as the set of all first order logic formulas. Thus, it follows that the satisfiability problem for TML (and hence $\mathsf{TML}_=$) is undecidable.

One obvious direction towards obtaining a decidable fragment is to restrict the atoms to propositions and consider PTML. In the previous chapter, we saw that term modal logic with just $(\top, \bot)$ and equality can already define functions. Note that with respect to the atoms, $\mathsf{PTML}^\top$ and $\mathsf{pure\text{-}PTML}_=$ the strongest possible restriction where the atoms are restricted to $(\top, \bot)$ and *equality* respectively.

In this chapter we first prove that the satisfiability problem continues to be undecidable for $\mathsf{PTML}^\top$. For $\mathsf{pure\text{-}PTML}_=$, we prove a 'Trakhtenbrot like' theorem i.e, the $\mathsf{FinSat}, \mathsf{UnSat}$ and InfAx are mutually recursively inseparable.

Note that when we consider $\mathsf{PTML}^\top$ and $\mathsf{pure\text{-}PTML}_=$, the valuation function is irrelevant. Hence we drop ρ from the model specification altogether when we consider these fragments.

4.1 Undecidability of $\mathsf{PTML}^{\top}$

For $\mathsf{PTML}^{\top}$ without loss of generality we can consider the models to be $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R})$ (dropping ρ function) where $\mathcal{M}, w \models \top$ for all $w \in \mathcal{W}$. In this section we will prove that the satisfiability problem for $\mathsf{PTML}^{\top}$ is undecidable. The proof is via reduction from FO satisfiability over a single binary predicate.

Let $Q \in \mathcal{P}^2$ be a binary predicate. Let $\mathsf{FO}(Q)$ be the set of all first order logic formulas (without equality) that mention only the predicate Q. The satisfiability problem for $\mathsf{FO}(Q)$ is known to be undecidable [Göd33]. We reduce the satisfiability problem of $\mathsf{FO}(Q)$ to satisfiability of $\mathsf{PTML}^{\top}$ thus proving that the satisfiability problem for $\mathsf{PTML}^{\top}$ is undecidable.

In [Kri62], Kripke proved that the satisfiability problem for first order modal logic with unary predicates (predicates with arity 1) is undecidable. In the proof he uses the reduction from the satisfiability of $\mathsf{FO}(Q)$. In particular he encodes $Q(x, y)$ as $\Diamond(P(x) \wedge R(y))$ where P and R are two unary predicates. This translation obtains an equi-satisfiable FOML formula thus proving the result. We will use the same idea and encode $Q(x, y)$ as $\Diamond_x \Diamond_y \top$ which gives us an equi-satisfiable $\mathsf{PTML}^{\top}$ formula.

Definition 4.1 ($\mathsf{FO}(Q)$ to $\mathsf{PTML}^{\top}$ translation)**.** *Given any $\mathsf{FO}(Q)$ formula φ, the translation of φ to a $\mathsf{PTML}^{\top}$ formula is defined inductively as follows:*

- $\mathit{Tr}_1(Q(x, y)) = \Diamond_x \Diamond_y \top$
- $\mathit{Tr}_1(\neg\varphi) = \neg\ \mathit{Tr}_1(\varphi)$
- $\mathit{Tr}_1(\varphi \wedge \psi) = \mathit{Tr}_1(\varphi) \wedge \mathit{Tr}_1(\psi)$
- $\mathit{Tr}_1(\exists x\ \varphi) = \exists x\ \mathit{Tr}_1(\varphi)$

Note that the translation gives us a $\mathsf{PTML}^\top$ formulas of modal depth 2. All the quantifiers appear outside the scope of modalities in the translated formula and the quantifier depth, number of variables are preserved.

Recall that an $\mathsf{FO}(Q)$ structure is given by $\mathfrak{A} = (\mathcal{D}, \mathcal{I})$ where $\mathcal{D}$ is the domain and $\mathcal{I} \subseteq (\mathcal{D} \times \mathcal{D})$ is the interpretation of the binary predicate Q and in the semantics we have $\mathfrak{A}, [x \mapsto c, y \mapsto d] \models Q(x, y)$ iff $(c, d) \in \mathcal{I}$.

Theorem 4.2. *For any sentence $\varphi \in \mathsf{FO}(Q)$, φ is satisfiable in some $\mathsf{FO}(Q)$ structure iff $\mathsf{Tr}_1(\varphi)$ is satisfiable in some TML model.*

Proof. ($\Rightarrow$) Suppose $(\mathcal{D}, \mathcal{I}) \models \varphi$, define the TML model $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R})$ (note that we use domain $\mathcal{D}$ of $\mathsf{FO}(Q)$ structure as the agent set in $\mathcal{M}$) where:

- $\mathcal{W} = \{r\} \cup \{u_{cd}, v_{cd} \mid (c, d) \in \mathcal{I}\}$ and for all $w \in \mathcal{W},\ \delta(w) = \mathcal{D}$.
- $\mathcal{R} = \{(r, c, u_{cd}) \mid u_{cd} \in \mathcal{W}\} \cup \{(u_{cd}, d, v_{cd}) \mid u_{cd}, v_{cd} \in \mathcal{W}\}$

Figure 4.1 illustrates an example translation. Note that $\mathcal{M}$ is a constant domain model with agent set $\mathcal{D}$. It is sufficient to prove that $\mathcal{M}, r \models \varphi$. For this, we set up the following induction.

Claim. For all $\psi \in \mathsf{SF}(\varphi)$ and for all interpretation $\sigma : \mathcal{V} \mapsto \mathcal{D}$ we have $\mathcal{D}, \mathcal{I}, \sigma \models \psi$ iff $\mathcal{M}, r, \sigma \models \mathsf{Tr}_1(\psi)$.

In the base case, we have $Q(x, y)$ and $\mathsf{Tr}_1(Q(x, y)) = \Diamond_x \Diamond_y \top$. For any interpretation $\sigma : \mathcal{V} \mapsto \mathcal{D}$, let $\sigma(x) = c$ and $\sigma(y) = d$.

If $\mathcal{D}, \mathcal{I}, \sigma \models Q(x, y)$ then $(c, d) \in \mathcal{I}$ iff $r \xrightarrow{c} u_{cd} \xrightarrow{d} v_{cd}$ iff $M, r, \sigma \models \Diamond_x \Diamond_y \top$. On the other hand, if $M, r, \sigma \models \Diamond_x \Diamond_y \top$ then there is some $r \xrightarrow{c} u \xrightarrow{d} v$ in $\mathcal{R}$. By construction, this is possible only if $u = u_{cd}$ and $v = v_{cd}$. and hence $(c, d) \in \mathcal{I}$. Thus, $\mathcal{D}, \mathcal{I}, \sigma \models Q(x, y)$.

The case of $\neg$ and $\wedge$ is standard.

For $\exists x\ \psi$, we have $\mathcal{D}, \mathcal{I}, \sigma \models \exists x\ \psi$ iff there is some $d \in \mathcal{D}$ such that $\mathcal{D}, \mathcal{I}, \sigma_{[x \mapsto d]} \models \psi$ iff (by induction) $\mathcal{M}, r, \sigma_{[x \mapsto d]} \models \mathsf{Tr}_1(\psi)$ iff $\mathcal{M}, r, \sigma \models \exists x\ \mathsf{Tr}_1(\psi)$.

Since $\mathcal{D}, \mathcal{I} \models \varphi$, it follows from the claim that $M, r \models \mathsf{Tr}_1(\varphi)$.

($\Leftarrow$) Suppose $\mathsf{Tr}_1(\varphi)$ is satisfiable. Let $\mathcal{M}$ be the TML model and $r \in \mathcal{W}$ such that $\mathcal{M}, r \models \mathsf{Tr}_1(\varphi)$. We need to show that φ is satisfiable.

Define $\mathcal{D}' = \delta(r)$ and $\mathcal{I}' = \{(c, d) \mid \mathcal{M}, r, [x \mapsto c, y \mapsto d] \models \Diamond_x \Diamond_y \top\}$. It is enough to prove that $\mathcal{D}', \mathcal{I}' \models \varphi$. Again, we set up an induction to prove this.

Claim. For all $\psi \in \mathsf{SF}(\varphi)$ and for all interpretation $\sigma : \mathcal{V} \mapsto \mathcal{D}'$ we have $\mathcal{M}, r, \sigma \models \mathsf{Tr}_1(\psi)$ iff $\mathcal{D}', \mathcal{I}', \sigma \models \psi$.

The proof is by induction on the structure of ψ.

In the base case, we have $Q(x, y)$ and $\mathsf{Tr}_1(Q(x, y)) = \Diamond_x \Diamond_y \top$. Suppose we have $\mathcal{M}, r, \sigma \models \Diamond_x \Diamond_y \top$ then $(\sigma(x), \sigma(y)) \in \mathcal{I}'$ which implies $\mathcal{D}', \mathcal{I}', \sigma \models Q(x, y)$. Conversely, if $\mathcal{D}', \mathcal{I}', \sigma \models Q(x, y)$ then $(\sigma(x), \sigma(y)) \in \mathcal{I}'$. Hence $\mathcal{M}, r, \sigma \models \Diamond_x \Diamond_y \top$.

The case of $\neg$ and $\wedge$ is standard. For $\exists x\ \psi$, we have $\mathcal{M}, r, \sigma \models \exists x\ \mathsf{Tr}_1(\psi)$ iff there is some $d \in \delta(r)$ such that $\mathcal{M}, r, \sigma_{[x \mapsto d]} \models \mathsf{Tr}_1(\psi)$ iff (by ind.) $\mathcal{D}', \mathcal{I}', \sigma_{[x \mapsto d]} \models \psi$ iff $\mathcal{D}', \mathcal{I}', \sigma \models \exists x\ \psi$.

$\square$

Corollary 4.3. *Satisfiability problem for $\mathsf{PTML}^\top$ (restricted to formulas of modal depth 2) is undecidable over both constant and increasing domain models.*

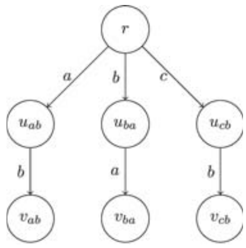


Figure 4.1: Model translation corresponding to the $\mathsf{FO}(Q)$ structure $(\mathcal{D}, \mathcal{I})$ where $\mathcal{D} = \{a, b, c\}$ and $\mathcal{I} = \{(a, b),\ (b, a),\ (c, b)\}$.

4.2 Recursive inseparability for pure-PTML$_=$

In the presence of equality, note that $\top$ can be encoded as $\forall x\ x = x$. Hence, from Theorem 4.2, the satisfiability problem for pure-PTML$_=$ is undecidable. For this fragment, we prove a stronger *Trakhtenbrot* type theorem of recursive inseparability.

Definition 4.4 (Recursive inseparability)**.** *Let X, Y be disjoint sets. We say that X and Y are recursively inseparable if there is no Z such that $Z \subseteq X$ and $Z \cap Y = \emptyset$ where checking for membership in Z is decidable.*

When we say *Trakhtenbrot* theorem holds for a logic, it means that the set of all formulas of the logic can be partitioned into FinSat, UnSat and InfAx such that these sets are mutually recursively inseparable, where FinSat is the set of all formulas that are satisfiable in some finite model, UnSat is the set of all formulas that are unsatisfiable and InfAx is the set of all formulas that are satisfiable in some infinite model but not in any finite model. For instance, FO$_=$ restricted to unary and binary predicates $(\mathcal{P}^1, \mathcal{P}^2)$ satisfies this property. Now we prove this for pure-PTML$_=$

Note that for pure-PTML$_=$, finite models are those models $\mathcal{M}$ where $\mathcal{W}^{\mathcal{M}}$ and $\mathcal{D}^{\mathcal{M}}$ both are finite. The following theorem states that with respect to finite satisfiability (for full PTML$_=$), it suffices to restrict $\mathcal{D}$ to be a finite set without any restriction on the size of $\mathcal{W}$.

Theorem 4.5 (Finite satisfiability for PTML$_=$)**.** *Let φ be any PTML$_=$ sentence. The following are equivalent.*

1. *φ is satisfiable in some model $\mathcal{M}$ where both $\mathcal{W}$ and $\mathcal{D}$ are finite.*

2. *φ is satisfiable in some model $\mathcal{M}$ where $\mathcal{D}$ is finite.*

Proof. It suffices to prove (2) implies (1). Let $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ be a model with finite $\mathcal{D}$ and $r \in \mathcal{W}$ such that $\mathcal{M}, r \models \varphi$. Let size of $\mathcal{D}$ be m.

Define an equivalence relation on $\mathcal{W}$ where for all $u, v \in \mathcal{W}$ we have $u \simeq v$ iff the following hold:

- For every proposition p occurring in φ, we have $\mathcal{M}, u \models p$ iff $\mathcal{M}, v \models p$.
- $\delta(u) = \delta(v)$.
- for all $\psi \in \mathsf{SF}(\varphi)$ such that $\mathsf{FV}(\psi) \subseteq \{x_1, \ldots x_n\}$, for all $d_1, \ldots d_n \in \delta(u)$, $\mathcal{M}, u \models \psi(d_1, \ldots, d_n)$ iff $\mathcal{M}, v \models \psi(d_1, \ldots, d_n)$.

Define $[u] = \{v \mid u \simeq v\}$ and $\mathcal{W}' = \{[u] \mid u \in \mathcal{W}\}$. Let $|\varphi| = l$ which implies $|\mathsf{SF}(\varphi)| \leq l$. Since $|\mathcal{D}| = m$, we have $|\mathcal{W}'| \leq 2^l \cdot 2^m \cdot (2^m \cdot 2^l) = 2^{O(m+l)}$. Define the model $\mathcal{N} = (\mathcal{W}', \mathcal{D}, \delta', \mathcal{R}', \rho')$ where $\delta'([w]) = \delta(w)$ for all $[w] \in \mathcal{W}'$ and $\mathcal{R}' = \{([u], d, [v]) \mid$ there is some $u' \in [u]$ and $v' \in [v]$ such that $(u', d, v') \in \mathcal{R}\}$. Define $\rho'([w]) = \{p \mid p$ occurs in φ and $\mathcal{M}, w \models p\}$.

Clearly $\mathcal{N}$ is well defined and has finite world set and finite agent set.

Claim. For all $\psi \in \mathsf{SF}(\varphi)$ and for all $w \in \mathcal{W}$, $\mathcal{M}, w, \sigma \models \psi$ iff $\mathcal{N}, [w], \sigma \models \psi$.

The proof is by induction on the structure of ψ. Note that for the case $x = y$ the claim follows since σ remains the same. The case of predicates, $\neg$ and $\wedge$ are standard.

For the case $\Diamond_x \psi$, suppose $\mathcal{M}, w, \sigma \models \Diamond_x \psi$, let $\sigma(x) = c$. By semantics, there is some $u \in \mathcal{W}$ such that $(w, c, u) \in \mathcal{R}$ and $\mathcal{M}, u, \sigma \models \psi$. By induction hypothesis, $\mathcal{N}, [u], \sigma \models \psi$ and by construction $([w], c, [u]) \in \mathcal{R}'$. Hence, $\mathcal{N}, [w], \sigma \models \Diamond_x \psi$.

On the other hand if $\mathcal{N}, [w], \hat{\sigma} \models \Diamond_x \psi$, let $\sigma(x) = c$. Let $w' \in [w]$, we need to prove that $\mathcal{M}, w', \sigma \models \Diamond_x \psi$. First note that σ is relevant at w' and $c \in \delta(w')$ since $\delta'([w]) = \delta(w) = \delta(w')$. Now, since $\mathcal{N}, [w], \hat{\sigma} \models \Diamond_x \psi$, by semantics, there is some $([w], c, [u]) \in \mathcal{R}'$ such that $\mathcal{N}, [u], \sigma \models \psi$ and by construction, there is some $w_1 \in [w]$ and $u_1 \in [u]$ such that $(w_1, c, u_1) \in \mathcal{R}$ and by induction hypothesis $\mathcal{M}, u_1, \sigma \models \psi$. Now since $(w_1, c, u_1) \in \mathcal{R}$ we have $\mathcal{M}, w_1, \sigma \models \Diamond_x \psi$. But then $w' \simeq w_1$ and hence $\mathcal{M}, w', \sigma \models \Diamond_x \psi$.

For the case $\exists x\ \psi$, if $\mathcal{M}, w, \sigma \models \exists x\ \psi$ then there is some $d \in \delta(w)$ such that $\mathcal{M}, w, \sigma_{[x \mapsto d]} \models \psi$. By induction hypothesis, $\mathcal{N}, [w], \sigma_{[x \mapsto d]} \models \psi$ and this implies $\mathcal{N}, [w], \sigma \models \exists x\ \psi$.

If $\mathcal{N}, [w], \sigma \models \exists x\ \psi$ let $w' \in [w]$. We need to prove that $\mathcal{M}, w', \sigma \models \exists x\ \psi$. Again, note that σ is relevant at w'. Now, by semantics, there is some $d \in \delta'([w])$ such that $\mathcal{N}, [w], \sigma_{[x \mapsto d]} \models \psi$ and by induction hypothesis $\mathcal{M}, w, \sigma_{[x \mapsto d]} \models \psi$. But then $w' \simeq w$ and hence $\mathcal{M}, w', \sigma_{[x \mapsto d]} \models \psi$ which implies $\mathcal{M}, w', \sigma \models \exists x\ \psi$.

□

Thus for $\mathsf{PTML}_=$ (and $\mathsf{pure\text{-}PTML}_=$ in particular) it is enough to define finite satisfiability by restricting $\mathcal{D}$ to be finite without worrying about the size of $\mathcal{W}$. This helps in simplifying the proofs.

Definition 4.6 ($\mathsf{FinSat}, \mathsf{UnSat}, \mathsf{InfAx}$)**.** *For any formula $\varphi \in \mathsf{pure\text{-}PTML}_=$ we say that φ is* finitely satisfiable *if there is some model $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \mathcal{R}, \delta)$ with finite $\mathcal{D}$ and some $w \in \mathcal{W}$ and interpretation $\sigma : Var \mapsto D$ relevant at w such that $\mathcal{M}, w, \sigma \models \varphi$. Further, φ is* unsatisfiable *if for all $\mathcal{M}$ and $w \in \mathcal{W}$ and all σ relevant at w, we have $\mathcal{M}, w, \sigma \not\models \varphi$.*

Let $\mathsf{FinSat} = \{\varphi \in \mathsf{pure\text{-}PTML}_= \mid \varphi \text{ is finitely satisfiable}\}$; $\mathsf{UnSat} = \{\varphi \in \mathsf{pure\text{-}PTML}_= \mid \varphi \text{ is unsatisfiable}\}$ and $\mathsf{InfAx} = \mathsf{pure\text{-}PTML}_= \setminus (\mathsf{FinSat} \cup \mathsf{UnSat})$.

Note that InfAx denotes the set of all formulas that are satisfiable in some model with infinite $\mathcal{D}$ but unsatisfiable in all models with finite $\mathcal{D}$. Also, $\mathsf{FinSat}, \mathsf{UnSat}$ and InfAx forms a disjoint partition over $\mathsf{pure\text{-}PTML}_=$.

We now prove that $\mathsf{FinSat}, \mathsf{UnSat}$ and InfAx are mutually recursively inseparable. To prove this, we use a reduction from tiling problem.

Tiling problem Let $\mathcal{C}$ be a set of colours. A tile type $\mathsf{t} \in \mathcal{C}^4$ is a tuple of 4 colours. For any tile type t we shall call the 4 colours *up, down, right* and *left* edge colours respectively, and write $\mathsf{t} = (u_\mathsf{t}, d_\mathsf{t}, r_\mathsf{t}, \ell_\mathsf{t})$. A *tiling instance* is given by $\mathsf{T} = (X, \mathsf{t}_0)$ where X is a finite set of tile types and $\mathsf{t}_0 \in X$. The tiling instance $\mathsf{T} = (X, \mathsf{t}_0)$ *has a proper tiling* if there is a tiling function $S : (\mathbb{N} \times \mathbb{N}) \mapsto X$ such that $S(0,0) = \mathsf{t}_0$ and for all $i, j \in \mathbb{N}$, if $S(i,j) = \mathsf{t}$ and $S(i+1,j) = \mathsf{t}'$ then $r_\mathsf{t} = \ell_{\mathsf{t}'}$ and similarly, if $S(i,j) = \mathsf{t}$ and $S(i,j+1) = \mathsf{t}'$ then $u_\mathsf{t} = d_{\mathsf{t}'}$. We say that S is a *periodic tiling function* if there exists some $n \in \mathbb{N}$ such that for all $i, j \in \mathbb{N}$, $S(i,j) = S(i+n,j) = S(i,j+n)$. A tiling function S is *aperiodic* if it is not periodic. Figure 4.2 illustrates a tiling instance.

Let $\mathsf{PT} = \{\mathsf{T} = (X, t_0) \mid \mathsf{T}$ has some periodic tiling$\}$ and similarly the set $\mathsf{NT} = \{\mathsf{T} = (X, t_0) \mid \mathsf{T}$ has no proper tiling$\}$ and $\mathsf{AT} = \{\mathsf{T} = (X, t_0) \mid \mathsf{T}$ has only aperiodic tilings$\}$. We use the following result about tiling problem.

Theorem 4.7 (Gurevich and Koryakov[GK72]). *The sets PT, AT and NT are mutually recursively inseparable.*

Now we show that the partitioning of $\mathsf{pure\text{-}PTML}_=$ formulas as $\mathsf{FinSat}, \mathsf{UnSat}$ and InfAx are mutually recursively inseparable, by showing a correspondence with PT, NT and AT respectively.

The key idea is to have natural numbers as the agent set. Then a grid point (i,j) can be identified by a path of length 2 where first edge is labelled i and second by j.

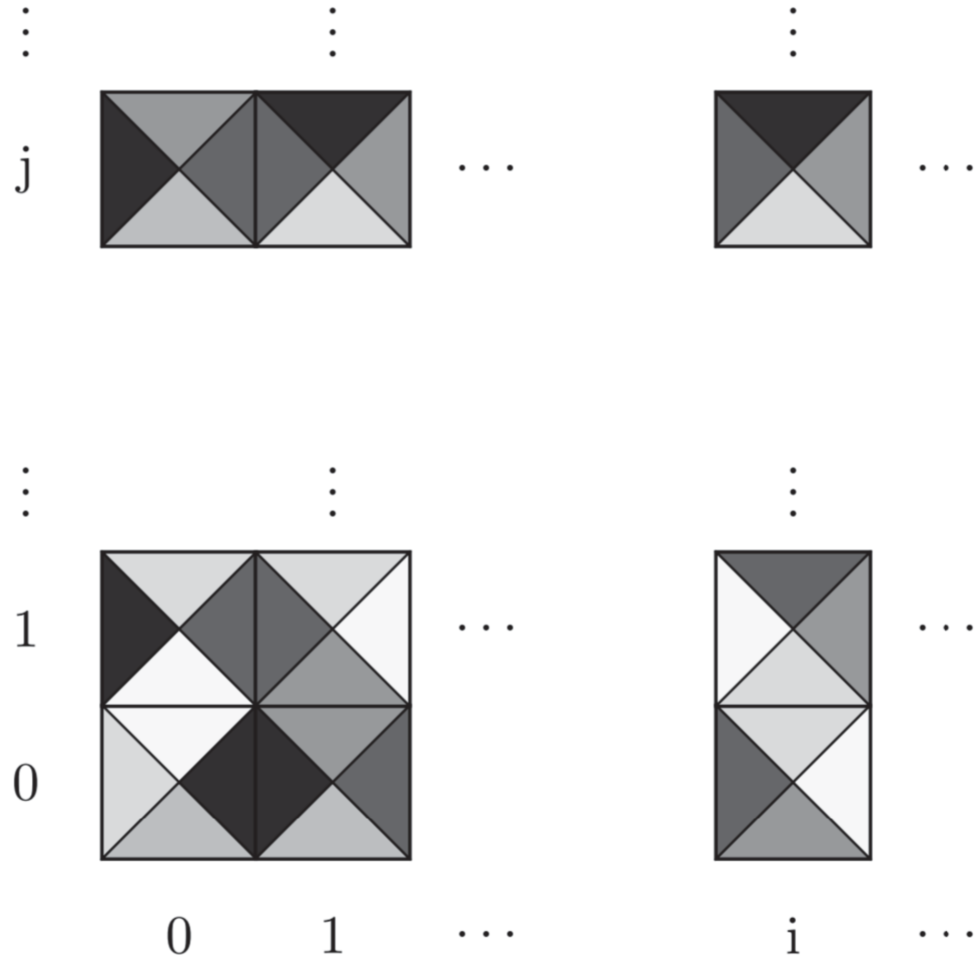


Figure 4.2: An instance of tiling.

Thus, before encoding the tiling instance, we need to have an initial segment of $\mathbb{N}$ either finite or infinite (depending upon periodic or aperiodic tiling) in the agent set $\mathcal{D}$ on which the grid is built. To achieve this, we define a finite set of $\mathsf{pure\text{-}PTML}_=$ formulas $\mathcal{O}$ that induces a discrete and total linear order (with min) over the agent set at any world in any model where the set of formulas $\mathcal{O}$ is true. To achieve this, we encode $i < j$ as $\Diamond_i \Diamond_j \top$. Recall that $\top$ can be encoded as $\forall x\ x = x$.

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}	:=	$\forall x\forall y\ \big((x \neq y) \rightarrow (\Diamond_x\Diamond_y\top \vee \Diamond_y\Diamond_x\top)\big)$	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 max or has a successor (discrete)
φ_{trans}	:=	$\forall x\forall y\forall z\ (\Diamond_x\Diamond_y\top \wedge \Diamond_y\Diamond_z\top) \rightarrow (\Diamond_x\Diamond_z\top)$	for all c, d, e if $c < d$ and $d < e$ then $c < e$
where,			
$zero(x)$	:=	$\forall y\ \neg\Diamond_y\Diamond_x\top$	for all c, $c \not< \sigma(x)$
$last(x)$	:=	$\forall y\ \neg\Diamond_x\Diamond_y\top$	for all c, $\sigma(x) \not< c$
$succ(x, y)$	:=	$\Diamond_x\Diamond_y\top \wedge \forall z\ \big((\Diamond_z\Diamond_y\top) \rightarrow (x = z \vee \Diamond_z\Diamond_x\top)\big)$	$\sigma(x) < \sigma(y)$ and for all c if $c < \sigma(y)$ then $\sigma(x) = c$ or $c < \sigma(x)$

Define $\mathcal{O} = \{\varphi_0, \varphi_{ir}, \varphi_{tot}, \varphi_{dis}, \varphi_{trans}\}$ and $\hat{\mathcal{O}} = \bigwedge_{\varphi \in \mathcal{O}} \varphi$. The following lemma states that in any model $\mathcal{M}$ at any world $w \in \mathcal{W}$ if $\mathcal{M}, w \models \hat{\mathcal{O}}$ then there is a way to induce a discrete linear order over $\delta(w)$.

Lemma 4.8. *The following statements hold for the formula $\hat{\mathcal{O}}$:*

1. *For every $\mathbb{N}' \subseteq \mathbb{N}$ (either finite or infinite) which is an initial segment of $\mathbb{N}$, there is some $\mathcal{M} = (\mathcal{W}, \mathbb{N}', \delta, \mathcal{R})$ and $r \in \mathcal{W}$ such that $\delta(r) = \mathbb{N}'$ and $\mathcal{M}, r \models \hat{\mathcal{O}}$.*

2. *For any model $\mathcal{M}$, suppose $\mathcal{M}, r \models \hat{\mathcal{O}}$ then let $\mathbb{N}' = [0, \ldots, n-1]$ if $\delta(r)$ is a finite set of size n and $\mathbb{N}' = \mathbb{N}$ otherwise. There exists a function $f : \mathbb{N}' \rightarrow \delta(r)$ such that for all $i, j \in \mathbb{N}'$ if $i < j$ then $\mathcal{M}, w \models \Diamond_{f(i)}\Diamond_{f(j)}\top$.*

Proof. To prove (1), pick any initial segment $\mathbb{N}'$ (either finite of infinite).

We give a model where $\hat{\mathcal{O}}$ is satisfied. Define the model $\mathcal{M} = (\mathcal{W}, \mathbb{N}', \delta, \mathcal{R})$ where

- $\mathcal{W} = \{r\} \cup \{u_i \mid i \in \mathbb{N}'\} \cup \{v_{ij} \mid i, j \in \mathbb{N}' \text{ and } i < j\}$.
- for all $w \in W$ define $\delta(w) = \mathbb{N}'$.
- $\mathcal{R} = \{(r,\ i,\ u_i) \mid i \in \mathbb{N}'\} \cup \{(u_i,\ j,\ v_{ij}) \mid i, j \in \mathbb{N}' \text{ and } i < j\}$.

It can be easily verified that $\mathcal{M}, r \models \hat{\mathcal{O}}$. Figure 4.3 describes this model for a finite $\mathbb{N}'$ and the model described in figure 4.4 satisfies $\hat{\mathcal{O}}$ when $\mathbb{N}' = \mathbb{N}$.

To prove (2), let $\mathcal{M}, r \models \hat{\mathcal{O}}$. Now we have two cases to consider depending on whether $\delta(r)$ is finite or infinite.

The first case is when $\delta(r)$ is finite. Let $|\delta(r)| = n$. Now, we construct a sequence of partial functions $f^0, f^1, \ldots, f^{n-1}$ where each $f^i : [0 \cdots i] \mapsto \delta(r)$ such that for all i, the following holds:

1. f^i is an extension of f^{i-1}.
2. Suppose $f^i(i) = c$ then for all $b \in \delta(r)$ if $\mathcal{M}, r \models \Diamond_b \Diamond_c \top$ then $b \in image(f^i)$.
3. For all $j, k \leq i$ if $j < k$ then $\mathcal{M}, r \models \Diamond_{f^i(j)} \Diamond_{f^i(k)} \top$.

Hence by condition (3), f^{n-1} is the required function. The construction of such a sequence f^i is by induction on i.

In the base case to construct f^0, by φ_0 there is some agent (say $a_0 \in \delta(r)$) such that $\mathcal{M}, r \models zero(a_0)$. Define $f^0(0) = a_0$. Condition (1) does not apply. Since $\mathcal{M}, r \models \forall y\ \neg \Diamond_y \Diamond_{a_0} \top$, condition (2) is vacuously true. Also, by φ_{ir}, we have $M, r \models \neg \Diamond_{a_0} \Diamond_{a_0} \top$ and hence condition (3) holds.

Now inductively assume that we have f^i for some $i < n-1$. Let $f^i(j) = a_j$ for all $j \leq i$. In particular, we have $f^i(i) = a_i$. Now by φ_{dis} either $\mathcal{M}, r \models \mathit{last}(a_i)$ or $\mathcal{M}, r \models \exists y\ \mathit{succ}(a_i, y)$. Since $i < n-1$, we claim that $\mathcal{M}, r \not\models \mathit{last}(a_i)$. Suppose not, then $\mathcal{M}, r \models \forall y\ \neg\Diamond_{a_i}\Diamond_y\top$. Now since $i < n-1$, there is at least one agent $d \in \delta(r)$ such that $d \notin \mathit{image}(f^i)$. This is because $f^i : [0 \cdots i] \to \delta(r)$ and hence the image size is at most $i + 1 \leq n - 1 < n = |\delta(r)|$. Now by assumption, $\mathcal{M}, r \models \neg\Diamond_{a_i}\Diamond_d\top$ and by φ_{tot}, $\mathcal{M}, r \models \Diamond_d\Diamond_{a_i}\top$. But then, by induction hypothesis condition (2), $d \in \mathit{image}(f^i)$ which is a contradiction.

Hence it has to be the case that $\mathcal{M}, r \models \exists y\ \mathit{succ}(a_i, y)$. Let a_{i+1} be the witness and hence we have $\mathcal{M}, r \models \Diamond_{a_i}\Diamond_{a_{i+1}}\top$. Define $f^{i+1}(j) = a_j$ for all $j \leq i$ and $f^{i+1}(i+1) = a_{i+1}$. Clearly f^{i+1} is an extension of f^i. To show condition (2), suppose $\mathcal{M}, r \models \Diamond_b\Diamond_{a_{i+1}}\top$ for some $b \in \delta(r)$. Since $\mathcal{M}, r \models \mathit{succ}(a_i, a_{i+1})$, we have $\mathcal{M}, r \models (b = a_i \vee \Diamond_b\Diamond a_i\top)$. By induction hypothesis condition (2), $b \in \mathit{image}(f^i)$ and hence $b \in \mathit{image}(f^{i+1})$. Finally to verify that condition (3) holds, let $j, k \in \{0, 1, \cdots i, i+1\}$. We consider all possible cases of j, k:

- For $j, k \leq i$: Since f^{i+1} is an extension of f^i the claim holds by ind. hyp.
- For $j < i$ and $k = i + 1$ by induction hypothesis we have $\mathcal{M}, r \models \Diamond_{a_j}\Diamond_{a_i}\top$. Also, since a_{i+1} is the successor of a_i we have $\mathcal{M}, r \models \Diamond_{a_i}\Diamond_{a_{i+1}}\top$. Thus, by φ_{trans} we have $\mathcal{M}, r \models \Diamond_{a_j}\Diamond_{a_{i+1}}\top$.
- When $j = i$ and $k = i + 1$ since a_{i+1} is the successor of a_i, $\mathcal{M}, r \models \Diamond_{a_i}\Diamond_{a_{i+1}}\top$
- Finally for $j = k = i + 1$, by φ_{ir} we have $\mathcal{M}, r \models \neg\Diamond_{a_{i+1}}\Diamond_{a_{i+1}}\top$.

For the case when $\delta(r)$ is infinite, as in the previous case, we construct a sequence of functions $f^0, f^1, \ldots$ inductively where each f^i again satisfies all the 3 properties stated.

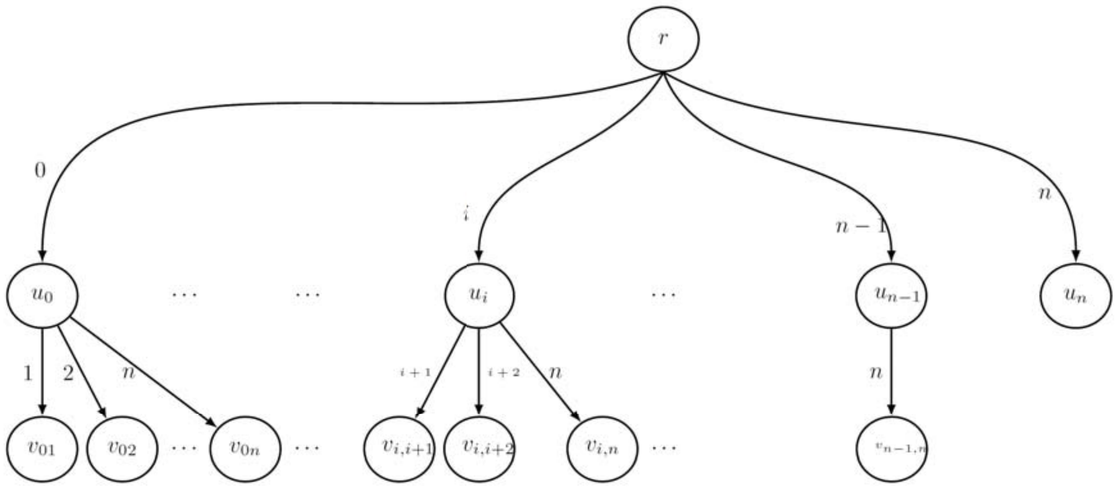


Figure 4.3: A model for $\mathcal{O}$ when $\mathbb{N}' = [0, 1 \ldots, n]$ is finite with $\delta(w) = \mathbb{N}'$ for all w. Observe that $i < j$ iff $M, r \models \Diamond_i \Diamond_j \top$.

Notice that we have at any step of construction f^i we have $\mathcal{M}, w \models \neg last(f^i(i))$ (otherwise $\mathcal{M}, r \models \forall y\ \neg \Diamond_{f^i(i)} \Diamond_y \top$ we can again hit a contradiction as argued in the previous case). Finally, the required function in this case is given by $f = \bigcup_{i \in \mathbb{N}} f^i$.

□

Therefore, whenever $\mathcal{M}, r \models \hat{\mathcal{O}}$, without loss of generality we assume that there is some initial fragment $\mathbb{N}'$ of $\mathbb{N}$ with $|\mathbb{N}'| = |\delta(r)|$ such that $\mathbb{N}' \subseteq \delta(r)$ and for all $i, j \in \mathbb{N}'$ if $i < j$ then $\mathcal{M}, w \models \Diamond_i \Diamond_j \top$.

For any finite set of tile types $X = \{\mathsf{t}_0, \ldots, \mathsf{t}_m\}$, we encode every tile t_i as a path of length i, given by

$$p_i := \bigwedge_{j \leq i} \big((\forall z \Box_z)^j (\exists z\ \Diamond_z \top) \big)\ \wedge (\forall z \Box_z)^{i+1} (\forall z \Box_z \bot)$$

where $(\forall z \Box_z)^i(\varphi) = \forall z \Box_z \big((\forall z \Box_z)^{i-1}(\varphi) \big)$ and $(\forall z \Box_z)^0(\varphi) = \varphi$.

Given an input tiling instance $\mathsf{T} = (X, \mathsf{t}_0)$ define a finite set of formulas Γ_{T} which corresponds to the tiling instance T. First we have $\mathcal{O} \subseteq \Gamma_{\mathsf{T}}$ so that we have natural numbers as agents.

Since the first 2 modal depths are used up to enforce the order, all tiling information is encoded at modal depth level > 2. Thus, the grid point (x, y) having tile t_i is encoded as $\forall z_1 \forall z_2\ \Box_{z_1}\Box_{z_2}(\Box_x \Box_y p_i)$. Using $\Box_{z_1}\Box_{z_2}$ as suffix, we can guarantee that the tiling information can be extracted by looking at any of the successors at depth 2, depending on the length of the paths reachable thereafter. We use the definition of $succ(x, y)$ to check for the horizontal and vertical constraints. Thus, for a given tiling instance the tiling constraint formulas $\mathsf{T} = (X, \mathsf{t}_0)$ are given by:

φ_{tile}	$:=$	$\forall z_1 \forall z_2 \forall x \forall y\ \Box_{z_1}\Box_{z_2}\big((\Diamond_x \Diamond_y \top) \wedge \Box_x \Box_y (\bigvee_{\mathsf{t}_i \in X} p_i)\big)$	every grid point has a unique tile
φ_{init}	$:=$	$\forall z_1 \forall z_2 \forall x\ \big(zero(x) \to (\Box_{z_1}\Box_{z_2}\Box_x \Box_x p_0)\big)$	$(0, 0)$ has tile t_0
φ_{hor}	$:=$	$\forall z_1 \forall z_2 \forall x \forall y \forall z$ $\big(((last(x) \wedge zero(y)) \vee succ(x, y)) \to$ $\Box_{z_1}\Box_{z_2}\ (\bigvee_{r_{\mathsf{t}_i} = \ell_{\mathsf{t}_j}} (\Box_x \Box_z (p_i) \wedge \Box_y \Box_z (p_j))\big)$	colours match across horizontal successors
φ_{ver}	$:=$	$\forall z_1 \forall z_2 \forall x \forall y \forall z$ $\big(((last(x) \wedge zero(y)) \vee succ(x, y)) \to$ $\Box_{z_1}\Box_{z_2}\ (\bigvee_{u_{\mathsf{t}_i} = d_{\mathsf{t}_j}} (\Box_z \Box_x (p_i) \wedge \Box_z \Box_y (p_j))\big)$	colours match across vertical successors

Define $\Gamma_{\mathsf{T}} = \mathcal{O}\ \cup\ \{\varphi_{tile}, \varphi_{init}, \varphi_{hor}, \varphi_{ver}\}$.

Note that, in any typical tiling reduction, to say every grid point has a unique tile, we use the formula of the form $\bigvee_i (p_i \wedge \bigwedge_{j \neq i} \neg p_j)$. But in our case, it is not needed since in any model $\mathcal{M}$ and any $w \in \mathcal{W}$ if $\mathcal{M}, w \models p_i$ then it means every path starting from w is exactly of length $i + 1$ and this implies that for all $j \neq i$ we have $\mathcal{M}, w \not\models p_j$.

Theorem 4.9. *For any given tiling instance* $\mathsf{T} = (X, \mathsf{t}_0)$*, let* $\varphi_{\mathsf{T}} ::= \bigwedge_{\psi \in \Gamma_{\mathsf{T}}} \psi$*. Then the following holds:*

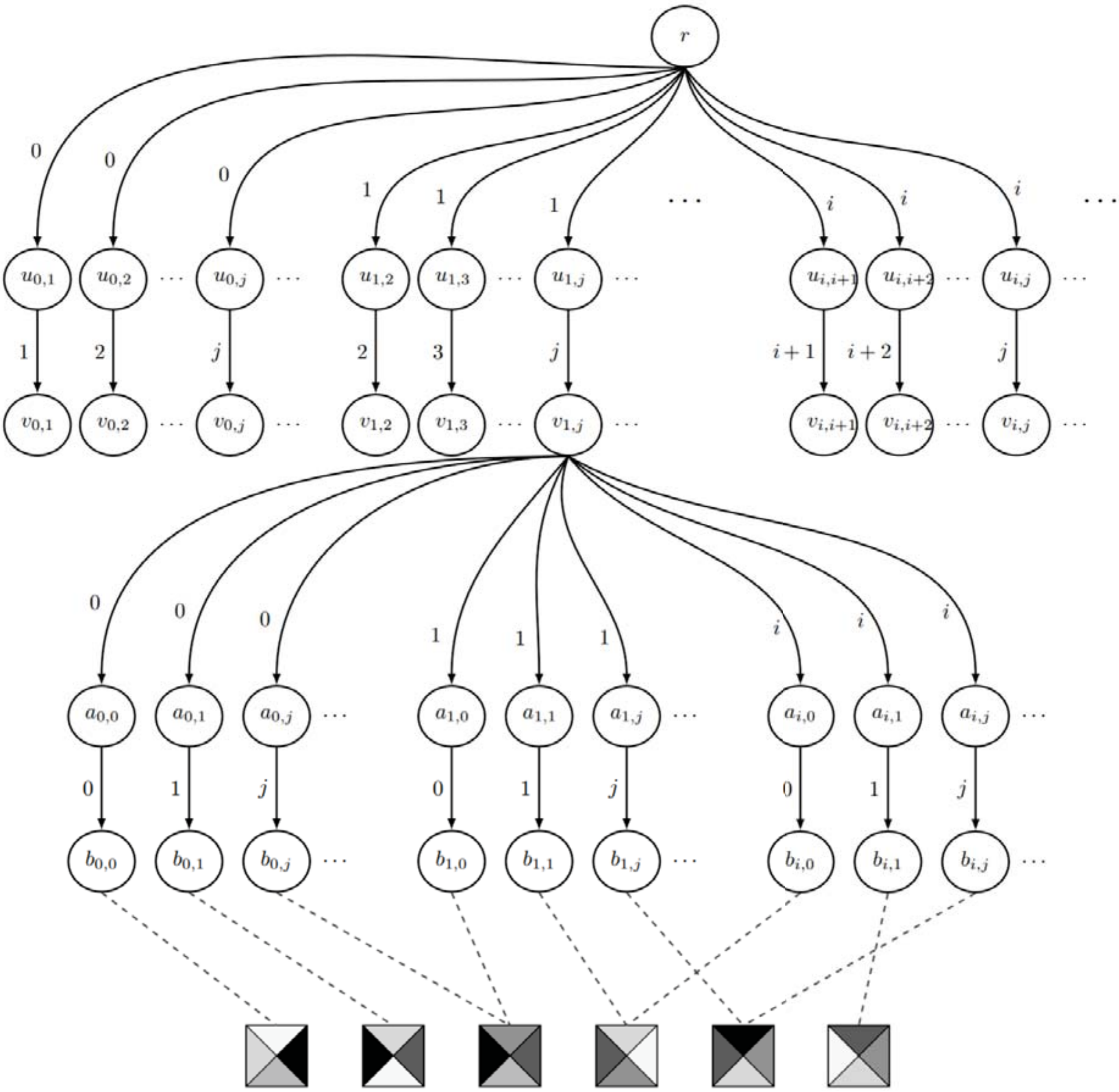


Figure 4.4: Model corresponding to the aperoidic tiling described in Fig 4.2. The subtree rooted at $v_{1,j}$ is present for every vertex $v_{k,k'}$ at level 2. The dotted line indicates an edge labelled by 0 and every tile $\mathtt{t}_i$ in the figure is a path of length i.

1. $\mathsf{T} \in \mathit{PT}$ *iff* $\varphi_{\mathsf{T}} \in \mathit{FinSat}$

2. $\mathsf{T} \in \mathit{AT}$ *iff* $\varphi_{\mathsf{T}} \in \mathit{InfAx}$

3. $\mathsf{T} \in \mathit{NT}$ *iff* $\varphi_{\mathsf{T}} \in \mathit{UnSat}$

Proof. For (1), ($\Rightarrow$) direction, suppose $\mathsf{T} = (X, \mathsf{t}_0)$ has a periodic tiling then let $S : \mathbb{N} \times \mathbb{N} \to X$ be the tiling and $n \in \mathbb{N}$ be the period such that for all $i, j \in \mathbb{N}$ we have $S(i, j) = S(i + n, j) = S(i, j + n)$.

Now we need to construct a model with finite $\mathcal{D}$ for φ_{T}. For this, we take $\mathcal{D} = [0, 1, \ldots, n-1]$ (call it $\mathbb{N}'$). Since tile types are encoded as paths, for every $\mathsf{t}_i \in X$ let $L_i^{\mathsf{t}} = (\mathcal{W}_i^{\mathsf{t}}, \mathcal{R}_i^{\mathsf{t}})$ be a path of length i where each edge is labelled by 0. Formally, $\mathcal{W}_i^{\mathsf{t}} = \{w_i^j \mid j \leq i\}$ and $\mathcal{R}_i^{\mathsf{t}} = \{(w_i^j, 0, w_i^{j+1}) \mid j < i\}$. Thus L_i^{t} is of the form $w_i^0 \xrightarrow{0} w_i^1 \xrightarrow{0} \cdots \xrightarrow{0} w_i^i$. Let $L^{\mathsf{T}} = \{L_i^{\mathsf{t}} \mid \mathsf{t}_i \in X\}$ and $\mathcal{W}^{\mathsf{T}} = \bigcup_{\mathsf{t}_i \in X} \mathcal{W}_i^{\mathsf{t}}$ and $\mathcal{R}^{\mathsf{T}} = \bigcup_{\mathsf{t}_i \in X} \mathcal{R}_i^{\mathsf{t}}$. Define $\mathcal{M} = (\mathcal{W}, \mathbb{N}', \delta, \mathcal{R})$ where:

- $\mathcal{W} = \{r\} \cup \{u_i \mid \in \mathbb{N}'\} \cup \{v_{ij} \mid i, j \in \mathbb{N}' \text{ and } i < j\} \cup \{a_i, b_{ij} \mid i, j \in \mathbb{N}'\} \cup \mathcal{W}^{\mathsf{T}}$.
- For all $w \in \mathcal{W}$ define $\delta(w) = \mathbb{N}'$.
- $\mathcal{R} = \{(r, i, u_i) \mid i \in \mathbb{N}'\} \cup \{(u_i, j, v_{ij}) \mid i, j \in \mathbb{N}' \text{ and } i < j\} \cup$
  $\{(v_{ij}, i, a_i) \mid i \in \mathbb{N}'\} \cup \{(a_i, j, b_{ij}) \mid i, j \in \mathbb{N}'\} \cup$
  $\{(b_{ij}, 0, w_k^0) \mid S(i, j) = k\} \cup \mathcal{R}^{\mathsf{T}}$.

Clearly $\mathcal{M}, w \models \hat{\mathcal{O}} \wedge \varphi_{init} \wedge \varphi_{tile}$. To verify that $\mathcal{M}, w \models \varphi_{hor}$ for any $c, d \in \mathbb{N}'$ suppose $\mathcal{M}, w \models succ(c, d)$ then $d = c + 1$. Now for any $e \in \mathbb{N}'$ let $S(c, e) = \mathsf{t}_i$ and $S(c + 1, e) = \mathsf{t}_j$ and since S is a proper tiling we have $r_{\mathsf{t}_i} = \ell_{\mathsf{t}_j}$ and hence $\mathcal{M}, w \models \forall z_1, \forall z_2 \forall z \Box_{z_1} \Box_{z_2} \big(\bigvee_{r_{\mathsf{t}_i} = \ell_{\mathsf{t}_j}} (\Box_c \Box_z(p_i) \wedge \Box_d \Box_z(p_j) \big)$. On the other hand if $\mathcal{M}, w \models zero(c) \wedge last(d)$ then $c = 0$ and $d = n-1$. For any $e \in \mathbb{N}'$ let $S(c, e) = \mathsf{t}_i$ and $S(0, e) = \mathsf{t}_j$. Since S is a proper tiling of period n, we have $S(n, e) = S(0, e)$ which implies $r_{\mathsf{t}_i} = \ell_{\mathsf{t}_j}$. Hence $\mathcal{M}, w \models \forall z_1, \forall z_2 \forall z \Box_{z_1} \Box_{z_2} \big(\bigvee_{r_{\mathsf{t}_i} = \ell_{\mathsf{t}_j}} (\Box_x \Box_z(p_i) \wedge \Box_y \Box_z(p_j) \big)$.

Similarly, $\mathcal{M}, w \models \varphi_{ver}$ can be verified.

($\Leftarrow$) Suppose $\varphi_{\mathsf{T}} \in \mathsf{FinSat}$. Then there is a model $\mathcal{M}$ with finite $\mathcal{D}$ such that $\mathcal{M}, r \models \varphi_{\mathsf{T}}$. By Lemma 4.8, *w.l.o.g* assume that $\delta(r) = \mathbb{N}' = [0, 1, \cdots n-1]$. Hence there is at least one path of length 2 from $r \xrightarrow{0} u \xrightarrow{1} v$. Define the tiling function $S : \mathbb{N} \times \mathbb{N} \to X$ where for all $j, k \in \mathbb{N}$, $S(j,k) = \mathsf{t}_i$ iff $\mathcal{M}, v \models \Diamond_{j'} \Diamond_{k'} p_i$ where $j' = j \mod (n)$ and $k' = k \mod (n)$.

Now by φ_{tile}, S is well-defined and total and by φ_{init}, $S(0,0) = \mathsf{t}_0$. Also by φ_{hor} if $S(i,j) = t$ and $S(i+1,j) = t'$ then $r_t = \ell_{t'}$ and similarly by φ_{ver}, if $S(i,j) = t$ and $S(i, j+1) = t'$ then $u_t = d_{t'}$. Finally, for all $j, k \in \mathbb{N}$ by construction, we have $S(j,k) = S(j+n, k) = S(j, k+n)$ and hence S is a proper periodic tiling of period length n.

To prove (2), ($\Rightarrow$) suppose $\mathsf{T} \in \mathsf{AT}$, then T has only aperiodic tiling. Let $S : \mathbb{N} \times \mathbb{N} \to X$ be some aperiodic tiling.

We need to show that $\varphi_{\mathsf{T}} \in \mathsf{InfAx}$. First note that $\varphi_{\mathsf{T}} \notin \mathsf{FinSat}$, otherwise by (1), $\mathsf{T} \in \mathsf{PT}$ which contradicts $\mathsf{T} \in \mathsf{AT}$. Hence it is sufficient to construct one model for φ_{T} with infinite $\mathcal{D}$. Again, let $L^{\mathsf{T}} = \{L_i^{\mathsf{t}} \mid \mathsf{t}_i \in X\}$ and $\mathcal{W}^{\mathsf{T}} = \bigcup_{\mathsf{t}_i \in X} \mathcal{W}_i^{\mathsf{t}}$ and $\mathcal{R}^{\mathsf{T}} = \bigcup_{\mathsf{t}_i \in X} \mathcal{R}_i^{\mathsf{t}}$ be as described in the previous case.

Now define $\mathcal{M} = (\mathcal{W}, \mathbb{N}, \delta, \mathcal{R})$ where

- $\mathcal{W} = \{r\} \cup \{u_i \mid \in \mathbb{N}\} \cup \{v_{ij} \mid i, j \in \mathbb{N} \text{ and } i < j\} \cup$
  $\{a_i, b_{ij} \mid i, j \in \mathbb{N}\} \cup \mathcal{W}^{\mathsf{T}}$.
- For all $w \in \mathcal{W}$ define $\delta(w) = \mathbb{N}$.
- $\mathcal{R} = \{(r, i, u_i) \mid i \in \mathbb{N}\} \cup \{(u_i, j, v_{ij}) \mid i, j \in \mathbb{N} \text{ and } i < j\} \cup$
  $\{(v_{ij}, i, a_i) \mid i \in \mathbb{N}'\} \cup \{(a_i, j, b_{ij} \mid i, j \in \mathbb{N}\} \cup \mathcal{R}^{\mathsf{T}} \cup$
  $\{(b_{ij}, 0, w_k^0) \mid S(i,j) = k\}$.

Note that the only difference from the previous case is that here $\mathcal{D} = \mathbb{N}$ (infinite) but in the models for periodic tiling the agent set $\mathcal{D}$ is finite whose size is the same as the period length.

Again, it can be verified that for all formulas $\psi \in \Gamma_{\mathsf{T}}$, $M, w \models \psi$. Figure 4.4 describes the corresponding model for the a periodic tiling instance described in Figure 4.2.

($\Leftarrow$) Suppose $\varphi_{\mathsf{T}} \in \mathsf{InfAx}$. Then there is a model $\mathcal{M}$ with infinite $\mathcal{D}$ such that $\mathcal{M}, r \models \varphi_{\mathsf{T}}$, but no models with finite $\mathcal{D}$. Now, if T has a periodic tiling, then $\mathsf{T} \in \mathsf{PT}$ then by (1) we have $\varphi_{\mathsf{T}} \in \mathsf{FinSat}$ which is a contradiction. Hence if there is any tiling of T, it has to be aperiodic. Thus it is enough to show that T has some tiling.

By Lemma 4.8, $w.l.o.g$ we assume that $\mathbb{N} \subseteq \delta(r)$. Also, there is at least one path of length 2 from $r \xrightarrow{0} u \xrightarrow{1} v$. Define the tiling function $S : \mathbb{N} \times \mathbb{N} \to X$ where for all $j, k \in \mathbb{N}$, $S(j,k) = \mathsf{t}_i$ iff $\mathcal{M}, v \models \Diamond_j \Diamond_k p_i$.

Now again, by φ_{tile}, S is well-defined and total and by φ_{init}, $S(0,0) = \mathsf{t}_0$. Also φ_{hor} and φ_{ver} ensure the horizontal and vertical colour constraints respectively.

Finally, for the case (3) let $\mathsf{T} \in \mathsf{NT}$. Suppose forward direction does not hold then $\mathsf{T} \in \mathsf{NT}$ and $\varphi_{\mathsf{T}} \notin \mathsf{UnSat}$. This implies $\varphi_{\mathsf{T}} \in (\mathsf{FinSat} \cup \mathsf{InfAx})$ and by (1), (2) we get $\mathsf{T} \in (\mathsf{PT} \cup \mathsf{AT})$ which contradicts $\mathsf{T} \in \mathsf{NT}$.

Similarly if $\varphi_{\mathsf{T}} \in \mathsf{UnSat}$ but $\mathsf{T} \notin \mathsf{NT}$ then $\mathsf{T} \in (\mathsf{PT} \cup \mathsf{AT})$ which will imply $\varphi_{\mathsf{T}} \in (\mathsf{FinSat} \cup \mathsf{InfAx})$. This is again contradiction to $\varphi_{\mathsf{T}} \in \mathsf{UnSat}$.

□

Corollary 4.10. *For $\mathsf{TML}^=$, the sets FinSat, InfAx and UnSat are mutually recursively inseparable.*

4.3 Discussion

Note that the equi-satisfiable translation from $\mathsf{FO}(Q)$ to $\mathsf{PTML}^{\top}$ with respect to satisfiability problem is in some sense canonical. As a consequence, many results that hold for $\mathsf{FO}(Q)$ can be lifted to $\mathsf{PTML}^{\top}$. For instance, the *finite satisfiability problem*[1] is undecidable. We also have other results like: 3-variable fragment of $\mathsf{PTML}^{\top}$, $\forall\exists\forall$ fragment of $\mathsf{PTML}^{\top}$ etc are undecidable which follow from undecidability results of $\mathsf{FO}(Q)$.

On the other hand, finite satisfiability problem for $\mathsf{PTML}^{\top}$ (in fact for full $\mathsf{TML}_{=}$) is recursively enumerable since we can enumerate all finite models and an algorithm can check if φ is satisfied in any of them, one by one. This procedure is guaranteed to halt and say *yes* for all formulas φ that is satisfiable in some finite model.

One cause for undecidability is that the quantifiers and modal indices act independently. By restricting them to occur in a certain from (bundling), we indeed get decidable fragments which we will discuss in the next chapter.

Also, note that all the undecidability discussed in this chapter goes through without the need for constants. However, when we consider the 2-variable fragment TML, where the constant free fragment is decidable, adding constants makes the fragment undecidable.

[1] which is to decide whether the given $\mathsf{PTML}^{\top}$ formula φ satisfiable in some finite model $\mathcal{M}$

Chapter 5

Decidable fragments

The undecidability results in the previous chapter motivate us to identify some decidable fragments. One natural question to ask is: if we consider some decidable fragment of first order logic, is the corresponding term modal logic decidable? Towards this, note that restricting the arity of predicates will not help since the satisfiability problem for $\mathsf{PTML}^{\top}$ is already undecidable. Another obvious candidate is to limit the number of variables used in the formula. Again, from Theorem 4.2, it follows that the 3-variable fragment of $\mathsf{PTML}^{\top}$ is undecidable.

Mortimer [Mor75] proved that the two variable fragment of FO (FO^2) is decidable. On the other hand, Grädel and Otto [GO99] proved that the satisfiability problem for many of the natural extensions of FO^2 (like transitive closure, lfp) are undecidable[1]. In contrast to these negative results, we will show that the 2-variable TML is yet another rare extension of FO^2 that still remains decidable.

For first order modal logic, Wolter and Zakharyaschev [WZ01] prove that the two variable fragment of FOML (without equality) is undecidable. In fact, Rybakov and Shkatov prove that the two variable fragment of FOML with countably many propositions and a single unary predicate is already undecidable [RS17].

[1]the only decidable extension of FO^2 they consider is that of the counting quantifiers.

Thus, proving the decidability of two variable fragment of TML clearly distinguishes term modal logic from first order modal logic.

5.1 Translating TML to FOML

Before going into the two variable fragment, we will first discuss what happens to the one variable fragment. Note that the satisfiable problem for one variable fragment of FO is NP-complete.

For first order modal logic, with a single agent (FOML), again the one variable fragment is decidable (result holds for multi-modal FOML as well). In fact, Wolter and Zakharyaschev [WZ01] prove a more general result for FOML that the monodic restriction built on any generic decidable fragment of FO (eg. unary predicates, guarded fragment, two variable fragment) is decidable[2]. An FOML formula α is said to be monodic if every subformula of the form $\Box\psi$ or $\Diamond\psi$ satisfies $|\mathsf{FV}(\psi)| \leq 1$ i.e, every modal subformula has at most 1 free variable. For instance, the formula $\forall x \exists y\ (\Box P(x) \to \Diamond\neg Q(y))$ is a monodic whereas $\forall x \exists y\ \Box(P(x) \wedge \neg Q(y))$ is not a monodic formula.

To define the analogous notion of monodicity for term modal logic, note that the modality itself will have a free variable as its index. Thus we can restrict the free variable of the subformula inside the scope of the modality to the variable appearing as the index itself. Thus a TML formula φ is monodic if every subformula of the form $\Delta_x\psi$ has $\mathsf{FV}(\psi) \subseteq \{x\}$ where $\Delta \in \{\Box, \Diamond\}$.

We give an equi-satisfiable translation of TML formulas to FOML formulas that preserves the monodicity property. Using this, we can identify some decidable fragments of TML by looking at the corresponding FOML fragments.

[2]Note that all 1-variable FOML formulas are monodic formulas and is built on 1-variable fragment of FO which is decidable.

Recall that we use the same set of predicates $\mathcal{P}$ as vocabulary and the same variable set $\mathcal{V}$ for both TML and FOML. Thus, when we consider FOML (with single agent), in terms of model description, the only technical difference between first order modal logic models and term modal logic models is in the accessibility relation (Def. 2.9, Def. 3.2 respectively). For first order modal logic, $\mathcal{R} \subseteq (\mathcal{W} \times \mathcal{W})$ whereas for term modal logic, the accessibility relation $\mathcal{R} \subseteq (\mathcal{W} \times \mathcal{D} \times \mathcal{W})$. In other words, TML structures can be thought of as modified FOML structures, obtained by interpreting domain set $\mathcal{D}$ of the FOML structure as the potential set of agents.

Note that, like most of the commonly used modal logics (including ML^n and FOML^n) TML also satisfies **tree model property**. i.e, for any TML formula φ, if $\mathcal{M}, r, \sigma \models \varphi$ then $\mathcal{M}^T, r, \sigma \models \varphi$ where $\mathcal{M}^T$ is the standard tree unravelling of $\mathcal{M}$ with r as the root. Further, we can also restrict the height of $\mathcal{M}^T$ to be at most the modal depth of φ. This claim is formally proved in the Chapter 6 (Theorem 6.9). For now we will assume that whenever a TML formula φ is satisfiable, it is satisfied in a tree model of height at most $\mathsf{md}(\varphi)$.

Suppose $\mathcal{M}$ is a TML model which is a rooted tree, we can simply ignore the edge labels on the accessibility relation and that gives us an FOML model induced on $\mathcal{M}$ (call it $\mathcal{N}$). Now since every non-root world has a unique incoming edge (and hence a unique agent as incoming edge label), the agent label of the incoming edge for a world can be encoded as a unary predicate. Formally, we can take a unary predicate E and define its valuation in $\mathcal{N}$ such that for all non-root $w \in \mathcal{W}$ we have $\mathcal{N}, w \models E(d)$ iff d is the incoming edge of w in $\mathcal{M}$. Figure 5.1 illustrates this translation.

Definition 5.1 (Embedding TML into FOML). *Given a TML formula φ, let $E \in \mathcal{P}$ be a new unary predicate not occurring in φ. The translation of φ into an FOML formula is inductively defined as follows:*

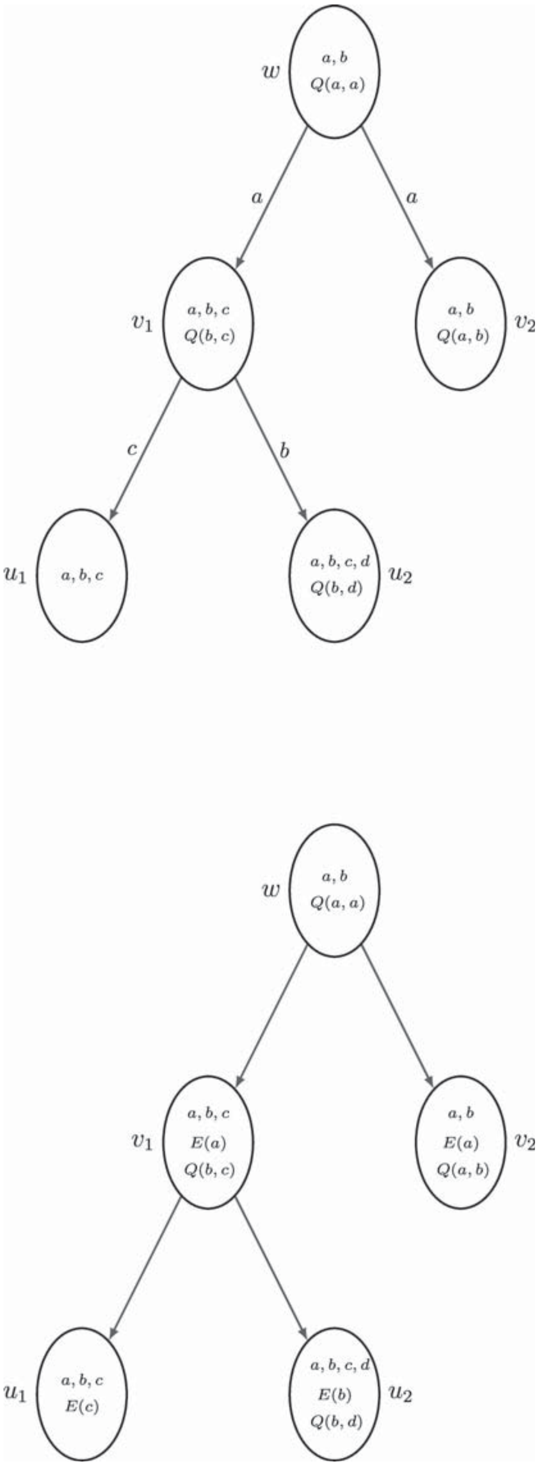


Figure 5.1: Illustration of a translation of term modal logic model to first order modal logic model. The edge information is encoded in the predicate E and all other valuations remain unchanged.

- $\mathit{Tr}_2(R(x_1, \ldots, x_n)) = R((x_1, \ldots, x_n)$

- $\mathit{Tr}_2(\neg\varphi) = \neg\ \mathit{Tr}_2(\varphi)$ *and* $\mathit{Tr}_2(\varphi \wedge \psi) = \mathit{Tr}_2(\varphi) \wedge \mathit{Tr}_2(\psi)$

- $\mathit{Tr}_2(\Diamond_x \varphi) = \Diamond\ (E(x) \wedge \mathit{Tr}_2(\varphi))$

- $\mathit{Tr}_2(\exists x\ \varphi) = \exists x\ \mathit{Tr}_2(\varphi)$

Note that the translation preserves modal depth, quantifier rank and the number of variables. Further, if we start with a monodic TML formula, the translation gives us a monodic FOML formula.

Theorem 5.2. *Let φ be any TML formula, φ is satisfiable in some TML model iff $\mathit{Tr}_2(\varphi)$ is satisfiable in some FOML model.*

Proof. ($\Rightarrow$) Suppose φ is satisfiable, let $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}^{\mathcal{M}}, \rho^{\mathcal{M}})$ be a TML tree model rooted at r and $\sigma : \mathcal{V} \mapsto \mathcal{D}_r$ such that $\mathcal{M}, r, \sigma \models \varphi$. We define the corresponding FOML tree model $\mathcal{N} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}^{\mathcal{N}}, \rho^{\mathcal{N}})$ as described before.

Define $\mathcal{R}^{\mathcal{N}} = \{(w, u) \mid \text{if there is some } d \in \delta(w) \text{ such that } (w, d, u) \in \mathcal{R}^{\mathcal{M}}\}$ and for every $w \in W$ the valuation function is defined such that for the new unary predicate E we have $\rho^{\mathcal{N}}(w, E) = \{d \mid \text{there is some } w' \in \mathcal{W} \text{ and } d \in \delta(w') \text{ such that } (w', d, w) \in \mathcal{R}^{\mathcal{M}}\}$ and for all the other predicates $Q \neq E$, define the valuation $\rho^{\mathcal{N}}(w, Q) = \rho^{\mathcal{M}}(w, Q)$. Figure 5.1 illustrates one such translation.

It is sufficient to prove that $\mathcal{N}, r, \sigma \models \mathsf{Tr}_2(\varphi)$. For this, we set up the following induction. Recall that $\mathsf{SF}(\varphi)$ is the set of all subformulas of φ and the new unary predicate E introduced in the translation does not occur in $\mathsf{SF}(\varphi)$.

Claim. For all $\psi \in \mathsf{SF}(\varphi)$, for all $w \in \mathcal{W}$ and for all interpretation $\sigma' : \mathcal{V} \mapsto \mathcal{D}$ (relevant at w) we have $\mathcal{M}, w, \sigma' \models \psi$ iff $\mathcal{N}, w, \sigma' \models \mathsf{Tr}_2(\psi)$.

The proof of the claim is by induction on the structure of ψ.

In the base case, we have $Q(x_1, \ldots, x_n)$ and the claim follows since Q is not the newly introduced predicate and $\mathsf{Tr}_2(R((x_1, \ldots, x_n)) = Q((x_1, \ldots, x_n)$ and also $\rho^{\mathcal{N}}(w, Q) = \rho^{\mathcal{M}}(w, Q)$. The case of $\neg\psi$ and $\psi \wedge \psi'$ are standard.

For the case $\Diamond_x \psi$, if $\mathcal{M}, w, \sigma' \models \Diamond_x \psi$ then there is some $(w, \sigma'(x), u) \in \mathcal{R}^{\mathcal{M}}$ such that $\mathcal{M}, u, \sigma' \models \psi$. By induction hypothesis, $\mathcal{N}, u, \sigma' \models \mathsf{Tr}_2(\psi)$. By construction $(w, u) \in \mathcal{R}'$ and $\sigma(x) \in \rho^{\mathcal{N}}(w, E)$. Hence $\mathcal{N}, w, \sigma' \models \Diamond(E(x) \wedge \mathsf{Tr}_2(\psi))$.

On the other hand, if $\mathcal{N}, w, \sigma' \models \Diamond(E(x) \wedge \mathsf{Tr}_2(\psi)$ then there is some $(w, u) \in \mathcal{R}^{\mathcal{N}}$ such that $\mathcal{N}, u, \sigma' \models E(x) \wedge \mathsf{Tr}_2(\psi)$ and by induction hypothesis, $\mathcal{M}, u, \sigma' \models \psi$. Now, since $\mathcal{M}$ is a tree model, w is the unique parent of u, with unique edge label (say d'). By construction, $\mathcal{N}, u \models E(d')$ which implies $\mathcal{N}, u, \sigma' \models E(d') \wedge E(x)$. Also, note that $\mathcal{M}$ is a tree model and since u is a non-root node, there is a unique incoming edge to u which is the only agent for which predicate E is true at u. Hence it has to be the case that $\sigma'(x) = d'$ and thus $(w, \sigma'(x), u) \in \mathcal{R}^{\mathcal{M}}$ which implies $\mathcal{M}, w, \sigma' \models \Diamond_x \psi$.

For $\exists x\ \psi$, we have $\mathcal{M}, w, \sigma' \models \exists x\ \psi$ iff there is some $d \in \delta(w)$ such that $\mathcal{M}, w, \sigma'_{[x \mapsto d]} \models \psi$ iff (by ind.) $\mathcal{N}, w, \sigma'_{[x \mapsto d]} \models \mathsf{Tr}_2(\psi)$ iff $\mathcal{N}, w, \sigma' \models \exists x\ \mathsf{Tr}_2(\psi)$.

Thus it follows from the claim that $\mathcal{N}, r, \sigma \models \varphi$ since $\mathcal{M}, r, \sigma \models \varphi$.

Now we prove the $(\Leftarrow)$ direction in the theorem.

Suppose $\mathcal{N} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}^{\mathcal{N}}, \rho)$ is an FOML model such that $\mathcal{N}, r, \sigma \models \mathsf{Tr}_2(\varphi)$ then to get the TML model, we just need to label the edges $(w, u) \in \mathcal{R}^{\mathcal{N}}$ by looking at the valuation of predicate E at every world u. Define the corresponding TML model[3] $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}^{\mathcal{M}}, \rho)$ where for all $w, u \in \mathcal{W}$ and $d \in \delta(w)$ we have $\mathcal{R}^{\mathcal{M}} = \{(w, d, u) \mid d \in \rho(w) \text{ and } (w, u) \in \mathcal{R}^{\mathcal{N}} \text{ and } d \in \rho(u, E)\}$.

[3] Note that this transformation will not always give us a tree model. In particular, if we have $(w, u) \in \mathcal{R}^{\mathcal{N}}$ and $\{a, b\} \subseteq \rho(u, E)$ then there will be at least two edges between w and u in $\mathcal{M}$ (i.e, $\{(w, a, u),\ (w, b, u)\} \subseteq \mathcal{R}^{\mathcal{M}}\}$). However, this does not matter since we are only concerned with existence of a model.

It is sufficient to prove that $\mathcal{M}, r, \sigma \models \varphi$. Again, we set up the following induction.

Claim. For all $\psi \in \mathsf{SF}(\varphi)$, for all $w \in \mathcal{W}$ and for all interpretation $\sigma' : \mathcal{V} \mapsto \mathcal{D}$ (relevant at w) we have $\mathcal{N}, w, \sigma' \models \mathsf{Tr}_2(\psi)$ iff $\mathcal{M}, w, \sigma' \models \psi$.

The proof follows exactly as in previous case by induction on the structure of ψ. We only illustrate the case $\Diamond_x \psi$.

If $\mathcal{N}, w, \sigma' \models \Diamond(E(x) \wedge \mathsf{Tr}_2(\psi))$ then there is some $(w, u) \in \mathcal{R}^{\mathcal{N}}$ such that $\mathcal{N}, u, \sigma' \models E(x) \wedge \mathsf{Tr}_2(\psi)$ and by ind. hyp, $\mathcal{M}, u, \sigma' \models \psi$. Since $\mathcal{N}, u, \sigma' \models E(x)$ and $(w, u) \in \mathcal{R}^{\mathcal{N}}$ and $\sigma'(x) \in \delta(w)$, by construction $(w, \sigma'(x), u) \in \mathcal{R}^{\mathcal{M}}$ and hence $\mathcal{M}, w, \sigma' \models \Diamond_x \psi$.

If $\mathcal{M}, w, \sigma' \models \Diamond_x \psi$ then there is some $(w, \sigma'(x), u) \in \mathcal{R}^{\mathcal{M}}$ such that $\mathcal{M}, u, \sigma' \models \psi$. Since $(w, \sigma'(x), u) \in \mathcal{R}^{\mathcal{M}}$, by construction $(w, u) \in \mathcal{R}^{\mathcal{N}}$ and $\mathcal{N}, u, \sigma' \models E(x)$. Also, by induction hypothesis $\mathcal{N}, u, \sigma' \models \mathsf{Tr}_2(\psi)$. Thus, $\mathcal{N}, w, \sigma' \models \Diamond(E(x) \wedge \mathsf{Tr}_2(\psi))$. □

In [WZ01], Wolter and Zakharyaschev prove that the monodic restriction for any FOML (without equality) built upon a decidable FO fragment continues to be decidable. Note that the translation in Def. 5.1 preserves monodicity *i.e*, if a TML formula φ is monodic then $\mathsf{Tr}_2(\varphi)$ is a monodic FOML formula. Thus, as a corollary from [WZ01], we have theorems of the following flavour:

Theorem 5.3. *Satisfiability problem for the monodic restriction of* TML *formulas with unary predicates as atoms is decidable.*

When we consider the two variable fragment of term modal logic, the translation to first order modal logic is not useful since the 2-variable fragment of FOML is undecidable [WZ01].

5.2 Constant and Increasing agent models

Recall that $\mathcal{M}$ is a *constant agent TML model* if for all $w \in \mathcal{W}$ we have $\delta(w) = \mathcal{D}$. In this section we will prove there is essentially no difference between constant agent models and increasing agent models in terms of satisfiability problem. In other words, the satisfiability problem for TML over constant agent structures and increasing agent structures is equally hard for most fragments.

Given an increasing agent TML model $\mathcal{M}$, we can obtain a constant agent TML model $\mathcal{N}$ by just setting $\delta^{\mathcal{N}}(w) = \mathcal{D}$ for all w. Further, we encode the information of $\delta^{\mathcal{M}}$ using a unary predicate E such that $d \in \rho^{\mathcal{N}}(w, E)$ iff $d \in \delta^{\mathcal{M}}(w)$. Thus, all quantifications have to be relativized with respect to the new predicate E. This approach is similar to the ones used to prove analogous results for first order modal logic [FM99, WZ01].

Definition 5.4. *Let φ be any TML formula and let E be a new unary predicate not occurring in φ. The translation is defined inductively as follows:*

- $\mathsf{Tr}_3(R(x_1, \ldots, x_n)) = R(x_1, \ldots, x_n)$
- $\mathsf{Tr}_3(\neg\varphi) = \neg\,\mathsf{Tr}_3(\varphi)$ *and* $\mathsf{Tr}_3(\varphi \land \psi) = \mathsf{Tr}_3(\varphi) \land \mathsf{Tr}_3(\psi)$
- $\mathsf{Tr}_3(\Box_x\varphi) = \Box_x(\mathsf{Tr}_3(\varphi))$
- $\mathsf{Tr}_3(\exists x\ \varphi) = \exists x\ (E(x) \land \mathsf{Tr}_3(\varphi))$

Note that in Def. 5.1 (translation of TML to FOML, Tr_2) we relativized the modal formulas with respect to the new predicate. Here we relativize the quantified formulas. Also, Tr_3 preserves the number of variables, quantifier depth and the modal depth.

Since predicate E is used to encode the δ function of the increasing agent model, we need to ensure that the predicate E respects monotonicity. Note that $\forall x \forall y(E(x) \rightarrow \Box_y E(x))$ encodes monotonicity condition for immediate successors. In other words, for any constant agent model $\mathcal{N}$ and $w \in \mathcal{W}^{\mathcal{N}}$ if it is the case that $\mathcal{N}, w \models \forall x \forall y(E(x) \rightarrow \Box_y E(x))$ then for all $c \in \delta^{\mathcal{N}}(w)$ and $u \in \mathcal{W}^{\mathcal{N}}$ $(w, c, u) \in \mathcal{R}^{\mathbb{N}}$ we have $\{d \mid \mathcal{M}, w \models E(d)\} \subseteq \{d' \mid \mathcal{M}, u \models E(d')\}$.

We want this property to be true at all worlds in the model. Since we are dealing with rooted tree models of finite depth (say h), we just have to say that the property $\forall x \forall y(E(x) \rightarrow \Box_y E(x))$ is true at all worlds at height $i \leq h$. Let

$$\gamma_h = \bigwedge_{i<h} (\forall y \Box_y)^i \big(\forall x\ E(x) \rightarrow (\forall y \Box_y\ E(x))\big)$$

where $(\forall z \Box_z)^j(\varphi) = \forall z \Box_z\big((\forall z \Box_z)^{j-1}(\varphi)\big)$ and $(\forall z \Box_z)^0(\varphi) = \varphi$.

Proposition 5.5. *Let $\mathcal{N}$ be any constant agent* TML *model rooted at r and height at most h. Suppose $\mathcal{N}, r \models \gamma_h$ then for all $w, u \in \mathcal{W}^{\mathcal{N}}$ and $c \in \mathcal{D}^{\mathcal{N}}(w)$ if $(w, c, u) \in \mathcal{R}^{\mathcal{N}}$ then $\{d \mid d \in \mathcal{D}$ and $\mathcal{N}, w \models E(d)\} \subseteq \{d' \mid d' \in \mathcal{D}$ and $\mathcal{N}, u \models E(d')\}$.*

Proof. Suppose the claim is false, then there exists $w, u \in \mathcal{W}^{\mathcal{N}}$ and some $c, d \in \mathcal{D}^{\mathcal{N}}$ such that $(w, c, u) \in \mathcal{R}^{\mathcal{N}}$ and $\mathcal{N}, w \models E(d)$ and $\mathcal{N}, u \not\models E(d)$. This together gives: $\mathcal{N}, w \models E(d) \wedge \neg \forall y \Box_y\ (E(d))$ (*).

Let w be at height $j \leq h$. Now since $\mathcal{N}, r \models \gamma_h$, in particular we have $\mathcal{N}, r \models (\forall y \Box_y)^j \big(\forall x\ E(x) \rightarrow (\forall y \Box_y\ E(x))\big)$ and hence $\mathcal{N}, w \models \forall x\ \big(E(x) \rightarrow (\forall y \Box_y\ E(x))\big)$. This is a contradiction to (*). $\Box$

The next lemma states that φ is satisfiable in an increasing agent model iff $\mathsf{Tr}_3(\varphi) \wedge \gamma_{\mathsf{md}(\varphi)}$ is satisfiable in some constant agent model. Moreover, both the formulas are satisfiable over the same agent set $\mathcal{D}$.

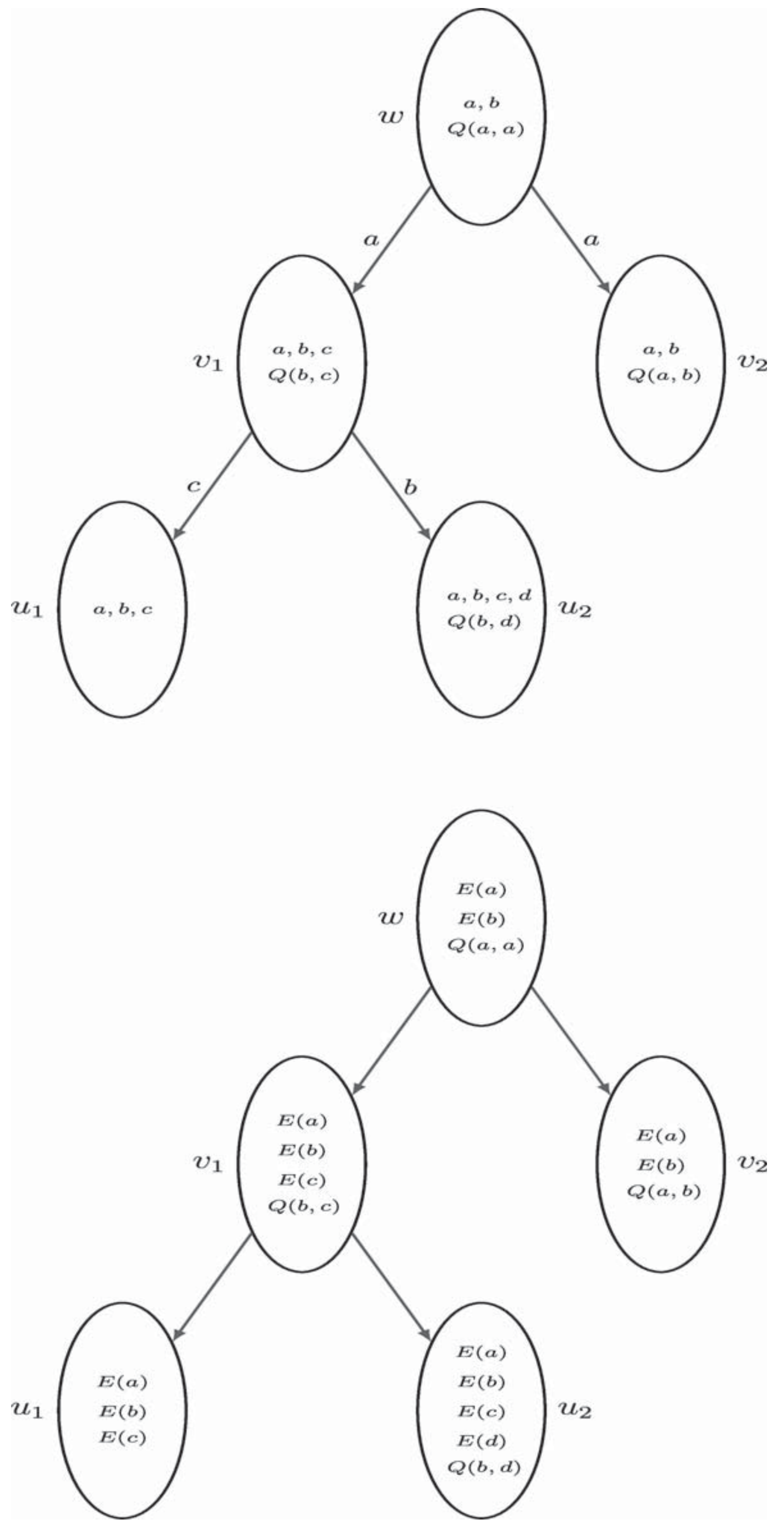


Figure 5.2: Illustration of a translation of increasing agent model to constant agent model. In the translated model, all worlds have the same set of agents $\{a, b, c, d\}$. The predicate E encodes the δ function of the original model.

Lemma 5.6. *Let φ be any* TML *formula. φ is satisfiable in an increasing agent model with agent set $\mathcal{D}$ iff $\gamma_{\mathsf{md}(\varphi)} \wedge \mathsf{Tr}_3(\varphi)$ is satisfiable in a constant agent model with agent set $\mathcal{D}$.*

Proof. ($\Rightarrow$) Suppose $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta^{\mathcal{M}}, \mathcal{R}, \rho^{\mathcal{M}})$ is an increasing agent TML model rooted at $r \in \mathcal{W}$ and σ relevant at r such that $\mathcal{M}, r, \sigma \models \varphi$. Define the constant domain model $\mathcal{N} = (\mathcal{W}, \mathcal{D}, \delta^{\mathcal{N}}, \mathcal{R}, \rho^{\mathcal{N}})$ where $\delta^{\mathcal{N}}(w) = \mathcal{D}$ for all $w \in \mathcal{W}$ and the valuation function is defined such that for the new unary predicate E we have $\rho^{\mathcal{N}}(w, E) = \{d \mid d \in \delta^{\mathcal{M}}(w)\}$ and for all the other predicates $Q \neq E$, define $\rho^{\mathcal{N}}(w, Q) = \rho^{\mathcal{M}}(w, Q)$. Figure 5.2 describes one such translation.

Since δ is monotone, at every $w \in \mathcal{W}$ we have $\mathcal{N}, w \models \forall x(E(x) \to \forall y \Box_y E(x)$. Hence, $\mathcal{N}, r, \sigma \models \gamma_{\mathsf{md}(\varphi)}$. Thus, we can set up a routine induction and prove that for all $\psi \in \mathsf{SF}(\varphi)$ and for all $w \in \mathcal{W}$ and for all interpretation σ' relevant at w, we have $\mathcal{M}, w, \sigma' \models \psi$ iff $\mathcal{N}, w, \sigma' \models \mathsf{Tr}_3(\psi)$.

($\Leftarrow$) Suppose $\mathcal{N} = (\mathcal{W}, \mathcal{D}, \delta^{\mathcal{N}}, \mathcal{R}, \rho^{\mathcal{N}})$ is a constant agent tree model of depth at most $\mathsf{md}(\varphi)$ rooted at $r \in \mathcal{W}$ and σ relevant at r such that $\mathcal{N}, r, \sigma \models \gamma_{\mathsf{md}(\varphi)} \wedge \mathsf{Tr}_3(\varphi)$. Define the increasing agent model $\mathcal{M} = (W, D, \delta^{\mathcal{M}}, \mathcal{R}, \rho)$ where $c \in \delta^{\mathcal{M}}(w)$ iff $\mathcal{M}, w \models E(c)$.

Note that $\mathcal{N}$ is a rooted tree of height at most $\mathsf{md}(\varphi)$ and $\mathcal{N}, r \models \gamma_{\mathsf{md}(\varphi)}$. Hence, by Prop. 5.5, $\delta^{\mathcal{M}}$ defined above is monotone. Again, we can set up a routine induction and prove that for all $\psi \in \mathsf{SF}(\varphi)$ and for all $w \in \mathcal{W}$ and for all interpretation σ' relevant at w we have $\mathcal{N}, w, \sigma' \models \mathsf{Tr}_3(\psi)$ iff $\mathcal{M}, w, \sigma' \models \psi$. $\square$

5.3 Translating TML to PTML

Recall that PTML is the propositional fragment of TML. Now we show that satisfiability problem for PTML is as hard as that for TML.

The reduction is based on the translation of an atomic predicate $P \in \mathcal{P}^n$ of the form $P(x_1, \ldots, x_n)$ to $\Diamond_{x_1} \ldots \Diamond_{x_n} p$ where p is a new proposition which represents the predicate P. However, the coding does not preserve equi-satisfiability all the time. For instance, if the given formula is $\forall x\ (R(x) \wedge \Box_x(P(x) \wedge \neg P(x))$ then the above translation does not work because the given formula asserts that there cannot be any successors for any agent, but the translation needs at least one successor for every agent to encode predicate R.

This problem occurs only when the formula does not allow successors for an agent and can be fixed. We use a new proposition q, to distinguish the worlds in the original model from the ones that are added because of the translation. But now, the modal formulas have to be relativised with respect to q.

Note that a TML formula itself can mention some propositions and these do not require any translation. Hence we distinguish the set of positive arity predicates and propositions occurring in the TML formula.

Definition 5.7. *Let φ be any TML formula where $\mathcal{P}_\varphi = \{P \mid P \in \mathcal{P}^n$ for some $n \geq 1$ and P occurs in $\varphi\}$ and let $\mathcal{P}^0_\varphi = \{s \mid s \in \mathcal{P}^0$ and s occurs in $\varphi\}$ be the set of all positive arity predicates and propositions occurring in φ respectively. Enumerate $\mathcal{P}_\varphi = \{P_1, \ldots P_m\}$ and let n_i be the arity for every $P_i \in P_\varphi$. Let $\{p_1, \ldots, p_m\} \cup \{q\} \subseteq \mathcal{P}^0$ be a new set of propositions not occurring in φ. The translation of φ to a PTML formula is defined inductively as follows:*

- $\mathit{Tr}_4(P_i(x_1, \ldots, x_{n_i})) = \Diamond_{x_1}(\neg q \wedge \Diamond_{x_2}(\ldots \neg q \wedge \Diamond_{x_{n_i}}(\neg q \wedge p_i) \ldots))$
- *For* $s \in \mathcal{P}^0_\varphi$, $\mathit{Tr}_2(s) = s$
- $\mathit{Tr}_4(\neg\varphi) = \neg\mathit{Tr}_4(\varphi)$ *and* $\mathit{Tr}_4(\varphi \wedge \psi) = \mathit{Tr}_4(\varphi) \wedge \mathit{Tr}_4(\psi)$
- $\mathit{Tr}_4(\Box_x\varphi) = \Box_x(q \rightarrow \mathit{Tr}_4(\varphi))$
- $\mathit{Tr}_4(\exists x\ \varphi) = \exists x\ \mathit{Tr}_4(\varphi)$

Thus, for the TML formula $\forall x\ (R(x) \wedge \Box_x(P(x) \wedge \neg P(x))$, the corresponding PTML translation is given by: $\forall x\ \big(\Diamond_x(\neg q \wedge r) \wedge \Box_x(q \rightarrow \Diamond_x(\neg q \wedge p) \wedge \neg \Diamond_x(\neg q \wedge p)\big)$.

Note that Tr_4 preserves the number of variables, and quantifier rank. If the given TML formula φ has modal depth m and highest arity of predicate occurring in φ is k then the modal depth of $\mathsf{Tr}_4(\varphi)$ is $m + k$.

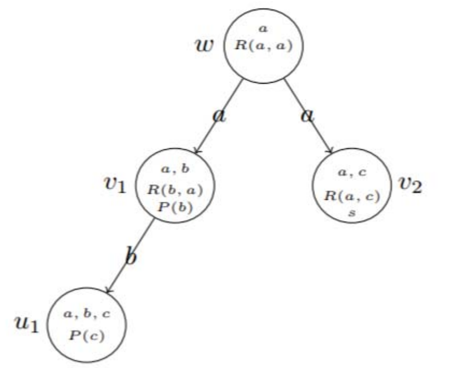
Recall that, in terms of model specification we distinguish TML and PTML models in the definition of the ρ function. For TML models, we have $\rho : (\mathcal{W} \times P) \mapsto \bigcup_n 2^{\mathcal{D}^n}$ which is the valuation function for predicates of arbitrary arity, whereas for PTML models we have $\rho : \mathcal{W} \mapsto 2^{\mathcal{P}^0}$ which is a valuation for propositions.

Lemma 5.8. *For any TML formula φ, φ is satisfiable in an increasing (constant) agent TML model with agent set $\mathcal{D}$ iff $q \wedge \mathsf{Tr}_4(\varphi)$ is satisfiable in an increasing (constant) agent PTML model with agent set $\mathcal{D}$.*

Proof. For any model $\mathcal{M}$ and $u \in \mathcal{W}$ let $\overline{c} \in \mathcal{D}_u^*$ denote a (possibly empty, denoted by ϵ) string of finite length over $\mathcal{D}_u$. Let φ be the given formula and we have $\mathcal{P}_\varphi^0$ and $\mathcal{P}_\varphi$ as defined above. Let k be the highest arity of the predicates occurring in $\mathcal{P}_\varphi$.

($\Rightarrow$) Suppose φ is satisfiable. Let $\mathcal{M}$ be a TML model and $w \in \mathcal{W}^{\mathcal{M}}$ and σ relevant at w such that $\mathcal{M}, w, \sigma \models \varphi$. Define the PTML model $\mathcal{N}$ where:

- $\mathcal{W}^{\mathcal{N}} = \{u_{\overline{c}} \mid u \in \mathcal{W}^{\mathcal{M}}$ and $\overline{c} \in \mathcal{D}_u^*$ of length at most $k\}$.
- For all $u_{\overline{c}} \in \mathcal{W}^{\mathcal{N}}$ we have $\delta^{\mathcal{N}}(u_{\overline{c}}) = \delta^{\mathcal{M}}(u)$.
- $\mathcal{R}^{\mathcal{N}} = \{(u_\epsilon, c, v_\epsilon) \mid (u, c, v) \in \mathcal{R}^{\mathcal{M}}\} \cup \{(u_{\overline{c}}, d, u_{\overline{c}d}) \mid u_{\overline{c}}, u_{\overline{c}d} \in \mathcal{W}^{\mathcal{N}}\}$
- $\rho^{\mathcal{N}}(u_\epsilon) = \{s \mid s \in \mathcal{P}_\varphi^0$ and $\mathcal{M}, u \models s\} \cup \{q\}$ and if $\overline{c} = c_1 \ldots c_n$ has length ≥ 1 then $\rho^{\mathcal{N}}(u_{c_1 \ldots c_n}) = \{p_i \mid \mathcal{M}, u \models P_i(c_1, \ldots, c_n)\}$.



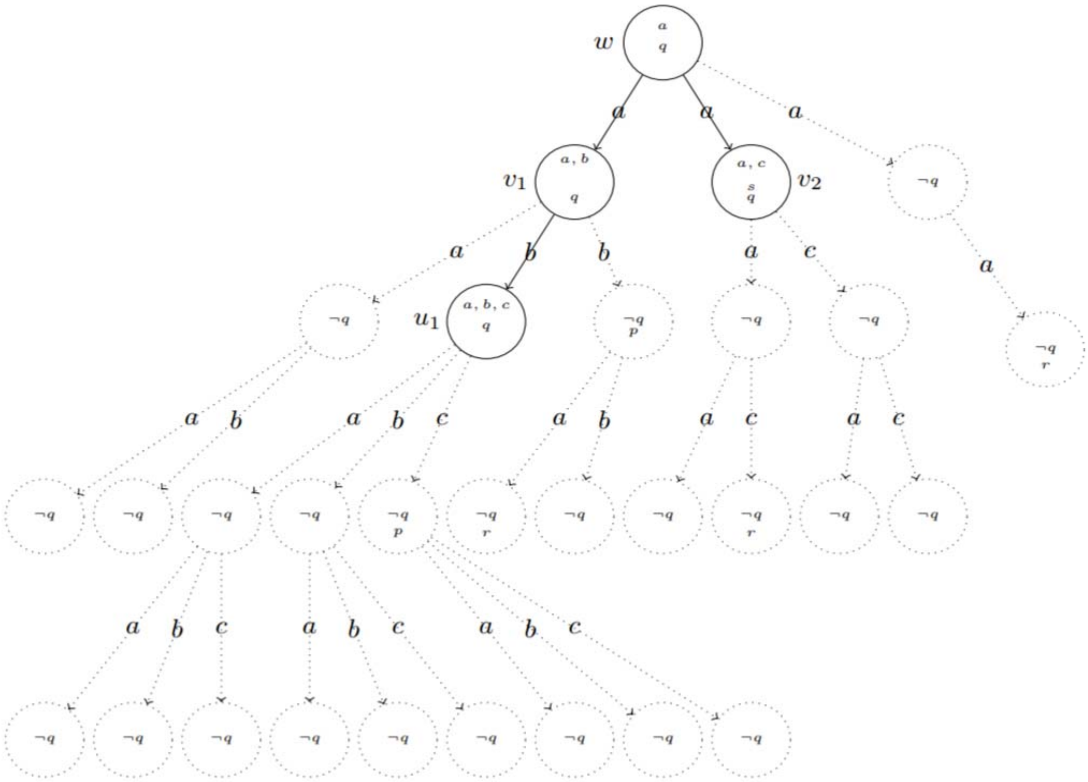


Figure 5.3: Illustration of a translation of TML model to PTML model. The dotted part in the second figure are the new components added during the translation and hence q is true in non-dotted (original) worlds and $\neg q$ holds at every dotted world. Propositions p and r encode predicates P and R respectively. For instance, since $R(b, a)$ is true at v_1, the proposition r is true at the leaf of path $v_1 \xrightarrow{b} v_{1.b} \xrightarrow{a} v_{1.ba}$.

Note that $\mathcal{M}, u, \sigma \models P_i(x_1, \ldots, x_n)$ iff p_i is true at the world $u_{c_1,\ldots,c_n}$ where $\sigma(x_i) = c_i$. Thus by construction, $\mathcal{N}, u_\epsilon, \sigma \models \Diamond_{x_1}(\neg q \wedge \Diamond_{x_2}(\ldots \neg q \wedge \Diamond_{x_n}(\neg q \wedge p_i) \ldots))$ (Refer Fig. 5.3). Also note that for all $u \in \mathcal{W}^{\mathcal{M}}$ we have $\mathcal{M}, u_\epsilon \models q$. Thus a standard inductive argument shows that for all $\psi \in \mathsf{SF}(\varphi)$ and for all $u \in \mathcal{W}^{\mathcal{M}}$ and for all interpretation σ' relevant at u we have $\mathcal{M}, u, \sigma' \models \psi$ iff $\mathcal{N}, u_\epsilon, \sigma' \models q \wedge \mathsf{Tr}_4(\psi)$.

Also note that if $\mathcal{M}$ is an increasing (constant) agent model over $\mathcal{D}$ then $\mathcal{N}$ is also an increasing (constant) agent model over $\mathcal{D}$. Figure 5.3 illustrates a model translation from TML model to PTML model.

($\Leftarrow$) Suppose $\mathcal{N} = (\mathcal{W}^{\mathcal{N}}, \mathcal{D}, \delta^{\mathcal{N}}, \mathcal{R}^{\mathcal{N}}, \rho^{\mathcal{N}})$ is a PTML model with $w \in \mathcal{W}^{\mathcal{N}}$ and σ relevant at w such that $\mathcal{N}, w \models q \wedge \mathsf{Tr}_4(\varphi)$. Define the TML model $\mathcal{M} = (W^{\mathcal{M}}, D, \delta^{\mathcal{M}}, \mathcal{R}^{\mathcal{M}}, \rho^{\mathcal{M}})$ where

- $\mathcal{W}^{\mathcal{M}} = \{u \in \mathcal{W}^{\mathcal{N}} \mid \mathcal{N}, u \models q\}$.
- For all $u \in \mathcal{W}^{\mathcal{M}}$ we have $\delta^{\mathcal{M}}(u) = \delta^{\mathcal{N}}(u)$.
- $\mathcal{R}^{\mathcal{M}} = \mathcal{R}^{\mathcal{N}} \cap (\mathcal{W}^{\mathcal{M}} \times \mathcal{W}^{\mathcal{M}})$.
- For all $P_i \in \mathcal{P}_\varphi$ define
  $\rho^{\mathcal{M}}(u, P) = \{(c_1, \ldots, c_{n_i}) \mid \mathcal{N}, u \models \Diamond_{c_1}(\neg q \wedge (\ldots \Diamond_{c_{n_i}}(\neg q \wedge p_i)\}$ and
  for all $s \in \mathcal{P}^0_\varphi$ define $\rho^{\mathcal{M}}(w, p) = \delta(w)$ iff $\mathcal{N}, w \models s$.

Note that for all $u \in \mathcal{W}^{\mathcal{M}}$ we have $\mathcal{M}, u \models q$. Again, an easy inductive argument shows that for all $\psi \in \mathsf{SF}(\varphi)$ and for all $u \in \mathcal{W}^{\mathcal{M}}$ and for all interpretation σ' relevant at u, we have $\mathcal{N}, u, \sigma' \models q \wedge \mathsf{Tr}_4(\psi)$ iff $\mathcal{M}, u, \sigma' \models \psi$. □

5.4 Bounded agent property

One typical strategy to prove decidability of the satisfiability problem for any logic is to prove a bounded model property.

In general, the theorem is of the following flavour: if a formula φ is satisfiable then φ is satisfiable in some model whose size is bounded by some computable function in the length of φ.

For term modal logic fragments, if we were to employ this strategy, we need to find a model where both $\mathcal{W}$ and $\mathcal{D}$ are of bounded size. Now we will show that it is enough to find a model with finite $\mathcal{D}$ without worrying about the size of $\mathcal{W}$ as long as the fragment is closed under the constant agent and propositional translations (Def 5.4, 5.7 respectively).

Definition 5.9. *Let $\mathcal{F} \subseteq \mathsf{TML}$ be any syntactic fragment of TML,*

- $\mathcal{F}$ *is* 'constant domain closed' *if for every formula $\varphi \in \mathcal{F}$ its corresponding translated formula that is equi-satisfiable in constant domain (Def. 5.4) is in $\mathcal{F}$ i.e, $\mathsf{Tr}_3(\varphi) \wedge \gamma_{md(\varphi)} \in \mathcal{F}$.*
- $\mathcal{F}$ *is* 'predicate closed' *if for every formula $\varphi \in \mathcal{F}$ its corresponding translation to equi-satisfiable PTML formula (Def. 5.7) is in $\mathcal{F}$ i.e, $q \wedge \mathsf{Tr}_4(\varphi) \in \mathcal{F}$.*
- *Let $f : \mathbb{N} \mapsto \mathbb{N}$ be any computable function. $\mathcal{F}$ satisfies* 'f-bounded agent property' *if for every formula $\varphi \in \mathcal{F}$ of length n, if φ is satisfiable then φ is satisfiable in a model $\mathcal{M}$ with size of agent set, $|\mathcal{D}| \leq f(n)$.*

If a fragment of TML, $\mathcal{F} \subseteq \mathsf{TML}$ is 'constant domain closed' it means that any formula $\varphi \in \mathcal{F}$ is satisfiable in an increasing agent model iff φ is satisfiable in a constant agent model. Similarly if $\mathcal{F}$ is 'predicate closed' it means that for any $\varphi \in \mathcal{F}$ there is a corresponding $\mathsf{Tr}_4(\varphi) \in \mathcal{F}$ which has only propositions as atoms such that φ and $\mathsf{Tr}_4(\varphi)$ are equi-satisfiable. Thus, when we consider satisfiability problem, for 'predicate closed fragments', it is enough to assume that the atoms are propositions. In other words, if $\mathcal{F}$ is predicate closed fragment then satisfiability problem for $\mathcal{F}$ is as hard as the satisfiability problem for $\mathcal{F} \cap \mathsf{PTML}$.

Observation 5.10. *Some examples of syntactic fragments:*

1. *TML is both constant domain closed and predicate closed.*
2. *One variable fragment of TML is predicate closed but not constant domain closed since $\gamma_{\mathsf{md}(\varphi)}$ in the translation uses two variables.*
3. *Monodic restriction*[4] *of TML is both constant domain closed and predicate closed.*
4. *Monodic restriction of two variable fragment of TML is constant domain closed but not predicate closed since translation of $R(x, y)$ will not remain monodic.*
5. *For all $k \geq 2$, the k-variable fragment of TML is both constant domain closed and predicate closed.*

Theorem 5.11. *Let $\mathcal{F} \subset$ TML be any syntactic fragment such that $\mathcal{F}$ is both constant domain closed and predicate closed. If $\mathcal{F}$ satisfies f-bounded agent property then satisfiability problem for $\mathcal{F}$ has an $O(n \cdot f(n))$-SPACE algorithm.*

Proof. We prove this by showing that for every formula $\varphi \in \mathcal{F}$ of length n, there is a corresponding equi-satisfiable formula ψ in propositional multi-modal logic ($\mathsf{ML}^{f(n)}$) with at most $f(n)$ agents where the size of ψ is $O(n \cdot f(n))$. Since satisfiability problem for $\mathsf{ML}^{f(n)}$ can be solved in PSPACE (in the size of ψ), the theorem follows.

Suppose $\varphi \in \mathcal{F}$ is a formula of length n, then by Lemma 5.8, there is a corresponding PTML formula $\varphi_1 \in \mathcal{F}$ that is equi-satisfiable to φ with $|\varphi_1| = O(|\varphi|)$. Let $|\varphi_1| = c_1 n$ for some constant c_1. Since $\mathcal{F}$ satisfies f-bounded agent property, without loss of generality, φ_1 is satisfiable iff φ_1 is satisfiable in a model with agent set $\mathcal{D} = \{1, \dots m\}$ for some $m \leq f(c_1 n)$.

[4] Recall that $\varphi \in$ TML is monodic if every subformula of the form $\Delta_x \psi \in \mathsf{SF}(\varphi)$ has $\mathsf{FV}(\psi) \subseteq \{x\}$ where $\Delta \in \{\Box, \Diamond\}$.

Now by Lemma 5.6, there is a corresponding formula $\varphi_2 \in \mathcal{F}$ for φ_1 such that φ_1 is satisfiable iff φ_2 is satisfiable in a constant domain modal with the same agent set $\mathcal{D}$. Let $|\varphi_2| = c_2 n$ for some constant c_2.

Now can expand the quantifiers of φ_2 inductively replacing $\forall x\ \alpha$ by $\bigwedge_{i=1}^{m} \alpha[i/x]$ and $\exists x\ \alpha$ by $\bigvee_{i=1}^{m} \alpha[i/x]$. The resulting formula obtained (say ψ) is a formula in classical propositional multi-modal (ML^m) logic over m agents such that φ_2 and ψ are equi-satisfiable. Note that size of ψ is at most $c_2 n \cdot f(c_1 n)$. Now, satisfiability of ψ can be checked in PSPACE.

$\square$

5.5 Two variable fragment of TML

From Theorem 4.2, it follows that the 3-variable fragment of $\mathsf{PTML}^{\top}$ (and hence TML) is undecidable. On the other hand, every 1-variable formula is also a monodic formula and hence by Theorem 5.3, the 1-variable fragment of TML is decidable. This leaves the 2 variable case. Let TML^2 denote the 2-variable fragment of TML (excluding equality).

The translation of TML formulas to FOML(Def. 5.1) does not introduce any new variables. But this does not help since 2-variable fragment of FOML is undecidable [WZ01]. Quoting Wolter and Zakharyaschev from [WZ01], where they discuss the root of undecidability of FOML fragments:

> All undecidability proofs of modal predicate logics exploit formulas of the form $\square\ \psi(x, y)$ 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 above property is not expressible in TML^2 where there is no 'free' modality; every modality is bound to an index (x or y). With a third variable z, we could indeed encode $\Box P(x, y)$ as $\forall z \Box_z P(x, y)$, but we do not have it. The decidability of the two variable fragment of TML, *without constants or equality*, hinges crucially on this lack of expressiveness. In particular, for any arbitrary TML model $\mathcal{M}$ at some world $w \in \mathcal{W}$ and $c, d \in \delta(w)$ suppose we have $\mathcal{M}, w \models \Box_c(P(c, d)) \wedge \Diamond_d(\neg P(c, d))$, then for all $a \neq c, d$ and for all $w \xrightarrow{a} u$ if we change the valuation of $P(c, d)$ at u then still $\mathcal{M}, w \models \Box_c(P(c, d)) \wedge \Diamond_d(\neg P(c, d))$ will continue to hold.

Note that the fragment TML^2 is both constant agent closed and predicate closed. Hence, it is enough to show agent bounded model property for TML^2 and the decidability follows from Theorem 5.11. Further, note that the translation of TML formulas into equi-satisfiable PTML formulas (Def. 5.7) preserves the number of variables. Therefore it suffices to consider the satisfiability problem for the two variable fragment of PTML. Let PTML^2 denote the 2-variable fragment of PTML.

As is standard with two variable logics, we first introduce a normal form for PTML^2 which is a combination of Fine's normal form for modal logics ([Fin75]) and the Scott normal form ([GKV97]) for FO^2. We then prove a bounded agent property using an argument that can be construed as *modal depth induction* over the 'classical' bounded model construction for FO^2.

5.5.1 Bounded model for FO^2

We now briefly recall the proof steps involved in showing that FO^2 fragment has bounded model property. Let x, y be the variables used. Recall that an FO structure is of the form $\mathfrak{A} = (\mathcal{D}, \rho)$ where$\mathcal{D}$ is the domain and $\rho : \mathcal{P} \mapsto \bigcup_i 2^{\mathcal{D}^i}$ is the interpretation for the predicates.

First it can be proved that every sentence $\varphi \in \mathsf{FO}^2$ has an equi-satisfiable sentence of the form $\forall x \forall y\ \alpha \wedge \bigwedge_j (\forall x \exists y\ \beta_j)$ where α and β_j are all quantifier free. This is called *Scott normal form* [GKV97]. It is obtained by rewriting the given formula where we introduce new unary predicates appropriately (this procedure will be discussed in detail when we take up the normal form for term modal logic).

For a given FO structure $\mathfrak{A}$ for any $c, d \in \mathcal{D}$ we can associate $2\text{-}\mathsf{type}(c, d) = (\Gamma_1; \Gamma_2)$ where Γ_1 and Γ_2 are atomic predicates or negated predicates that are true when (x, y) is assigned to (c, d). Formally,$\Gamma_1 = \{(\neg)P(x, y) \mid \mathfrak{A} \models (\neg)P(c, d)\}$ and $\Gamma_2 = \{(\neg)P(x, y) \mid \mathfrak{A} \models (\neg)P(d, c)\}$.

The 1-type of $c \in \mathcal{D}$ is given by $1\text{-}\mathsf{type}(c) = (\Lambda_1; \Lambda_2)$ where $\Lambda_1 = 2\text{-}\mathsf{type}(c, c)$ and $\Lambda_2 = \{2\text{-}\mathsf{type}(c, d) \mid d \in \mathcal{D}\}$. Let $1\text{-}\mathsf{type}(\mathfrak{A}) = \{1\text{-}\mathsf{type}(c) \mid c \in \mathcal{D}\}$.

Given a Scott normal sentence φ that is satisfiable in $\mathfrak{A}$, we can define a bounded model based on $1\text{-}\mathsf{type}(\mathfrak{A})$ (a similar construction is explained in detail for PTML^2). Since the size of $1\text{-}\mathsf{type}(\mathfrak{A})$ (restricted to the predicates occurring in φ) is at most exponential in the length of φ, the new model that we obtain has size $2^{O(|\varphi|)}$. Thus we get an exponentially bounded model in which φ is satisfiable.

Theorem 5.12. *Satisfiability problem for FO^2 is in* NEXPTIME.

We will essentially follow the same proof steps for PTML^2 but we need to handle modalities indexed by variables along the way.

5.5.2 Normal form for PTML^2

Recall that from Lemma 5.8, proving that satisfiability for PTML^2 is decidable implies that decidability for TML^2 also. Thus we consider PTML^2. We use $x, y \in \mathcal{V}$ as the two variables of PTML^2.

In [Fin75], Fine introduces a normal form for ML (single agent) which is a disjunctive normal form (DNF) where every clause of the form $(\bigwedge_i(s_i) \wedge \Box\alpha \wedge \bigwedge_j \Diamond\beta_j)$ where every s_i is a proposition or its negation and α, β_j are again in the normal form. For FO^2, we have Scott normal form [GKV97] where every FO^2 sentence has an equi-satisfiable sentence of the form $\forall x\forall y\ \varphi \wedge \bigwedge_i \forall x\exists y\ \psi_i$ where φ and every ψ_i are quantifier free.

For PTML^2, we introduce a combination of these two normal forms, which is a DNF formula where every 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 $a, m_x, m_y, n_x, n_y, b \geq 0$ and s_i denotes literals. Further, α and β_j are recursively in the normal form and the formulas $\gamma, \delta_k, \varphi, \psi_l$ do not have quantifiers at the outermost level and all modal subformulas occurring in these formulas are (recursively) in the normal form.

Note that the first two conjuncts mimic the modal normal form and the last two conjuncts mimic the FO^2 normal form. The additional conjuncts handle the intermediate step where only one of the variable is quantified and the other is free. We assume that the formulas are given in negation normal form (NNF) where the negations appear only at the $\mathsf{literals}$.

We use z to refer to either x or y and refer to variables z_1, z_2 to indicate the variables x, y in either order. We use Δ_z to denote any modal operator $\Delta \in \{\Box, \Diamond\}$ and $z \in \{x, y\}$. A $\mathsf{literal}$ is either a proposition or its negation.

Definition 5.13 (FSNF normal form)**.** *We define the following terms to introduce the* Fine Scott normal form *(FSNF) for PTML^2:*

- *A formula φ is a* module *if φ is a* literal *or φ is of the form $\Delta_z\alpha$ where α is any* PTML^2 *formula and $\Delta \in \{\Box, \Diamond\}$ and $z \in \{x, y\}$.*

- *For any formula φ, the outermost components of φ given by $\mathsf{C}(\varphi)$ is defined inductively where for any φ which is a* module*, $\mathsf{C}(\varphi) = \{\varphi\}$ and $\mathsf{C}(Qz\ \varphi) = \{Qz\ \varphi\}$ where $z \in \{x, y\}$ and $Q \in \{\forall, \exists\}$. Finally $\mathsf{C}(\varphi \odot \psi) = \mathsf{C}(\varphi) \cup \mathsf{C}(\psi)$ where $\odot \in \{\wedge, \vee\}$.*

- *A formula φ is* quantifier-safe *if every $\psi \in \mathsf{C}(\varphi)$ is a* module*.*

- *We define Fine Scott normal form(*FSNF*) normal form (*DNF *and conjunctions) inductively as follows:*

  - *Any conjunction of* literals *is an* FSNF *conjunction.*

  - *φ is said to be in* FSNF DNF *if φ is a disjunction of formulas each of which is an* FSNF *conjunction.*

  - *Suppose φ is* quantifier-safe *and for every $\Delta_z\psi \in \mathsf{C}(\varphi)$ if ψ is in* FSNF DNF *normal form then we call φ a* quantifier-safe *normal formula.*

  - *Let $a, b, m_x, m_y, n_x, n_y \geq 0$. Suppose $s_1, \ldots, s_a$ are* literals*, $\alpha^x, \alpha^y, \beta^x_1, \ldots, \beta^x_{m_x}, \beta^y_1, \ldots, \beta^y_{m_y}$ are formulas in* FSNF DNF *and $\gamma^x, \gamma^y, \delta^x_1, \ldots, \delta^x_{n_x}, \delta^y_1, \ldots, \delta^y_{n_y}, \varphi, \psi_1, \ldots, \psi_b$ are* quantifier-safe *normal formulas then the following is an* FSNF *conjunction:*

$$\bigwedge_{i \leq a} s_i \wedge \bigwedge_{z \in \{x,y\}} (\Box_z \alpha^z \wedge \bigwedge_{j \leq m_z} \Diamond_z \beta^z_j) \wedge \bigwedge_{z_1 \in \{x,y\}} (\forall z_2\ \gamma^{z_1} \wedge \bigwedge_{k \leq n_z} \exists z_2\ \delta^{z_1}_k) \wedge \forall x \forall y\ \varphi \wedge \bigwedge_{l \leq b} \forall x \exists y\ \psi_l$$

Quantifier-safe formulas are those in which no quantifiers occur outside the scope of modalities. For instance $\alpha := \forall x(\exists y \Box_x \Diamond_y p \wedge \forall y \Diamond_x \Diamond_y \top)$ is not quantifier-safe. Such formulas are complicated to handle since there is no straight-forward induction parameters that we make can use of. On the other hand, there is an equi-satisfiable formula for α given by $\forall x \exists y(\Box_x \Diamond_y (q \wedge p) \wedge \Diamond_x(\neg q \wedge r)) \wedge \forall x \forall y(\Diamond_x(\neg q \wedge r) \Leftrightarrow \Diamond_x \Diamond_y q)$

where all formulas inside the scope of quantifiers are quantifier-safe and q is newly introduced proposition. With quantifier-safe formulas, we know that all formulas inside the scope of quantifiers are either atoms or have modality at the outer-most level. Thus, in such formulas we can use modal depth of the sub-formulas as an induction parameter.

Note that the superscripts in α^x, α^y etc only indicate which variable the formula is associated with, so that it simplifies the notation. For instance, α^x *does not say anything about the free variables in* α^x. In fact there is no restriction on free variables in any of these formulas.

Also, by setting the appropriate indices to 0, we can have FSNF conjunctions where one or more of the components corresponding to $s_i, \beta^x, \beta^y, \delta^x, \delta^y, \psi_l$ are absent. We also consider the conjunctions where one or more of the components corresponding to $\Box_x \alpha^x, \Box_y \alpha^y, \varphi$ are also absent. As we will see in the next lemma, for any sentence $\varphi \in \mathsf{PTML}^2$, we can obtain an equi-satisfiable sentence, which at the outer most level, is a DNF of formulas of the form $\bigwedge\limits_{i \leq a} s_i \wedge \forall x \forall y\ \varphi \wedge \bigwedge\limits_{l \leq b} \forall x \exists y\ \psi_l$.

For a given PTML^2 formula, we keep rewriting it to get the formula in the normal form. For this, we introduce some new unary predicates in the intermediate steps and finally get rid of them using the translation for TML formulas to PTML formulas (Def. 5.7). The proof essentially follows that of reducing an FO^2 formula into its equi-satisfiable Scott normal form [GKV97].

For the given formula φ, first observe that we can get an equivalent DNF over $\mathsf{C}(\varphi)$ using propositional validities. If φ is modal free, then we can simply ignore the quantifiers, since valuations of propositions do not depend on the quantifiers and the agent set is always non-empty. For instance if $\exists x \forall y\ (p \vee \neg q)$ is the given formula, then $(p \vee \neg q)$ is an equivalent formula[5].

[5]Note that they are equivalent since the local agent set at every world is non-empty.

Thus, for modal free formulas we get an equivalent propositional DNF by erasing the quantifiers and this is in the required form.

If φ contains modal formulas, then we need to reduce every clause of the DNF to an FSNF conjunction. We first translate the formulas at the outer most level to the required form. This is the classical Scott-normal form construction which can be obtained by introducing new unary predicates appropriately to get rid of the nested quantifiers at the outer most level. For instance, if we consider the formula $\alpha := \forall x(\exists y \Box_x \Diamond_y p \wedge \forall y \Diamond_x \Diamond_y \top)$ then we can introduce a new unary predicate R to encode $\forall y \Diamond_x \Diamond_y \top$. Thus, $\beta := \forall x \exists y(\Box_x \Diamond_y p \wedge R(x)) \;\wedge\; \forall x \forall y(R(x) \Leftrightarrow \Diamond_x \Diamond_y \top)$ is the equi-satisfiable translation for α.

Note that β has new unary predicates. This can be eliminated using the translation of TML formulas to PTML formulas given by Def. 5.7 (Tr_4) and we get an equi-satisfiable PTML formula. In the above example, the translation of β is given by $\forall x \exists y(\Box_x \Diamond_y(q \wedge p) \wedge \Diamond_x(\neg q \wedge r)) \wedge \forall x \forall y(\Diamond_x(\neg q \wedge r) \Leftrightarrow \Diamond_x \Diamond_y q)$. We repeat the above step until all formulas inside the scope of quantifiers are quantifier-safe. After this, we replace conjuncts of the form $\Box_z \varphi$ and $\Box_z \psi$ by $\Box_z(\varphi \wedge \psi)$ for $z \in \{x, y\}$ to obtain the resulting formula which has at most one subformula of the from $\Box_x \alpha^x$ and $\Box_y \alpha^y$.

Note that after this translation, the resulting formula is in the required form at the outermost level. We now only need to repeat the entire process for every sub-formula inside the scope of modalities.

The following lemma formally describes the above construction.

Lemma 5.14. *For every formula $\theta \in \mathsf{PTML}^2$ there is a corresponding formula $\hat{\theta} \in \mathsf{PTML}^2$ where $\hat{\theta}$ is a FSNF* DNF *such that θ and $\hat{\theta}$ are equi-satisfiable.*

Proof. We prove this by induction on the modal depth of θ.

Suppose θ has modal depth 0, then all modules occurring in φ are literals. Observe that if α is a propositional formula then for $Q \in \{\forall, \exists\}$ and $z \in \{x, y\}$ and for all model $\mathcal{M}$ we have $\mathcal{M}, w, \sigma \models Qz\ \alpha$ iff $\mathcal{M}, w, \sigma \models \alpha$. Hence we can simply ignore all the quantifiers and get an equivalent DNF over literals, which is an FSNF DNF.

For the induction step, suppose $\mathsf{md}(\theta) = h$. First, let θ_1 be a formula equivalent to θ where θ_1 is a DNF over the outermost components of θ given by $\mathsf{C}(\theta)$. Such a formula θ_1 can be obtained by rewriting θ using propositional validities applied to $\mathsf{C}(\theta)$. Now if θ_1 is an FSNF DNF then we are done.

Otherwise, there are some clauses in θ_1 that are not FSNF conjunctions. Let $\theta_1 := \bigvee_i \zeta_i$ and $I_\theta = \{\zeta_i \mid \zeta_i \text{ is not a FSNF conjunction}\}$ be the clauses that are not FSNF conjunctions. To reduce θ_1 to FSNF DNF, we replace every $\zeta_i \in I_\theta$ with their corresponding equi-satisfiable FSNF DNF in θ_1.

Pick a clause $\zeta \in I_\theta$ and let $\zeta := \omega_1 \wedge \ldots \wedge \omega_n$ that is not an FSNF conjunction. If $\mathsf{md}(\zeta) < h$ then by induction hypothesis, there is an equi-satisfiable FSNF DNF formula of ζ. Thus ζ can be replaced by its corresponding equi-satisfiable FSNF DNF in θ_1. Now suppose $\mathsf{md}(\zeta) = h$.

In the first step, consider the conjuncts with exactly 1 free variable. Let $I_z = \{\omega_i \mid \mathsf{FV}(\omega) = \{z\}\}$ for $z \in \{x, y\}$ be the index of all conjuncts where z is the only free variable. Let z_1, z_2 be the variables x, y in either order. Pick any $\omega_i \in I_{z_1}$ which means z_2 is bound in ω_i. Without loss of generality, ω_i is of the form $\forall z_2\ \eta$. We will first ensure that η is quantifier-safe. This is done by iteratively removing the non-modules from $\mathsf{C}(\eta)$ and replacing it with a equi-satisfiable quantifier-safe formula. Set $\chi_0 := \forall z_2\ \eta$.

a. if there is some strict subformula of the form $Qz_2\ \lambda \in \mathsf{C}(\chi_0)$ where λ is quantifier-safe, let P be a new (intermediate) unary predicate.

Define $\chi_1 := \chi_0[P(z_1)/Qz_2\ \lambda]$ and $\tau_1 := P(z_1) \Leftrightarrow Qz_2\ \lambda$. Note that if $Q = \forall$ then τ_1 can be equivalently written as $\forall z_2\ (\neg P(z_1) \vee \lambda) \wedge \exists z_2\ (P(z_1) \vee \neg\lambda)$ and if $Q = \exists$ then τ_1 will be $\exists z_2\ (\neg P(z_1) \vee \lambda) \wedge \forall z_2\ (P(z_1) \vee \neg\lambda)$.

b. if there is some strict subformula of the form $Qz_1\ \lambda \in \mathsf{C}(\chi_0)$ where λ is quantifier-safe, then we again use a similar translation as in previous case, but since z_2 is bounded, it is universally quantified in τ_1. Let P be a new unary predicate.

Define $\chi_1 := \chi_0[P(z_2)/Qz_1\ \lambda]$ and $\tau_1 := \forall z_2\ (P(z_2) \Leftrightarrow Qz_1\ \lambda)$. Again, that if $Q = \forall$ then τ_1 is equivalent to $\forall z_2 \forall z_1 (\neg P(z_2) \vee \lambda) \wedge \forall z_2 \exists z_1 (P(z_2) \vee \neg\lambda)$ and if $Q = \exists$ then τ_1 is $\forall z_2 \forall z_1\ (P(z_2) \vee \neg\lambda) \wedge \forall z_2 \exists z_1\ (\neg P(z_2) \vee \lambda)$.

Now remove the conjunct ω_i from ζ and replace it with $\chi_1 \wedge \tau_1$. Note that χ_1 has at least one less quantifier than χ_0 and τ_1 introduces either conjuncts with no free variables or a formula with one free variable of the form $Qz\ \lambda$ where λ is quantifier-safe. To see that this step preserves equi-satisfiability, note that in both cases, $\chi_1 \wedge \tau_1$ implies $\forall z_2\ \eta$ and for the other direction, we can define the valuation ρ for the new unary predicate P appropriately in the same model in which ψ is satisfiable.

Repeat this step for $\chi_1, \chi_2, \ldots, \chi_m$ till χ_m is of the form $\forall z_2 \lambda$ where the formula λ is quantifier-safe. Then we would have $\chi_m \wedge \tau_1 \ldots \wedge \tau_m$ as new conjuncts replacing ω_i in ζ. Now this step increases the number of conjuncts in ζ which have no free variables, but all new conjuncts with one free variable are of the form $Qz\ \lambda$ where λ is quantifier-safe. Note that λ needs to be further refined since it is not yet quantifier-safe FSNF (which will be taken up later).

Rewrite all $\omega_i \in I_z$ using the above steps for all $z \in \{x, y\}$. Let the resulting clause be ζ_1 which is equi-satisfiable to ζ.

Now for $z \in \{x, y\}$, if there are two conjuncts of the form $\forall z\ \lambda$ and $\forall z\ \lambda'$ in ζ_1, remove both of them and add $\forall z\ (\lambda \wedge \lambda')$ to ζ_1 and keep doing this until there is a single conjunct in ζ_1 of the form $\forall z\ \gamma^z$ for each $z \in \{x, y\}$ where γ^z is quantifier-safe.

Let $\zeta_1 := \omega'_1 \wedge \ldots \wedge \omega'_{n_1}$ which is the result of rewriting of the clause ζ after the above steps. Note that there are some new unary predicates introduced and hence this intermediate formula ζ_1 may not be in PTML^2 (but is in TML^2).

Now consider conjuncts with no free variables and make them quantifier-safe. Let $I = \{\omega'_i \mid \mathsf{FV}(\psi') = \{x, y\}\}$. For any $\omega'_i \in I$, since neither variable is free, without loss of generality assume that ω'_i is of the form $\forall x\ \eta$.

Pick any $\omega'_i \in I$ and set $\chi_0 := \forall x\ \eta$ and z_1, z_2 refer to x, y in either order. If $Qz_2\ \lambda \in \mathsf{C}(\eta)$, let P be a new unary predicate. Define $\chi_1 := \chi_0[P(z_1)/Qz_2\ \lambda]$ and $\tau_1 := \forall z_1\ (P(z_1) \Leftrightarrow Qz_2\ \lambda)$. Similar to previous step, τ_1 can be equivalently written as two conjuncts of the form $\forall z_1 \forall z_2\ \lambda \wedge \forall z_1 \exists z_2\ \lambda$ where λ and λ' are quantifier-safe formulas (but not quantifier-safe FSNF, yet).

Now remove the conjunct ω'_i from ζ_1 and replace it with $\chi_1 \wedge \tau_1$. Note that χ_1 has at least one less quantifier than χ_0 and τ_1 introduces only conjuncts of the form $Q_1 z_1\ Q_2 z_2\ \lambda$ where λ is quantifier-safe. Again for the equi-satisfiability argument, note that $\chi_1 \wedge \tau_1 \rightarrow \chi_0$ is a validity and for the other direction, the new predicates can be interpreted appropriately in the same model of ζ_1.

Repeat this step for $\chi_1, \chi_2, \ldots, \chi_m$ till χ_m is of the form $\forall x \lambda$ such that λ is quantifier-safe. Then we would have $\chi_m \wedge \tau_1 \ldots \wedge \tau_m$ as new conjuncts replacing ω'_i. Now rename variables appropriately in the newly introduced conjuncts so that we have formulas only of the form $\forall x \forall y\ \lambda$ or $\forall x \exists y\ \lambda'$ where λ, λ' are quantifier-safe formulas.

Rewrite all clauses $\omega'_i \in I$ using the steps described above and let the resulting conjunct be ζ_2 which is equi-satisfiable to ζ_1.

Now if there are two conjuncts of the form $\forall x \forall y\ \lambda$ and $\forall x \forall y\ \lambda'$ in ζ_2, remove both of them and add a new conjunct $\forall x \forall y\ (\lambda \wedge \lambda')$ to ζ_2. Repeat this till at most one conjunct the form $\forall x \forall y\ \lambda$ in ζ_2. Note that we still have unary predicates in ζ_2 and hence ζ_2 is also a TML^2 formula but not a PTML^2 formula. Further, all subformulas inside the scope of quantifiers are now quantifier-safe, but need to be converted into quantifier-safe FSNF.

Let $\zeta_2 := \omega''_1 \wedge \ldots \wedge \omega''_{n_2}$ be the resulting formula after the above steps. Now to eliminate the newly introduced unary predicates, apply the translation in Definition 5.7 to ζ_2 and obtain an equi-satisfiable PTML formula ζ_3. It is clear from the construction that the new predicates are introduced only at the outermost level (not inside the scope of any modality). Thus, in the translation, any occurrence of the newly introduced predicate of the form $P(z)$ will be replaced by $\Diamond_z(\neg q \wedge p)$ and $\neg P(q)$ will be translated to $\neg\Diamond_z(\neg q \wedge p)$ which can be equivalently written as $\Box_z(q \vee \neg p)$. Thus we eliminate the newly introduced unary predicates and ensure all formulas within the scope of quantifiers are quantifier-safe.

Now consider conjuncts that are modal formulas. For $z \in \{x, y\}$, if there are two conjuncts of the form $\Box_z\ \lambda$ and $\Box_z\ \lambda'$ in ζ_3, remove both of them from ζ_3 and add $\Box_z\ (\lambda \wedge \lambda')$ to ζ_3. Repeat this till there is at most one conjunct in ζ_3 of the form $\Box_z\ \alpha^z$ for each $z \in \{x, y\}$. Note that this step preserves equi-satisfiability because of the validity $\forall z\ \big((\Box_z\alpha \wedge \Box_z\beta) \Leftrightarrow \Box_z(\alpha \wedge \beta)\big)$.

By rearranging the conjuncts, we obtain the formula ζ_3 in the form:

$$\bigwedge_{i\leq a} s_i \wedge \bigwedge_{z\in\{x,y\}} (\Box_z\alpha^z \wedge \bigwedge_{j\leq m_z} \Diamond_z\beta^z_j) \wedge \bigwedge_{z\in\{x,y\}} (\forall z\ \gamma^z \wedge \bigwedge_{k\leq n_z} \exists z\ \delta^z_k) \wedge \forall x\forall y\ \varphi \wedge \bigwedge_{l\leq b} \forall x\exists y\ \psi_l$$

where $\gamma^z, \delta^z_k, \varphi$ and ψ_l are all quantifier-safe.

As a final step, we need to ensure that $\alpha^x, \alpha^y, \beta_1^x, \ldots, \beta_{m_x}^x, \beta_1^y, \ldots, \beta_{m_y}^y$ are formulas in FSNF DNF and also the formulas $\gamma^x, \gamma^y, \delta_1^x, \ldots, \delta_{n_x}^x, \delta_1^y, \ldots, \delta_{n_y}^y, \varphi, \psi_1, \ldots, \psi_b$ are not just quantifier-safe, but also quantifier-safe FSNF formulas.

Towards this, note α^z, β_j^z have modal depth less than h. Hence, inductively we have equi-satisfiable FSNF DNF for each of them which can be correspondingly replaced in ζ_3. This preserves equi-satisfiability since we can inductively maintain that the translated formulas are satisfied in the same model of the given formula by just tweaking the ρ function.

To translate $\gamma^x, \gamma^y, \delta_1^x, \ldots, \delta_{n_x}^x, \delta_1^y, \ldots, \delta_{n_y}^y, \varphi, \psi_1, \ldots, \psi_b$ into quantifier-safe FSNF, first note that these formulas are already quantifier-safe. Now for every $\Delta_z\chi \in \mathsf{C}(\mu)$ for μ is one of the above formulas, we have $\mathsf{md}(\chi) \leq h$. Again, inductively we have equi-satisfiable FSNF formulas for each of them. Replacing each such subformula with its corresponding FSNF DNF formula gives us the required FSNF conjunction ζ_4 which is equi-satisfiable to ζ that we started with. Thus ζ can be replaced by ζ_4 in θ_1.

Repeating this for every $\zeta \in I_\theta$ and replacing it in θ_1 we obtain an equi-satisfiable FSNF DNF for θ.

$\square$

Since we repeatedly convert the formula into DNF (inside the scope of every modality), if we start with a formula of length n, the final translated formula has length at most $2^n \cdot 2^n \cdots 2^n$ (n times). Hence the resulting normal form has length $2^{O(n^2)}$.

To illustrate the construction, consider the formula $\alpha := \forall x(\exists y \Box_x \Diamond_y p \wedge \forall y \Diamond_x \Diamond_y \top)$ which we had used to explain the construction informally. First we can introduce a new unary predicate R to encode $\forall y \Diamond_x \Diamond_y \top$.

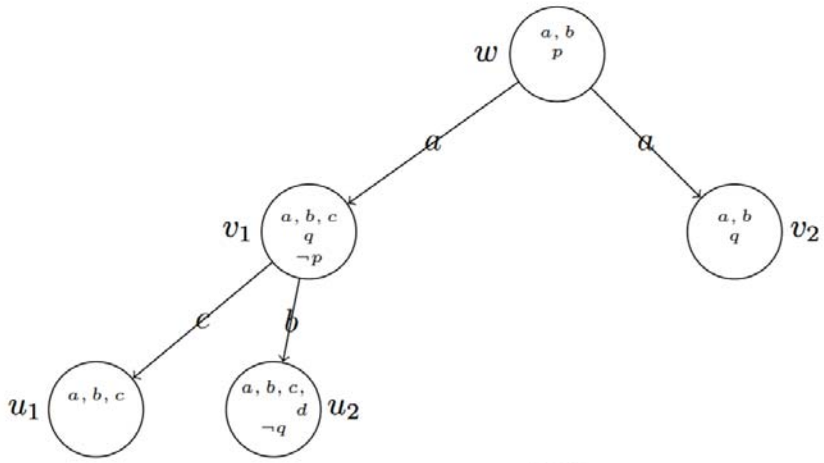
Thus, we get $\beta := \forall x \exists y(\Box_x \Diamond_y p \wedge R(x)) \ \wedge \ \forall x \forall y(R(x) \Leftrightarrow \Diamond_x \Diamond_y \top)$ which is equisatisfiable formula for α. Now the newly introduced unary predicate is eliminated and we obtain: $\forall x \exists y(\Box_x \Diamond_y (q \wedge p) \wedge \Diamond_x(\neg q \wedge r)) \wedge \forall x \forall y(\Diamond_x(\neg q \wedge r) \Leftrightarrow \Diamond_x \Diamond_y q)$ which is in the required form. This process is repeated for sub-formulas inside the scope of a modalities if the formula has larger modal depth.

5.5.3 Model extension

To show bounded model property for PTML^2, if a PTML^2 formula θ is satisfiable in a tree model, the strategy is to inductively come up with bounded agent models for every subtree of the given tree (based on types), starting from leaves to the root. While doing this, when we add new type based agents to a world at height h, to maintain monotonicity, we need to propagate the newly added agents throughout its descendants. For this, we define the notion of extending any tree model by addition of some new set of agents.

Suppose in a tree model $\mathcal{M}$, world w has local agent set $\mathcal{D}_w$ and we want to extend $\mathcal{D}_w$ to $\mathcal{D}_w \cup \mathcal{C}$, then first we have $\Omega : \mathcal{C} \mapsto \mathcal{D}_w$ which assigns every new agent to some already existing agent. The intended meaning is that the newly added agent $c \in \mathcal{C}$ at w mimics the 'type' of $\Omega(c)$. If w is a leaf node, we can simply extend $\delta(w)$ to $\mathcal{D}_w \cup \mathcal{C}$. If w is at some arbitrary height, along with *adding the new agents to the live agent set* to w, we also need to create successors for every $c \in \mathcal{C}$, one for each successor subtree of $\Omega(c)$ and inductively add $\mathcal{C}$ to all the successor subtrees.

Definition 5.15 (Model extension)**.** *Let $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ be a tree model rooted at r. For every $w \in \mathcal{W}$ let $\mathcal{M}^w$ be the sub-tree rooted at w. Let $\mathcal{C}$ be any finite set of new agents such that $\mathcal{C} \cap \mathcal{D} = \emptyset$ and for any $w \in W$ let $\Omega : \mathcal{C} \mapsto \mathcal{D}_w$ be some function mapping $\mathcal{C}$ to the local agent set at w. Define the operation of 'adding $\mathcal{C}$ to $\mathcal{M}^w$ guided by Ω' by induction on the height of w to obtain a new sub-tree rooted at w (denoted by $\mathcal{M}^w_{(\mathcal{C},\Omega)}$).*



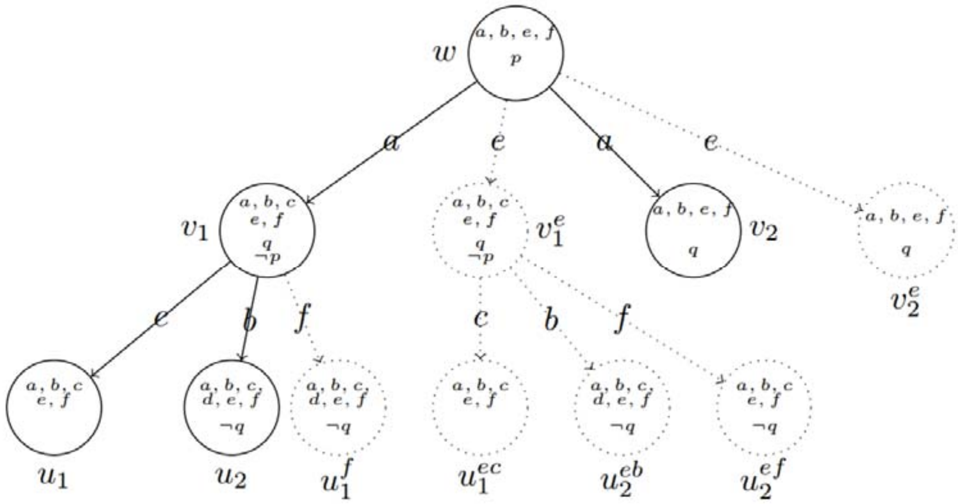


Figure 5.4: Illustration of extending the model by adding e, f as new agents at w with $\Omega(e) = a$ and $\Omega(f) = b$.

- *If w is a leaf, then $M^w_{(\mathcal{C},\Omega)}$ is a tree with a single node w with new $\delta^{\mathcal{M}^w_{(\mathcal{C},\Omega)}}(w) = \delta(w) \cup \mathcal{C}$ and $\rho^{\mathcal{M}^w_{(\mathcal{C},\Omega)}}(w) = \rho(w)$.*

- *If w is at height h, for all $(w, d, u) \in \mathcal{R}$ inductively we have $\mathcal{M}^u_{(\mathcal{C},\Omega)}$ which is a tree model rooted at u obtained by adding $\mathcal{C}$ to $\mathcal{D}_u$ guided by Ω. The new tree $\mathcal{M}^w_{(C,\Omega)}$ is obtained from $\mathcal{M}^w$ rooted at w with $\delta^{\mathcal{M}^w_{(\mathcal{C},\Omega)}}(w) = \delta(w) \cup \mathcal{C}$ and $\rho^{\mathcal{M}^w_{(\mathcal{C},\Omega)}}(w) = \rho(w)$ and for every $(w, d, u) \in \mathcal{R}$ replace the sub-trees $\mathcal{M}^u$ rooted at u by $\mathcal{M}^u_{(\mathcal{C},\Omega)}$, again rooted at u. Further, for every $c \in \mathcal{C}$ and every $(w, \Omega(c), u) \in \mathcal{R}$ create a new copy of $\mathcal{M}^u_{(\mathcal{C},\Omega)}$ and rename its root as u^c and add an edge $(w, c, u^c) \in \mathcal{R}^{\mathcal{M}^w_{(\mathcal{C},\Omega)}}$.*

Figure 5.4 illustrates an effect of this tree operation. The next lemma states that this transformation preserves PTML formulas.

Lemma 5.16. *Let* $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ *be a tree model of finite depth rooted at* r*. Let* $\mathcal{C}$ *be some new agent set such that* $\mathcal{C} \cap \mathcal{D} = \emptyset$ *and for all* $w \in \mathcal{W}$ *and for all* $\Omega : \mathcal{C} \mapsto \mathcal{D}_w$ *let* $\mathcal{M}^w_{(\mathcal{C},\Omega)}$ *(rooted at* w*) be the appropriate model extension of* $\mathcal{M}^w$ *(rooted at* w*). For any interpretation* $\sigma : \mathcal{V} \mapsto (\mathcal{C} \cup \mathcal{D}_w)$ *define* $\hat{\sigma} : \mathcal{V} \mapsto \mathcal{D}_w$ *where* $\hat{\sigma}(x) = \Omega(\sigma(x))$ *if* $\sigma(x) \in \mathcal{C}$ *and* $\hat{\sigma}(x) = \sigma(x)$ *if* $\sigma(x) \in \mathcal{D}_w$*.*

Then for all $w \in \mathcal{W}^w_{(\mathcal{C},\Omega)}$ *for all* $\sigma : \mathcal{V} \mapsto (\mathcal{C} \cup D_w)$ *and for all* PTML *formulas* φ*, we have* $\mathcal{M}^w_{(\mathcal{C},\Omega)}, w, \sigma \models \varphi$ *iff* $\mathcal{M}, w, \hat{\sigma} \models \varphi$*.*

Proof. The proof is by reverse induction on the height of w.

In the base case w is a leaf. Note that $\rho(w)$ remains the same in both the models. Hence all propositional formulas continue to equi-satisfy at w in both the models. Since w is a leaf, there are no descendants in both the models and hence all modal formulas continue to equi-satisfy. Finally, since δ is non-empty in both the models at w, for all formulas $\alpha \in$ PTML we have $\mathcal{M}^w_{(\mathcal{C},\Omega)}, w, \sigma \models Q\ x\ \alpha$ iff $\mathcal{M}, w, \hat{\sigma} \models Q\ x\ \alpha$ where for $Q \in \{\forall, \exists\}$.

For the induction step, let w be at height h. Now we induct on the structure of φ. Again, if φ is a proposition, then the claim follows since $\rho(w)$ remains same. The cases of $\neg$ and $\wedge$ are standard.

For $\Diamond_x\ \varphi$, we need to consider two cases: when $\sigma(x) \in \mathcal{C}$ and $\sigma(x) \in \mathcal{D}_w$.

- If $\sigma(x) \in \mathcal{C}$ then let $\Omega(c) = d$ and hence $\hat{\sigma}(x) = d$. If $M^w_{(\mathcal{C},\Omega)}, w, \sigma \models \Diamond_x\varphi$ then there is some $(w, c, w') \in R^w_{(\mathcal{C},\Omega)}$ such that $\mathcal{M}^w_{(\mathcal{C},\Omega)}, w', \sigma \models \varphi$. By construction, w' is of the form u^c and the subtree rooted at u^c is a copy of $\mathcal{M}^u_{(\mathcal{C},\Omega)}$ for some $(w, d, u) \in \mathcal{R}$. Hence $\mathcal{M}^u_{(\mathcal{C},\Omega)}, u, \sigma \models \varphi$ and by induction hypothesis $\mathcal{M}, u, \hat{\sigma} \models \varphi$. Thus, $\mathcal{M}, w, \hat{\sigma} \models \Diamond_x\varphi$.

Suppose $\mathcal{M}, w, \hat{\sigma} \models \Diamond_x \varphi$, then there is some $(w, d, u) \in \mathcal{R}$ s.t $\mathcal{M}, u, \hat{\sigma} \models \varphi$.

By induction hypothesis, $\mathcal{M}^u_{(\mathcal{C},\Omega)}, u, \sigma \models \varphi$. Now, since $\Omega(c) = d$, by construction there is $(w, d, u^c) \in \mathcal{R}^w_{(C,\Omega)}$ such that the sub-tree rooted at u^c is a copy of $\mathcal{M}^u_{(\mathcal{C},\Omega)}$. Hence $\mathcal{M}^w_{(\mathcal{C},\Omega)}, u^c, \sigma \models \varphi$. Thus $M^w_{(\mathcal{C},\Omega)}, w, \sigma \models \Diamond_x \varphi$.

- If $\sigma(x) \in \mathcal{D}_w$, let $\sigma(x) = d$. Now $\mathcal{M}^w_{(C,\Omega)}, w, \sigma \models \Diamond_x \varphi$ iff there is some $(w, d, u) \in \mathcal{R}^w_{(\mathcal{C},\Omega)}$ such that $\mathcal{M}^w_{(\mathcal{C},\Omega)}, u, \sigma \models \varphi$ iff (by construction) $(w, d, u) \in \mathcal{R}$ and the sub-tree rooted at u in $\mathcal{M}^w_{(C,\Omega)}$ is a copy of $\mathcal{M}^u_{(\mathcal{C},\Omega)}$ iff $\mathcal{M}^u_{(\mathcal{C},\Omega)}, u, \sigma \models \varphi$ iff (by induction) $\mathcal{M}, u, \hat{\sigma} \models \varphi$ iff $\mathcal{M}, w, \hat{\sigma} \models \Diamond_x \varphi$.

For the case of $\exists x\ \varphi$, we have $\mathcal{M}^w_{(C,\Omega)}, w, \sigma \models \exists x\ \varphi$ iff there is some $c \in \mathcal{C} \cup D_w$ such that $\mathcal{M}^w_{(C,\Omega)}, w, \sigma_{[x \mapsto c]} \models \varphi$ iff (by induction) $\mathcal{M}, w, \hat{\sigma}_{[x \mapsto c]} \models \varphi$ iff $\mathcal{M}, w, \hat{\sigma} \models \exists x\ \varphi$. $\square$

5.5.4 Bounded agent property

Now we prove the bounded agent property for PTML^2 formulas that are in FSNF form. Suppose a formula θ in PTML^2 is satisfiable in some tree model $\mathcal{M}$ (of height at most $\mathsf{md}(\theta)$), we build a *'bounded type-based model'* for θ using $\mathcal{M}$.

We need to define the notion of types for agents at every world. In FO^2 we defined 2-types with respect to atomic predicates. In PTML^2 we define the types with respect to modules. In any given tree model $\mathcal{M}$ rooted at r, the 2-type of (c, d) at some world $w \in \mathcal{W}$ is simply the set of all modules that are true at w where the two variables are assigned c, d in either order. The 1-type of c at w includes the set of all modules that are true at w when both x, y are assigned c. Also, for every non-root node w, if $(w' \xrightarrow{a} w)$ then the 1-type of any $c \in D_w$ should also capture how c behaves with respect to a. Further, 1-type(w, c) should also include how c acts with respect to d, for every $d \in \mathcal{D}_w$.

Thus the 1-type of c at w is given by a 3-tuple where the first component is the set of all modules that are true when both x, y are assigned c, the second component captures how c behaves with respect to the incoming edge of w and the third component is a set of subset of formulas such that for each $d \in D_w$ there is a corresponding subset of formulas that captures the 2-type of c, d. To ensure that the type definition also carries the information of the height of the world w, if w is at height h then we restrict 1-type and 2-type at w to modules of modal depth at most $\mathsf{md}(\varphi) - h$.

Recall that for any formula $\varphi \in \mathsf{PTML}^2$, $\mathsf{SF}(\varphi)$ is the set of all subformulas. We always assume that $\top \in \mathsf{SF}(\varphi)$. Let $\mathsf{SF}^h(\varphi) \subseteq \mathsf{SF}(\varphi)$ be the set of all subformulas of modal depth at most $\mathsf{md}(\varphi) - h$. Thus, $\mathsf{SF}(\varphi) = \mathsf{SF}^0(\varphi) \supseteq \mathsf{SF}^1(\varphi) \ldots \supseteq \mathsf{SF}^{\mathsf{md}(\varphi)}(\varphi)$.

Definition 5.17 (PTML type). *For any PTML^2 formula φ and for any tree model $\mathcal{M}$ rooted at r with height at most md(φ), for all $w \in \mathcal{W}$ at height h (let $\Delta \in \{\Box, \Diamond\}$ and $z \in \{x, y\}$):*

- *For all $c, d \in \delta(w)$, define 2-type$(w, c, d) = (\Gamma_{xy}; \Gamma_{yx})$ where*
  $\Gamma_{xy} = \{\Delta_z\psi(x, y) \mid \Delta_z\psi \in SF^h(\varphi)$ *and* $\mathcal{M}, w \models \Delta_z\psi(c, d)\}$ *and*
  $\Gamma_{yx} = \{\Delta_z\psi(x, y) \mid \Delta_z\psi \in SF^h(\varphi)$ *and* $\mathcal{M}, w \models \Delta_z\psi(d, c)\}$.
- *If w is a non root node, (say $w' \xrightarrow{a} w$) then for all $c \in \delta(w)$ define*
  $1\text{-}type(w, c) = (\Lambda_1; \Lambda_2; \Lambda_2)$ *where* $\Lambda_1 = 2\text{-}type(w, c, c)$ *and* $\Lambda_2 = 2\text{-}type(w, c, a)$ *and* $\Lambda_3 = \{2\text{-}type(w, c, d) \mid d \in \delta(w)\}$.
- *For the root node r, for all $c \in \delta(r)$ define* $1\text{-}type(w, c) = (\Lambda_1; \Lambda_2; \Lambda_3)$ *where* $\Lambda_1 = 2\text{-}type(w, c, c)$ *and* $\Lambda_2 = \{\top\}$ *and* $\Lambda_3 = \{2\text{-}type(w, c, d) \mid d \in \delta(w)\}$.

The Λ_2 component in 2-type for root r is added for uniformity.

For all $w \in \mathcal{W}$ define 1-type$(w) = \{$1-type$(w, c) \mid c \in \mathcal{D}_w\}$ and similarly we have 2-type$(w) = \{$2-type$(w, c, d) \mid c, d \in \mathcal{D}_w\}$.

For any formula in normal form, we use the same notations as in Def. 5.13. For a given formula $\theta \in \mathsf{PTML}^2$ in FSNF DNF, let $\boldsymbol{\delta}^x_\theta = \{\exists y\ \delta^x \in \mathsf{SF}(\theta)\}$. Similarly, let $\boldsymbol{\delta}^y_\theta = \{\exists x\ \delta^y \in \mathsf{SF}(\theta)\}$ and $\boldsymbol{\psi}_\theta = \{\forall x \exists y\ \psi \in \mathsf{SF}(\varphi)\}$.

For any tree model $\mathcal{M}$, let $\# \notin \mathcal{D}$. For every $w \in \mathcal{W}$ and for all $\exists y\ \delta \in \boldsymbol{\delta}^x_\theta$ let $g^w_\delta : \mathcal{D}_w \mapsto \mathcal{D}_w \cup \{\#\}$ such that $\mathcal{M}, w \models \delta(c, g^w_\delta(c))$ and $g^w_\delta(c) = \#$ only if there is no $d \in \mathcal{D}_w$ such that $\mathcal{M}, w \models \delta(c, d)$.

Similarly for all $\exists x\ \delta \in \boldsymbol{\delta}^y_\theta$ let $h^w_\delta : \mathcal{D}_w \mapsto \mathcal{D}_w \cup \{\#\}$ such that $\mathcal{M}, w \models \delta(h^w_\delta(c), c)$ and $h^w_\delta(c) = \#$ only if there is no $d \in \mathcal{D}_w$ such that $\mathcal{M}, w \models \delta(d, c)$.

Also for all $\forall x \exists y\ \psi \in \boldsymbol{\psi}_\theta$ let $f^w_\psi : \mathcal{D}_w \mapsto \mathcal{D}_w \cup \{\#\}$ such that $\mathcal{M}, w \models \psi(c, f^w_\psi(c))$ and $f^w_\psi(c) = \#$ only if there is no $d \in \mathcal{D}_w$ such that $\mathcal{M}, w \models \psi(c, d)$.

The functions g, h, f provide the witnesses at a world for every agent (if they exist) for the existential formulas respectively. Let $\mathsf{E}_\theta = \boldsymbol{\delta}^x_\theta \cup \boldsymbol{\delta}^y_\theta \cup \boldsymbol{\psi}_\theta$, note that $|\mathsf{E}_\theta| \leq |\varphi|$.

Definition 5.18 (Witness sets)**.** *Let $\theta \in \mathsf{PTML}^2$ be an FSNF* DNF *and let $\mathsf{E}_\theta = \{\chi_1, \dots \chi_q\}$ for some $q \leq |\mathsf{SF}(\theta)|$. Let $\mathcal{M}$ be any tree model of height at most $\mathsf{md}(\theta)$. For every $w \in \mathcal{W}$ and $a \in \delta(w)$ define $\Upsilon(a) = \{b_1, \dots, b_q\}$ which is a multi-set that gives the witnesses for a where $b_i = g^w_\delta(a)$ if χ_i is of the form $\exists y\ \delta \in \boldsymbol{\delta}^x_\theta$ (similarly $b_i = h^w_\delta(a)$ or $b_i = f^w_\psi(a)$ corresponding to χ_i of the from $\exists x\ \delta^y$ and $\forall x \exists y\ \psi$ respectively). If $b_i = \#$ then set $b_i = b$ for some arbitrary but fixed $b \in \delta(w)$.*

Example 5.19. *Consider a PTML^2 sentence $\theta := \forall x\ \Box_x \Box_x \bot \wedge \forall x \exists y\ (\Box_x(\Diamond_y(\neg p) \wedge \exists y\ \Diamond_y p))$ which is in FSNF* DNF*. Let $\mathcal{M}$ be the model described in Fig. 5.5. Clearly, $\mathcal{M}, r \models \theta$. Let $f^r : \mathcal{D}_r \mapsto \mathcal{D}_r$ be defined by $f^r(2i) = 2i+2$ and at all w^i, $g^i(j) = 2i+1$ for all $i \in \mathbb{N}$ be the two (relevant) witness functions.*

At leaf nodes u^i and v^i there is only one distinct one type and two types. At w^i, note that $r \xrightarrow{2i} w_i$ is the incoming edge and only $2i+1$ and $2i+2$ have outgoing

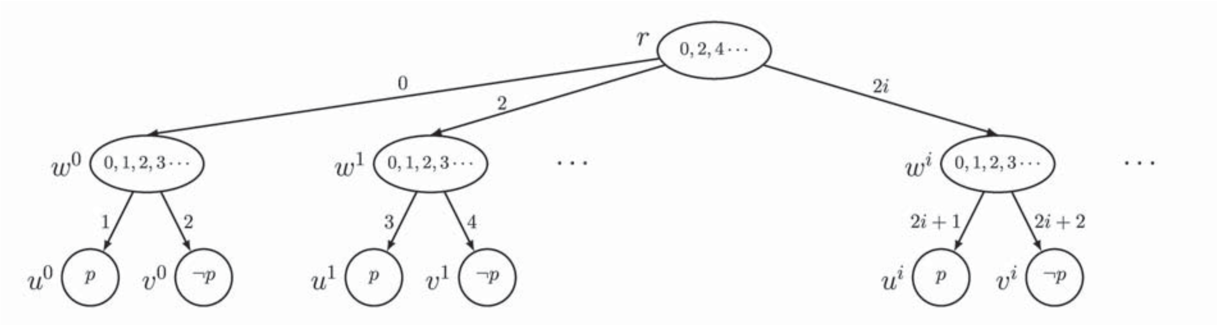


Figure 5.5: Given model such that $\mathcal{M}, r \models \forall x\ \Box_x \Box_x \bot \wedge \forall x \exists y\ (\Box_x(\Diamond_y(\neg p) \wedge \exists y\ \Diamond_y p))$
.

edges. Thus, there are 3 *distinct 1-*type *members at* w^i*, each for* $(2i+1), (2i+2)$ *and [the rest]. At the root again we have only a single distinct* 1-type*.*

Theorem 5.20. *Let* $\theta \in \mathit{PTML}^2$ *be an* FSNF DNF *sentence. Then* θ *is satisfiable iff it is satisfiable in a model with bounded number of agents.*

Proof. It suffices to prove ($\Rightarrow$). Let $\mathcal{M}$ be a PTML tree model of height at most $\mathsf{md}(\theta)$ rooted at r such that $\mathcal{M}, r \models \theta$.

Let $\mathsf{E}_\theta = \{\chi_1, \ldots \chi_q\}$ be some enumeration. For every $w \in \mathcal{W}$ and $a \in \delta(w)$ let $\Upsilon(a) = \{b_1 \ldots b_q\}$ be the witnesses as described above.

For all $w \in \mathcal{W}$ and $\Lambda \in$ 1-$\mathsf{type}(w)$ fix some $a^w_\Lambda \in \delta(w)$ such that 1-$\mathsf{type}(w, a^w_\Lambda) = \Lambda$. Furthermore, if c is the incoming edge of w and 1-$\mathsf{type}(w, c) = \Lambda$ then let $a^w_\Lambda = c$. Let $A^w = \{a^w_\Lambda \mid \Lambda \in$1-$\mathsf{type}(w)\}$.

Now we define the bounded agent model. For every $w \in W$ let $\mathcal{M}^w$ be the subtree model rooted at $w \in \mathcal{W}$. For every such M^w, we define a corresponding *type based model* with respect to θ (denoted by T^w_θ with components denoted by $\delta^w_\theta, \rho^w_\theta$ etc) inductively as follows:

- If w is a leaf then T^w_θ is a tree with a single node w with
  $\delta^w_\theta(w) =$ 1-$\mathsf{type}(w) \times [1 \ldots q] \times \{0, 1, 2\}$ and $\rho^w_\theta(w) = \rho(w)$.

- If w is at height h, T^w_θ is a tree rooted at w with $\delta^w_\theta(w) = 1\text{-}\mathsf{type}(w) \times [1 \dots q] \times \{0,1,2\}$ and $\rho^w_\theta(w) = \rho(w)$.

  Before defining the successors of w in T^w_θ note that for every $(w,a,u) \in \mathcal{R}$ we have T^u_θ which is the inductively constructed type based model rooted at u. Also, inductively we have $\delta^u_\theta(u) =1\text{-}\mathsf{type}(u) \times [1 \dots q] \times \{0,1,2\}$.

  Now for every $a^w_\Lambda \in A^w$ let $\{b_1 \dots b_q\}$ be the corresponding witnesses as described above. For every successor $(w, a^w_\Lambda, u) \in \mathcal{R}$ and for every $1 \le e \le q$ and $f \in \{0,1,2\}$, create a new copy of T^u_θ (call it $\mathcal{N}^{(\Lambda,e,f)}$) and name its root as $u^{(\Lambda,e,f)}$. Now add $\delta^w_\theta(w)$ to $\mathcal{N}^{(\Lambda,e,f)}$ at $u^{(\Lambda,e,f)}$ guided by Ω where Ω is defined as follows:

  - For all $\Pi \in 1\text{-}\mathsf{type}(w)$ we have $a^w_\Pi \in A^w$.

    Define $\Omega((\Pi, e, f)) = (1\text{-}\mathsf{type}(u, a^w_\Pi), e, f)$.
  - For all $k \le q$ if $1\text{-}\mathsf{type}(u, b_k) = \Pi$ then $\Omega((\Pi, k, f')) = (1\text{-}\mathsf{type}(u, b_k), e, f)$ where $f' = f + 1 \mod 3$.
  - Let $f' = f - 1 \mod 3$. For all $\Pi \in$ $1\text{-}\mathsf{type}(w)$ let the witness set of a^w_Π be $\{d_1 \dots d_q\}$. For all $l \le q$ if $1\text{-}\mathsf{type}(w, d_l) = \Lambda$ then by Λ_3 component, there is some $a \in \delta(w)$ such that $2\text{-}\mathsf{type}(w, d_l, a^w_\Pi) = 2\text{-}\mathsf{type}(w, a^w_\Lambda, a)$.

    Define $\Omega((\Pi, l, f')) = (1\text{-}\mathsf{type}(u, a), e, f)$.
  - For all $(\Pi, e', f') \in \delta^w_\theta(w)$ if $\Omega(\Pi, e', f')$ is not yet defined, then set $\Omega(\Pi, e', f') = (1\text{-}\mathsf{type}(u, a^w_\Pi), e, f)$.

  Add an edge $(w, (\Lambda, e, f), u^{(\Lambda,e,f)})$ to $\mathcal{R}^w_\theta$.

Note that Ω is well defined since the first three steps are defined for the indices $f, (f{+}1 \mod 3)$ and $(f\text{-}1 \mod 3)$ respectively, which are always distinct. Also note that T^r_θ is a model that satisfies bounded agent property. Thus, it is sufficient to prove that $T^r_\theta, r \models \theta$.

Claim. For every $w \in \mathcal{W}$ at height h and for all $\lambda \in \mathsf{SF}^h(\theta)$ the following holds:

1. Suppose λ is a sentence and $\mathcal{M}, w \models \lambda$ then $T_\theta^w, w \models \lambda$.

2. If $\mathsf{FV}(\lambda) \subseteq \{x, y\}$ and for all $\Lambda, \Pi \in$ 1-$\mathsf{type}(w)$ if $\mathcal{M}, w, [x \mapsto a_\Lambda^w, y \mapsto a_\Pi^w] \models \lambda$ then for all $1 \leq e \leq q$ and $f \in \{0, 1, 2\}$ we have $T_\theta^w, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi, e, f)] \models \lambda$.

Note that the theorem follows from claim (1), since θ is sentence and $M, r \models \theta$. The proof of the claim is by reverse induction on h.

In the base case $h = \mathsf{md}(\theta)$ which implies λ is modal free and hence λ is a boolean combination of propositions. Thus, both the claims follow since $\rho(w) = \rho_\theta^w(w)$.

For the induction step, let w be at height h. Now we induct on the structure of λ. Again if λ is a $\mathsf{literal}$ then both the the claims follow since $\rho(w) = \rho_\theta^w(w)$. The case of $\wedge$ and $\vee$ are standard.

For the case $\Box_x\lambda$, only claim (2) applies. Let $\mathcal{M}, w, [x \mapsto a_\Lambda^w, y \mapsto a_\Pi^w] \models \Box_x\lambda$. Pick arbitrary e and f. To prove: $T_\theta^w, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi, e, f)] \models \Box_x\lambda$.

Pick any $(w, (\Lambda, e, f), u^{(\Lambda,e,f)}) \in \mathcal{R}_\theta^w$, then by construction we have $(w, a_\Lambda^w, u) \in \mathcal{R}$ and since $\mathcal{M}, w, [x \mapsto a_\Lambda^w, y \mapsto a_\Pi^w] \models \Box_x\lambda$, we have $\mathcal{M}, u, [x \mapsto a_\Lambda^w, y \mapsto a_\Pi^w] \models \lambda$.

Let $a_{\Pi'}^u \in A^u$ such that 1-$\mathsf{type}(u, a_{\Pi'}^u)$ =1-$\mathsf{type}(u, a_\Pi^w)$. Since a_Λ^w is the incoming edge of u, by Π_2 component, we have 2-$\mathsf{type}(u, a_\Pi^w, a_\Lambda^w)$ =2-$\mathsf{type}(u, a_{\Pi'}^u, a_\Lambda^w)$ and also $a_\Lambda^w \in A^u$. Hence $\mathcal{M}, u, [x \mapsto a_\Lambda^w, y \mapsto a_{\Pi'}^u] \models \lambda$ and by induction hypothesis we have $T_\theta^u, u, [x \mapsto (1\text{-}\mathsf{type}(u, a_\Lambda^w), e, f), y \mapsto (1\text{-}\mathsf{type}(u, a_{\Pi'}^u), e, f)] \models \lambda$.

Now by construction (since case 1 of Ω definition applies), at $u^{(\Lambda,e,f)}$ we have $\Omega(\Lambda, e, f) = (1\text{-}\mathsf{type}(w, a_\Lambda^w), e, f)$ and $\Omega(\Pi, e, f) = (1\text{-}\mathsf{type}(u, a_{\Pi'}^u), e, f)$. and by Lemma 5.16 we have $T_\theta^w, u^{(\Lambda,e,f)}, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi, e, f)] \models \lambda$. Since we picked $u^{(\Lambda,e,f)}$ arbitrarily, we have $T_\theta^w, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi, e, f)] \models \Box_x\lambda$.

The case for $\Box_y\lambda$ is analogous.

For case $\Diamond_y\lambda$, again only claim(2) applies. Let $\mathcal{M}, w, [x \mapsto a^w_\Lambda, y \mapsto a^w_\Pi] \models \Diamond_y\lambda$. Now pick any e and f. To prove: $T^w_\theta, w, [x \mapsto (\Gamma, e, f), y \mapsto (\Pi, e, f)] \models \Diamond_y\lambda$. By supposition, there is some $w \xrightarrow{a^w_\Pi} u$ such that $\mathcal{M}, u, [x \mapsto a^w_\Lambda, y \mapsto a^w_\Pi] \models \lambda$. With the argument similar to the previous case, we can show that $T^w_\theta, u^{(\Lambda,e,f)}, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi, e, f)] \models \lambda$ and hence $T^w_\theta, w, [x \mapsto (\Gamma, e, f), y \mapsto (\Pi, e, f)] \models \Diamond_y\lambda$.

The case of $\Diamond_x\lambda$ is symmetric.

For the case $\exists y\ \lambda$ (where x is free at the outer most level), to prove claim (2) first note that since θ is in the normal form, λ is quantifier-safe. Also note that $\exists y\ \lambda = \chi_i$ for some $\chi_i \in E_\theta$. Now suppose $\mathcal{M}, w, [x \mapsto a^w_\Lambda] \models \exists y\ \lambda$. We need to prove that $T^w_\theta, w, [x \mapsto (\Lambda, e, f)] \models \exists y\ \lambda$.

Let the i^{th} witness of a^w_Λ be b_i and hence $M, w, [x \mapsto a^w_\Lambda, y \mapsto b_i] \models \lambda$. Let 1-type$(w, b_i) = \Pi'$, we claim that $T^w_\theta, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \models \lambda$ where $f' = f + 1 \mod 3$. Suppose not, then $\wedge$ and $\vee$ can be broken down and we get some module such that $M, w, [x \mapsto a^w_\Lambda, y \mapsto b_i] \models \Delta_z\lambda'$ and $T^w_\theta, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \not\models \Delta_z\lambda'$ where $\Delta \in \{\Box, \Diamond\}$ and $z \in \{x, y\}$. Assume $\Delta = \Box$ and $z = x$ (other cases are analogous).

Thus $T^w_\theta, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \models \Diamond_x\neg\lambda'$ and hence there is some $w \xrightarrow{(\Lambda,e,f)} u^{(\Lambda,e,f)}$ such that $T^w_\theta, u^{(\Lambda,e,f)}, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \models \neg\lambda'$ (*).

By construction, there is a corresponding $w \xrightarrow{a^w_\Lambda} u$ in $\mathcal{M}$. Now since $\mathcal{M}, w, [x \mapsto a^w_\Lambda, y \mapsto b_i] \models \Box_x\lambda'$, we have $\mathcal{M}, u, [x \mapsto a^w_\Lambda, y \mapsto b_i] \models \lambda'$. Let $b'_i \in A^u$ such that 1-type(u, b_i) =1-type(u, b'_i). Since a^w_Λ is the incoming edge to u by Π'_2 component, we have 2-type(u, b_i, a^w_Λ) =2-type(u, b'_i, a^w_Λ) and $a^w_\Lambda \in A^u$. Thus, $M, u, [x \mapsto a^w_\Lambda, y \mapsto b'_i] \models \lambda'$ and by induction hypothesis we have $T^u_\theta, u, [x \mapsto (\Lambda, e, f), y \mapsto$ (1-type$(u, b'_i), e, f)] \models \lambda'$.

Now by construction, at u we have $\Omega((\Lambda, e, f)) = (\Lambda, e, f)$ and (by case 2 of Ω definition) $\Omega((\Pi', i, f')) = (\textsf{1-type}(u, b_i'), e, f)$ and hence by Lemma 5.16 we have $T_\theta^w, u^{(\Lambda,e,f)}, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \models \lambda'$ which is a contradiction to (*).

The case of $\exists y\ \lambda$ is analogous.

For the case of $\forall x\ \lambda$ (where y is free at the outer most level), to prove claim (2), suppose $\mathcal{M}, w, [y \mapsto a_\Pi^w] \models \forall x\ \lambda$. To prove: $T_\theta^w, w, [y \mapsto (\Pi, e, f)] \models \forall x\ \lambda$.

Pick any $(\Lambda', e', f') \in \delta_\theta^w(w)$. We claim that $T_\theta^w, w, [x \mapsto (\Lambda', e', f'), y \mapsto (\Pi, e, f)] \models \lambda$ (otherwise, like in the previous case, since λ is quantifier-safe, we can reach a module where they differ and obtain a contradiction). The case $\forall y\ \lambda$ is analogous.

Finally we come to sentences which are relevant for claim (1). Note that in the normal form, at the outermost level, a sentence will have only literals or formulas of the form $\forall x \exists y\ \psi_l$ or $\forall x \forall y\ \varphi$.

For the case $\mathcal{M}, w \models \forall x \exists y\ \psi_l$, let $\forall x \exists y\ \psi_l$ be i^{th} formula in E_θ. We need to prove $T_\theta^w, w \models \forall x \exists y\ \psi_l$. Pick any $(\Lambda, e, f) \in \delta_\theta^w(w)$ and we have $a_\Lambda^w \in A^w$. Let the i^{th} witness for a_Λ^w be b_i. Thus we have $M, w, [x \mapsto a_\Gamma, y \mapsto b_i] \models \psi_l$.

Let $\textsf{1-type}(w, b_i) = \Pi'$. We claim that $T_\theta^w, w, [x \mapsto (\Gamma, e, f), y \mapsto [\Pi', e, f')] \models \psi_l$ where $f' = f + 1 \mod 3$. Suppose not, $\wedge$ and $\vee$ can be broken down and we get some module such that $\mathcal{M}, w, [x \mapsto a_\Lambda^w, y \mapsto b_i] \models \Delta_z \lambda'$ and $T_\theta^w, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \not\models \Delta_z \lambda'$ where $\Delta \in \{\Box, \Diamond\}$ and $z \in \{x, y\}$. Assume $\Delta = \Diamond$ and $z = y$ (other cases are analogous).

Hence, $T_\theta^w, w, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \models \Box_y \neg \lambda'$ (*).

Let $a_{\Pi'}^w \in A^w$ such that $\textsf{1-type}(w, a_{\Pi'}^w) = \textsf{1-type}(w, b_i) = \Pi'$. By Π_3' component, there is some $d \in \delta_\theta^w$ such that $\textsf{2-type}(w, a_{\Pi'}^w, d) = \textsf{2-type}(w, b_i, a_\Lambda^w)$ and hence $\mathcal{M}, w, [x \mapsto d, y \mapsto a_{\Pi'}^w] \models \Diamond_y \lambda'$.

Hence there is some $w \xrightarrow{a^w_{\Pi'}} u$ such that $\mathcal{M}, u, [x \mapsto d, y \mapsto a^w_{\Pi'}] \models \lambda'$. Now let $\textsf{1-type}(u, d) = \textsf{1-type}(u, d')$ such that $d' \in A^u$ and since $a^w_{\Pi'}$ is the incoming edge, we have $\mathcal{M}, u, [x \mapsto d', y \mapsto a^w_{\Pi'}] \models \lambda'$ and by induction hypothesis,

$$T^u_\theta, u, [x \mapsto (\textsf{1-type}(u, d'), i, f'), y \mapsto (\textsf{1-type}(u, a^w_{\Pi'}), i, f')] \models \lambda'.$$

Now, while constructing $u^{(\Pi', i, f')}$ (case 3 of Ω definition applies for a^w_Λ) we have $\Omega((\Lambda, e, f' - 1)) = (\textsf{1-type}(u, d'), i, f')$. By Lemma 5.16, $T^w_\theta, u^{(\Pi', i, f')}, [x \mapsto (\Lambda, e, f), y \mapsto (\Pi', i, f')] \models \lambda'$ which contradicts (*).

Finally, for the case $\forall x \forall y\ \varphi$ if $\mathcal{M}, w \models \forall x \forall y\ \varphi$, then we need to prove that $T^w_\theta, w \models \forall x \forall y\ \varphi$. Pick any (Γ, e, f), $(\Delta, e', f') \in \delta^w_\theta(w)$, and we claim that $T^w_\theta, w, [x \mapsto (\Gamma, e, f), y \mapsto (\Delta, e', f')] \models \varphi$ (else again, go to the smallest module where they differ and prove contradiction). □

Corollary 5.21. *Satisfiability problem for $\textsf{TML}^2$ is in* 2-$\textsf{EXPSPACE}$.

Proof. First note that $\textsf{TML}^2$ is both constant domain closed and predicate closed (Def 5.9). Also, $\textsf{TML}^2$ has bounded agent model property, because if a given $\varphi \in \textsf{TML}^2$ is satisfiable then its corresponding $\textsf{PTML}^2$ translation φ is satisfiable iff (by Theorem 5.20) the corresponding normal form θ of φ is satisfiable over agent set $\mathcal{D}$ of size $2^{2^{O(|\varphi|)}}$ iff (by Lemma. Hence $\textsf{TML}^2$ has *bounded agent model* property. Thus by Theorem 5.11, satisfiability problem for $\textsf{TML}^2$ has a 2-$\textsf{EXPSPACE}$ algorithm. □

Note that a $\textsf{NEXPTIME}$ lower bound follows since $\textsf{FO}^2$ is already $\textsf{NEXPTIME}$-complete. This leaves a gap between the known upper bound and the lower bound which needs to be worked out.

Consider the Example 5.19. Recall that the $\textsf{PTML}^2$ sentence under consideration is $\theta := \forall x\ \Box_x \Box_x \bot \wedge \forall x \exists y\ (\Box_x(\Diamond_y(\neg p) \wedge \exists y\ \Diamond_y p))$ which is in $\textsf{FSNF DNF}$ and the model $\mathcal{M}$ described in Fig. 5.5 is given by $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ where

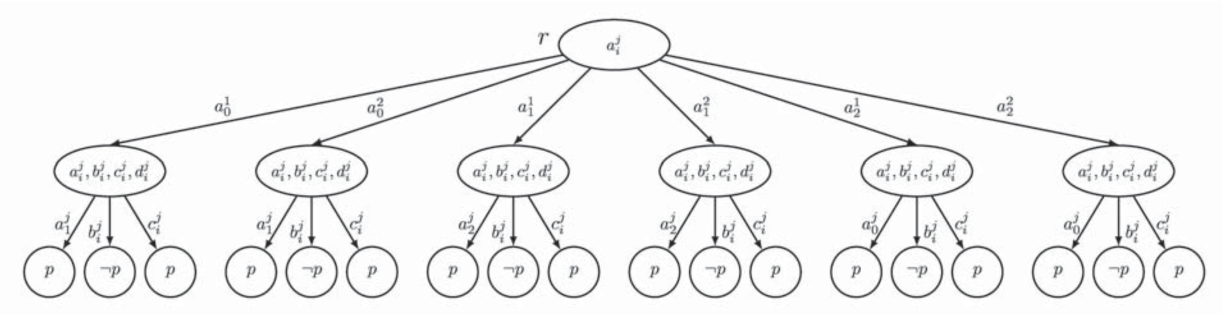


Figure 5.6: Corresponding bounded agent model with $\mathcal{N}, r \models \theta$. a_i^j, b_i^j, c_i^j corresponds to agents with $1 \leq j \leq 2$ and $i \in \{0, 1, 2\}$. The edge a_i^j, b_i^j, c_i^j indicate one successor for every $1 \leq j \leq 2$ and $i \in \{0, 1, 2\}$.

- $\mathcal{W} = \{r\} \cup \{u^i, v^i, w^i \mid i \in \mathbb{N}\}$
- $\mathcal{D} = \mathbb{N}$
- $\delta(r) = \{2i \mid i \in \mathbb{N}\}$ (all even numbers) and
  $\delta(w^i) = \delta(u^i) = \delta(v^i) = \mathbb{N}$
- $\mathcal{R} = \{(r, 2i, w^i), (w^i, 2i + 1, u^i), (w^i, 2i + 2, v^i) \mid i \in \mathbb{N}\}$
- $\rho(r) = \rho(w^i) = \rho(v^i) = \emptyset$ and $\rho(u^i) = p$ for all $i \in \mathbb{N}$.

Clearly, $\mathcal{M}, r \models \theta$ and the witnesses are given by $f^r : \mathcal{D}_r \mapsto \mathcal{D}_r$ be defined by $f^r(2i) = 2i + 2$ and at all w^i, $g^i(j) = 2i + 1$ for all $i \in \mathbb{N}$. Recall that, at leaf nodes u^i and v^i there is only one distinct one type and two types.

At w^i, note that $r \xrightarrow{2i} w_i$ is the incoming edge and only $2i + 1$ and $2i + 2$ have outgoing edges. Thus, there are 3 distinct 1-type members at w^i, each for $(2i+1), (2i+2)$ and [the rest]. Let b, c, d be the respective types. At the root again we have only a single distinct type (call it a).

Since there are 2 existential formulas, the root of the type based model has $(1 \times 2 \times 3) = 6$ agents let it be $\{a_f^e \mid 1 \leq e \leq 2, \ 0 \leq f \leq 2\}$ and 0 be the representative. At w^0 we have $(3 \times 2 \times 3) = 18$ agents. Let the representatives be $1, 2, 0$ for b, c, d respectively.

Note that we cannot pick any other representative for [the rest] other than 0 since 0 is the incoming edge to w^0. Let the bounded agent set be $\{b^e_f,\ c^e_f,\ d^e_f \mid 1 \leq e \leq 2,\ 0 \leq f \leq 2\}$. The corresponding bounded model $\mathcal{N}$ is described in Figure 5.6. It can be verified that $\mathcal{N}, r \models \theta$.

5.6 Bundled fragment

As observed in Chapter 4, one cause for undecidability is the occurrence of quantifiers and modalities independently. Thus we can try to restrict the occurrence of modality and quantifiers in a restricted fashion to look for decidable fragments. One promising approach towards this direction is where we have formulas only of the form $\exists x \Box_x \alpha$ and $\forall x \Box_x \alpha$ (and $\forall x \Diamond_x \alpha$ and $\exists x \Diamond_x \alpha$ dually). Note that we do not impose any restriction on the arity of the predicates.

In [PRW18], we consider the analogous notion of *bundling* for FOML where we have formulas of the form $\forall x \Box \alpha$ and $\exists x \Box \alpha$ (again without any restriction on the arity of predicates). This fragment is suitable to study epistemic logics for the notions of knowing-how, knowing-why, knowing-what (see Wang, [Wan17]). For instance, $\exists x \Box \varphi$ may mean that there exists a mechanism which agent knows such that executing it will make sure in a φ state.

For this fragment, it critically matters if the models considered are increasing domain FOML models or constant domain FOML models. Table 5.1 describes the summary of decidability results.

Note that the translation of TML formulas to FOML formulas (Def. 5.1, Tr_2) preserves bundled formulas i.e, if $\varphi \in$ TML is a bundled formula then $\mathsf{Tr}_2(\varphi)$ is a bundled FOML formula. Thus positive results of Table 5.1 hold for bundled fragment of TML.

Language	Model	Decidability	Remark
$\forall\Box$, P^1	Const.	undecidable	
$\exists\Box$, P	Const.	decidable	PSPACE-complete
$\exists\Box, \forall\Box$, P	Inc	decidable	PSPACE-complete

Table 5.1: Satisfiability problem classification for Bundled FOML fragment, P refers to predicates of arbitrary arity and P^1 refers to unary predicates. Models are either constant domain or increasing domain.

5.6.1 Implicitly quantified modal logic

When we consider the bundled fragment of PTML, we have formulas of the form $\forall x \Box_x \alpha$ and $\exists x \Box_x \alpha$ where α contains only propositions as atoms. Since there are no predicates of positive arity, the variables can be eliminated from the syntax and we can define a bimodal propositional logic where $\exists x \Box_x\ \alpha$ can be replaced by a variable free modality $[\exists]\alpha$, and similarly $\forall x \Box_x\ \alpha$ is replaced by $[\forall]\ \alpha$. In this logic, the modalities are *implicitly quantified* and thus we call this *implicitly quantified modal logic* (IQML) [PR19b]. Note that even though PTML has only propositions, it is still a quantified logic. In the spirit of propositional modal logic, IQML eliminates variables altogether.

In an epistemic setting, IQML coincides with the notion of *somebody knows* and *everybody knows*, when the set of reasoners is not fixed a priori. Grove and Halpern [GH93, Gro95] discuss such a logic where the agent set is not fixed and the agent names are not common knowledge. Khan et al. [KP18] use a logic similar to IQML to study approximations in the context of rough sets.

Definition 5.22 (IQML syntax)**.** *Let $\mathcal{P}^0$ be a countable set of propositions. The syntax of IQML is given by:*

$$\varphi := p \in \mathcal{P}^0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\exists]\ \varphi \mid [\forall]\ \varphi$$

The dual modalities of $[\exists]$ and $[\forall]$ are respectively defined by $\langle\forall\rangle\varphi := \neg[\exists]\neg\varphi$ and $\langle\exists\rangle\varphi := \neg[\forall]\neg\varphi$.

Note that we do not have free variables to talk about and hence we can relax the monotonicity condition imposed for PTML structures. Thus, we can specify structures for IQML in a way closer to that of propositional multi-modal logic (ML^n). The only difference is that the Kripke structure for ML^n is given by $\mathcal{M} = (\mathcal{W}, \mathcal{R}_1, \ldots, \mathcal{R}_n, \rho)$ where each $R_i \subseteq (\mathcal{W} \times \mathcal{W})$ is the accessibility relation for the corresponding index and ρ is the valuation of propositions at every world, whereas in the case of IQML, the modal index set is specified along with the model.

Definition 5.23 (IQML structure)**.** *An IQML structure is given by the tuple $\mathcal{M} = (\mathcal{W}, \mathcal{I}, \mathbb{R}, \delta, \rho)$ where $\mathcal{W}$ is a non-empty set of worlds, $\mathcal{I}$ is a non-empty countable index set and $\mathbb{R} = \{\mathcal{R}_i \mid i \in \mathcal{I}\}$ where each $\mathcal{R}_i \subseteq (\mathcal{W} \times \mathcal{W})$ and $\delta : \mathcal{W} \mapsto 2^{\mathcal{I}}$ such that whenever $(w, u) \in \mathcal{R}_i$ we have $i \in \delta(w)$ and $\rho : \mathcal{W} \mapsto 2^{\mathcal{P}}$ is the valuation function.*

The only difference between IQML and PTML models is that the monotonicity condition on δ is relaxed for IQML. The agent set $\mathcal{I}$ is countable, and hence we assume $\mathcal{I}$ to be some initial segment of $\mathbb{N}$ or $\mathbb{N}$ itself.

Definition 5.24 (IQML semantics)**.** *Given a IQML model $\mathcal{M}$, an IQML formula φ and $w \in \mathcal{W}^{\mathcal{M}}$, define $\mathcal{M}, w \models \varphi$ inductively as follows:*

$\mathcal{M}, w \models p$	$\Leftrightarrow$	$p \in \rho(w)$
$\mathcal{M}, w \models \neg\varphi$	$\Leftrightarrow$	$\mathcal{M}, w, \not\models \varphi$
$\mathcal{M}, w \models (\varphi \wedge \psi)$	$\Leftrightarrow$	$\mathcal{M}, w \vDash \varphi$ *and* $\mathcal{M}, w \models \psi$
$M, w \models [\exists]\varphi$	$\Leftrightarrow$	*there is some* $i \in \delta(w)$ *such that for all* $u \in \mathcal{W}$ *if* $(w, u) \in R_i$ *then* $\mathcal{M}, u \models \varphi$
$\mathcal{M}, w \vDash [\forall]\varphi$	$\Leftrightarrow$	*for all* $i \in \delta(w)$ *and for all* $u \in \mathcal{W}$ *if* $(w, u) \in R_i$ *then* $\mathcal{M}, u \models \varphi$

A formula $\varphi \in \mathsf{IQML}$ is *satisfiable* if there is some model $\mathcal{M}$ and $w \in \mathcal{W}$ such that $\mathcal{M}, w \models \varphi$. A formula φ is said to be *valid* if $\neg\varphi$ is not satisfiable.

Given any model $\mathcal{M}$ with $w \in \mathcal{W}$ and a formula $[\exists]\varphi$, if $\mathcal{M}, w \models [\exists]\varphi$ and $i \in \mathcal{I}$ is the corresponding witness then we define the notation $\mathcal{M}, w \models \Box_i\varphi$. Similarly we have $\mathcal{M}, w \models \Diamond_i\varphi$ for $\langle\exists\rangle\varphi$. Note that the elements of $\mathcal{I}$ are not present in the syntax.

Note that IQML is a sub-fragment of bundled fragment of TML. In fact, IQML is exactly the set of bundled TML formulas whose atoms are restricted to propositions. Thus, the satisfiability for IQML is decidable in PSPACE. In this section we give a complete axiom system for IQML and show that there is a canonical IQML model. This gives an alternate decidability proof for IQML. Table 5.2 gives a complete axiom system for the valid formulas of IQML. The proof is along the standard lines, with the main interest being in how agent names are synthesized in the model construction since the syntax is variable free.

The axioms and inference rules are standard. Axiom $A2$ describes the interaction between $[\forall]$ and $\langle\forall\rangle$ operators. The ($[\exists]$Nec) rule is sound since $\mathcal{I}$ is non-empty. Note that the axiom system is similar to the one discussed by Grove and Halpern [GH93], except for ($[\forall]$Nec) and ($[\exists]$Nec). This is because IQML has no names, as opposed to the logic considered in [GH93].

Lemma 5.25. *The axiom system $\vdash_{\mathcal{AX}_\mathcal{A}}$ is sound for IQML.*

Proof. We only prove that $A2$ is a validity. For any model $\mathcal{M}$ and any world w let $\mathcal{M}, w \models [\forall](\varphi \to \psi)$ and $\mathcal{M}, w \models \langle\forall\rangle\varphi$. Since $\mathcal{M}, w \models [\forall](\varphi \to \psi)$ for all $i \in \delta(w)$ and for all $w \xrightarrow{i} u$ we have $\mathcal{M}, u \models \varphi \to \psi$. Further since $\mathcal{M}, w \models \langle\forall\rangle\varphi$, for all $i \in \delta(w)$ there is some v such that $w \xrightarrow{i} v$ and $\mathcal{M}, v \models \varphi$. But then $\mathcal{M}, v \models \varphi \to \psi$ and hence $\mathcal{M}, v \models \psi$. Thus by semantics, $\mathcal{M}, w \models \langle\forall\rangle\psi$. $\square$

	$\vdash_{\mathcal{AX}_{\mathcal{A}}}$
$A0.$	All instances of propositional validities.
$A1.$	$[\forall](\varphi \to \psi) \to ([\forall]\varphi \to [\forall]\psi)$
$A2.$	$[\forall](\varphi \to \psi) \to (\langle\forall\rangle\varphi \to \langle\forall\rangle\psi)$
(MP)	$\dfrac{\varphi \to \psi, \varphi}{\psi}$
([∀]Nec)	$\dfrac{\varphi}{[\forall]\varphi}$
([∃]Nec)	$\dfrac{\varphi}{[\exists]\varphi}$

Table 5.2: IQML axiom system ($\mathcal{AX}_{\mathcal{A}}$)

A set of IQML formulas Γ is *consistent* if there is no formula α such that $\Gamma \vdash_{\mathcal{AX}_{\mathcal{A}}} \alpha$ and $\Gamma \vdash_{\mathcal{AX}_{\mathcal{A}}} \neg\alpha$. Also, Γ is *maximally consistent* if Γ is consistent and for every $\psi \in$ IQML either $\psi \in \Gamma$ or $\neg\psi \in \Gamma$. Before proving completeness, we first prove some useful lemmas.

Lemma 5.26. *Given a set of formulas $\Gamma \subseteq \overline{SF}(\varphi)$, if Γ is a maximal consistent set then*

1. *if $\langle\exists\rangle\beta \in \Gamma$ then $\{\beta\} \cup \{\psi \mid [\forall]\psi \in \Gamma\}$ is consistent.*
2. *if $\{\langle\forall\rangle\gamma,\ [\exists]\delta\} \subseteq \Gamma$ then $\{\gamma, \delta\} \cup \{\psi \mid [\forall]\psi \in \Gamma\}$ is consistent.*

Proof. To prove (1), let Γ be a maximal consistent set of formulas and $\langle\exists\rangle\beta \in \Gamma$. Define $\Lambda = \{\beta\} \cup \{\psi \mid [\forall]\psi \in \Gamma\}$. We need to prove that Λ is consistent. Suppose not, then there are some $\psi_1, \psi_2 \cdots \psi_n \in \Lambda$ such that

$$\vdash_{\mathcal{AX}_{\mathcal{A}}} (\psi_1 \wedge \psi_2 \cdots \psi_n) \to \neg\beta.$$

By ([∀]Nec) we have $\vdash_{\mathcal{AX}_{\mathcal{A}}} [\forall]\big((\psi_1 \wedge \psi_2 \cdots \psi_n) \to \neg\beta\big)$.

By ($A1$) and (MP), $\vdash_{\mathcal{AX}_{\mathcal{A}}} [\forall](\psi_1 \wedge \psi_2 \cdots \psi_n) \to [\forall]\neg\beta$.

Also note that $([\forall]\psi_1 \wedge [\forall]\psi_2 \cdots [\forall]\psi_n) \to [\forall](\psi_1 \wedge \psi_2 \cdots \psi_n)$ is a theorem in this system. Hence $\vdash_{\mathcal{AX}_{\mathcal{A}}} ([\forall]\psi_1 \wedge [\forall]\psi_2 \ldots \wedge [\forall]\psi_n) \to [\forall]\neg\beta$. This implies $[\forall]\neg\beta \in \Gamma$ which is a contradiction to $\langle\exists\rangle\beta \in \Gamma$ (since Γ is maximally consistent).

To prove (2), again let Γ be a maximal consistent set of formulas and let $\{\langle\forall\rangle\gamma,\ [\exists]\delta\} \subseteq \Gamma$. Define $\Lambda = \{\gamma,\ \delta\} \cup \{\psi \mid [\forall]\psi \in \Gamma\}$. We need to prove that Λ is consistent. Suppose not, then there are some $\psi_1, \psi_2 \cdots \psi_n \in \Lambda$ such that $\vdash_{\mathcal{AX_A}} (\psi_1 \wedge \psi_2 \cdots \psi_n) \rightarrow (\gamma \rightarrow \neg\delta)$.

Now arguing in the same way as in (1) we have

	$\Gamma \vdash_{\mathcal{AX_A}} [\forall](\gamma \rightarrow \neg\delta)$
By $(A2)$	$\Gamma \vdash_{\mathcal{AX_A}} [\forall](\gamma \rightarrow \neg\delta) \rightarrow (\langle\forall\rangle\gamma \rightarrow \langle\forall\rangle\neg\delta)$
By (MP)	$\Gamma \vdash_{\mathcal{AX_A}} \langle\forall\rangle\gamma \rightarrow \langle\forall\rangle\neg\delta$
Since $\langle\forall\rangle\gamma \in \Gamma$,	$\Gamma \vdash_{\mathcal{AX_A}} \langle\forall\rangle\neg\delta$.

This is a contradiction since $[\exists]\delta \in \Gamma$ and Γ is consistent. □

Now we define the canonical model. Let $EB = \{[\exists]\alpha \mid [\exists]\alpha \in \mathsf{IQML}\}$ be the set of all $[\exists]$ IQML formulas. Let $EB = \{\varphi_1, \varphi_2, \ldots\}$ be some enumeration. These formulas will be used as 'agents' in the canonical model.

Definition 5.27. *The canonical IQML model is given by* $\hat{\mathcal{M}} = (\hat{\mathcal{W}}, \hat{\mathcal{I}}, \hat{\mathcal{R}}, \hat{\delta}, \hat{\rho})$ *where*

- $\hat{\mathcal{W}}$ *is set of all* maximal consistent sets.
- $\hat{\mathcal{I}} = \{i_{[\exists]\alpha} \mid [\exists]\alpha \in EB\} \cup \{0\}$ *where* 0 *is a new distinguished agent.*
- *For all* $w, u \in \hat{\mathcal{W}}$,

  For all $[\exists]\alpha \in EB$ *we have* $w \xrightarrow{i_{[\exists]\alpha}} u$ *if* $\{\alpha\} \cup \{\psi \mid [\forall]\psi \in w\} \subseteq u$.

  For $0 \in \hat{\mathcal{I}}$ *we have* $w \xrightarrow{0} u$ *if* $\{\psi \mid [\forall]\psi \in w\} \subseteq u$.
- *For all* $w \in \hat{\mathcal{W}}$ *define* $\hat{\delta}(w) = \{i_{[\exists]\alpha} \mid [\exists]\alpha \in w\} \cup \{0\}$
- $\hat{\rho}(w) = w \cap \mathcal{P}$.

Lemma 5.28. *In the canonical model, for all* $w, u \in \hat{\mathcal{W}}$ *and* $i \in \hat{\delta}(w)$ *if* $w \xrightarrow{i} u$ *then for all* $\psi \in u$ *we have* $\langle\exists\rangle\psi \in w$.

Theorem 5.29. $\vdash_{\mathcal{AX_A}}$ *is a complete axiom system for IQML.*

Proof. We show this by proving that any consistent formula $\varphi \in \mathsf{IQML}$ is satisfiable. First note that any consistent set of formulas Γ can be extended to a maximal consistent set by the standard *Lindenbaum construction*. Hence for any consistent set of formulas Γ, there is some world $w \in \hat{\mathcal{W}}$ such that $\Gamma \subseteq w$. Now, we prove the truth lemma.

Claim. For any $w \in \hat{\mathcal{W}}$, $\hat{\mathcal{M}}, w \models \varphi$ iff $\varphi \in w$.

The proof is by induction on the structure of φ. In the base case we have propositions and the claim follows by definition of $\hat{\rho}$. The $\neg$ and $\wedge$ cases are standard.

For the case $\varphi := \langle \exists \rangle \beta$, suppose $\hat{\mathcal{M}}, w \models \langle \exists \rangle \beta$ then there is some $a \in \hat{\delta}(w)$ and some $(w, a, u) \in \hat{\mathcal{R}}$ such that $\hat{\mathcal{M}}, u \models \beta$. By induction hypothesis $\beta \in u$ and by lemma 5.28, $\langle \exists \rangle \beta \in w$.

For the other direction, suppose $\langle \exists \rangle \beta \in w$ then since w is a consistent set (by lemma 5.26(1)) we have $\Gamma = \{\beta\} \cup \{\psi \mid [\forall]\psi \in w\}$ is consistent. Thus there is some world $u \in \hat{\mathcal{W}}$ such that $\Gamma \subseteq u$. Now since $\beta \in u$, by induction hypothesis $\hat{M}, u \models \beta$ and also since $\{\psi \mid [\forall]\psi \in w\} \subseteq u$ we have $w \xrightarrow{0} u$ and hence $\hat{M}, w \models \langle \exists \rangle \beta$.

For the case $\varphi := [\exists]\beta$, To prove ($\Rightarrow$), we consider the contrapositive. We prove that if $[\exists]\beta \notin w$ then $\hat{M}, w \models \langle \forall \rangle \neg \beta$. Let $[\exists]\beta \notin w$. Since w is maximally consistent $\langle \forall \rangle \neg \beta \in w$. Pick arbitrary $i \in \hat{\delta}(w)$, then i is of the form $i_{[\exists]\alpha}$ or $i = 0$.

For $i_{[\exists]\gamma} \in \hat{\delta}(w)$, we have $[\exists]\gamma \in w$ and by Lemma 5.26(2), $\Gamma = \{\neg\beta, \gamma\} \cup \{\psi \mid [\forall]\psi \in w\}$ is consistent. Thus there is some world $v \supseteq \Gamma$ and by construction of the canonical model, $w \xrightarrow{i_{[\exists]\gamma}} v$. Also since $\neg\beta \in v$ by induction $\hat{M}, v \models \neg\beta$.

For $0 \in \hat{\delta}(w)$, let $\top$ be any validity. By ([$\exists$]Nec) we have $\vdash_{\mathcal{AX}_{\mathcal{A}}} [\exists]\top$ and hence $[\exists]\top \in w$. Again, by Lemma 5.26(2), $\Gamma = \{\neg\beta, \top\} \cup \{\psi \mid [\forall]\psi \in w\}$ is consistent. Hence there is some $v \in \hat{\mathcal{W}}$ such that $\Gamma \subseteq v$. And thus $w \xrightarrow{0} v$ and by induction hypothesis, $\hat{M}, v \models \neg\beta$.

Thus for every $a \in \gamma(w)$ there is some v such that $w \xrightarrow{a} v$ and $\hat{\mathcal{M}}, v \models \neg\beta$. Hence $\hat{\mathcal{M}}, w \models \langle\forall\rangle\neg\beta$.

For the other direction, suppose $[\exists]\beta \in \hat{\delta}(w)$ then by definition of the canonical model we have for any $w \xrightarrow{i_{[\exists]\beta}} u$ it is always the case that $\beta \in u$. By induction hypothesis, for any $w \xrightarrow{i_{[\exists]\beta}} u$ we have $\hat{\mathcal{M}}, u \models \beta$. Hence $\hat{\mathcal{M}}, w \models [\exists]\beta$. □

Corollary 5.30. *Satisfiability problem for IQML is decidable.*

In the next chapter, we will discuss bisimulation for IQML and compare the expressiveness of IQML to the 1-variable fragment of PTML.

5.7 Discussion

We have proved that the two variable fragment of PTML^2 (and hence TML^2) is decidable. The upper bound shown is in 2-EXPSPACE. A NEXPTIME lower bound follows since FO^2 satisfiability can be reduced to PTML^2 satisfiability. We believe that by careful management of the normal form, space can be reused and the upper bound can in fact be brought down by one exponent. That would still leave a significant gap between lower and upper bounds to be addressed in future work.

If we have even a single constant $\mathbf{c}$ in the vocabulary then the 2-variable fragment of TML becomes undecidable. This is because we can translate 2-variable FOML formulas into 2-variable TML formulas by using the constant $\mathbf{c}$ to index the modalities. Formally, $\Box\alpha$ is translated to $\Box_{\mathbf{c}}\alpha$. This translation preserves satisfiability and hence TML^2 (by Lemma 5.8, PTML^2) extended with a single constant is undecidable.

However, the story is unclear when we have equality in the syntax. Note that the proof strategy for PTML^2 cannot be directly used when equality is involved.

In particular, we can no longer use model extension (Def.5.15 and Lemma 5.16) since equality might restrict the number of agents at every world. The status of decidability of $\mathsf{TML}^2_=$ is currently open.

For the bundled fragment, the positive decidability results follow from translating bundled TML formulas to bundled FOML formulas. On the other hand, if the translation takes us to undecidable fragments then the status is not clear. In particular, the bundled fragment of the form $\forall x \Box_x$ is over constant agent models is open.

For first order modal logic itself there are other natural bundles to consider: $\Box\exists$ and $\Box\forall$ and a sequence of quantifiers followed by a modality, but these fragments do not make sense for TML since the variable quantification should occur before it appears as modal index.

The fragments discussed in this thesis are motivated by syntactic restrictions. There have been some investigations in the literature towards identifying some decidable fragments of term modal logic motivated by semantic restrictions. These fragments arise from their interest in the epistemic logic to model the notion of 'everyone knows' and 'someone knows' and community knowledge[6] (see Grove and Halpern [GH93, Gro95]). Orlandelli and Corsi [OC17] consider two such decidable fragments:

1. Atoms are restricted to propositions and quantifiers, modality occurrence is restricted to the form: $\exists x \Box_x \alpha$ (and $\forall x \Diamond_x \alpha$ dually). This fragment corresponds to the $[\exists]$ operator of IQML that we have discussed.
2. Quantifiers appear in a restricted guarded form: $\forall x(P(x) \rightarrow \Box_x \alpha)$ and $\exists x(P(x) \wedge \Box_x \alpha)$ (and their duals) where atoms in α are restricted to propositions.

[6]Example: All lawyers know that everybody who has signed in the contract knows that they are guilty : $\forall x(Lawyer(x) \rightarrow \Box_x(\forall y\ Sign(y) \rightarrow \Box_y\ (Guilty(y))$.

They prove that the satisfiability is in PSPACE for both the fragments. The PSPACE decidability for (1) is proved via translation into classical propositional modal logic and a tableau proceduce again gives a PSPACE procedure for (2) [OC17].

Shtakser ([Sht18]) considers a second order version of the restricted guards (with 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 at every world w as $\rho(w, P) \subseteq 2^{2^{\mathcal{D}_w}}$. Also, $\Box_X \alpha$ is true at w if for every $c \in X$, $\Box_c \alpha$ is true at w. This fragment is proved to be in PSPACE via translation into the loosely guarded fragment of first order logic [Sht18].

Table 5.3 gives a summary of various fragments of term modal logic and its decidability status.

Syntax Restriction	Predicate Restriction	Decidability	Remark
$\mathsf{PTML}^{\top}$	$(\top, \bot)$	✗	implies TML^3 is undec.
$\mathsf{TML}^3_=$	only =	✗	mutually recursively inseparable
Monodic	unary	✓	translation to monodic FOML [WZ01]
Bundled	No restriction	✓	translation to bundled FOML [PRW18]
TML^2	No restriction	✓	bounded agent property
PTML^2 +1 const.	Proposition	✗	translation from 2-var FOML
$\mathsf{TML}^2_=$	with =	open	
Guarded PTML	Propositions	✓	[OC17], [Sht18]

Table 5.3: Summary of satisfiability problem for fragments of TML.

Chapter 6

Expressivity

The notion of bisimulation plays a crucial role in understanding the relationship between the structural properties of the model and the semantics of the logic (see book [BdRV01]). In general, bisimulation induces an equivalence relation over the *pointed structures* such that all models belonging to the same equivalence class satisfy the same set of formulas. In other words, there is no formula in the logic that can distinguish two models that are *bisimilar*. For more details on bisimulation for ML, refer [BdRV01, GO07, Ben10b] and refer [Ben10a, Wan17] for bisimulations for first order modal logic. The notion of bisimulation is akin to the notion of *partial isomorphism* studied for first order logic [PG92].

The notion of bisimulation for PTML is along the same lines as that of the bisimulation for first order modal logic [Ben10a] with obvious adjustments to suit term modal logic. However, since FOML also has predicates, the bisimulation needs to incorporate the notion of *Partial isomorphism* associated with FO. On the other hand, PTML has only propositions as atoms with quantified modalities and this leads to a slightly different way to define the notion of bisimulation. We take this up in detail in this chapter.

6.1 Bisimulation for PTML

Note that in ML^n, the truth of a formula at a world in a model depends on the valuation function at that world and the truth of subformulas in the successor worlds in the model. On the other hand, the truth of an FO formula in a model depends on the truth of subformulas in the model along various extensions of the interpretation function. In our logic, we have both, and hence bisimilarity of PTML is given by a pair (G, H) where G captures the bisimulation for worlds (corresponding to the ML aspect of PTML) and H captures the bisimulation for agents (corresponding to the FO aspect of PTML).

Definition 6.1 (PTML bisimulation)**.** *Let $\mathcal{M}_1 = (\mathcal{W}_1, \mathcal{D}_1, \delta_1, \mathcal{R}_1, \rho_1)$ and $\mathcal{M}_2 = (\mathcal{W}_2, \mathcal{D}_2, \delta_2, \mathcal{R}_2, \rho_2)$ be two PTML models. Let $\mathrm{G} \subseteq (\mathcal{W}_1 \times \mathcal{W}_2)$ be a non-empty binary relation over worlds and $\mathrm{H} = \{\mathrm{H}_{(w_1,w_2)} \mid (w_1, w_2) \in G\}$ where for every $(w_1, w_2) \in \mathrm{G}$, $\mathrm{H}_{(w_1,w_2)} \subseteq (\delta_1(w_1) \times \delta_2(w_2))$ is a non-empty binary relation over agents. The pair (G, H) is called a* bisimulation *if for all $(w_1, w_2) \in \mathrm{G}$ the following conditions hold:*

val: $\rho_1(w_1) = \rho_2(w_2)$.

agent-forth: *For all $d_1 \in \delta_1(w_1)$ there is some $d_2 \in \delta_2(w_2)$ such that $(d_1, d_2) \in \mathrm{H}_{(w_1,w_2)}$.*

agent-back: *For all $d_2 \in \delta_2(w_2)$ there is some $d_1 \in \delta_1(w_1)$ such that $(d_1, d_2) \in \mathrm{H}_{(w_1,w_2)}$.*

For all $(d_1, d_2) \in \mathrm{H}_{(w_1,w_2)}$

world-forth: *for all $u_1 \in \mathcal{W}_1$ if $w_1 \xrightarrow{d_1} u_1 \in \mathcal{R}_1$ there exists some $u_2 \in \mathcal{W}_2$ such that $w_2 \xrightarrow{d_2} u_2 \in \mathcal{R}_2$ and $(u_1, u_2) \in \mathrm{G}$ and $\mathrm{H}_{(w_1,w_2)} \subseteq \mathrm{H}_{(u_1,u_2)}$.*

world-back: *for all $u_2 \in \mathcal{W}_2$ if $w_2 \xrightarrow{d_2} u_2 \in \mathcal{R}_2$ there exists $u_1 \in \mathcal{W}_1$ such that $w_1 \xrightarrow{d_1} u_1 \in \mathcal{R}_1$ and $(u_1, u_2) \in \mathrm{G}$ and $\mathrm{H}_{(w_1,w_2)} \subseteq \mathrm{H}_{(u_1,u_2)}$.*

In the bisimulation tuple (G, H), G relates the *bisimilar worlds*. Now since the set of local agents differs at each world, for every bisimilar world pair $(w_1, w_2) \in \mathrm{G}$, we need a *bisimulation over agents* and that is captured by $\mathrm{H}_{(w_1,w_2)}$. Accordingly, G is called world bisimulation and H is called agent bisimulation.

The first condition in the Def. 6.1 says that the bisimilar worlds agree on valuation of propositions. Agent back and forth properties state that for every bisimilar world pair, every agent in one world can be mapped to some agent in the other. The last condition is the analogue of the back and forth condition of bisimulation of ML which should hold for all the bisimilar agent pairs. Note that along with the standard back and forth conditions there is an additional constraint $\mathrm{H}_{(w_1,w_2)} \subseteq \mathrm{H}_{(u_1,u_2)}$. This ensures that bisimilar agent pairs continue to be bisimilar in the successor worlds, corresponding to the monotonicity property of δ.

Given two PTML models $\mathcal{M}_1$ and $\mathcal{M}_2$ with $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$ and for all $n \geq 0$ and for all $c_1, \ldots, c_n \in \delta_1(w_1)$ and $d_1, \ldots, d_n \in \delta_2(w_2)$ we say that $(\mathcal{M}_1, w_1, c_1, \ldots, c_n)$ is bisimilar to $(\mathcal{M}_2, w_2, d_1, \ldots, d_n)$ if there is some bisimulation (G, H) such that $(w_1, w_2) \in \mathrm{G}$ and $\{(c_i, d_i) \mid i \leq n\} \subseteq \mathrm{H}_{(w_1,w_2)}$. We denote this by $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \rightleftharpoons (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$. If $n = 0$, we have $(\mathcal{M}_1, w_1) \rightleftharpoons (\mathcal{M}_2, w_2)$ which does not impose any condition on $H_{(w_1,w_2)}$.

Bisimulation is the semantic counterpart of the classical notion of elementary equivalence for modal logics which preserve formulas.

Definition 6.2 (Elementary equivalence). *Let* $\mathcal{M}_1 = (\mathcal{W}_1, \mathcal{D}_1, \delta_1, \mathcal{R}_1, \rho_1)$ *and* $\mathcal{M}_2 = (\mathcal{W}_2, \mathcal{D}_2, \delta_2, \mathcal{R}_2, \rho_2)$ *be two PTML models. For all* $w_1 \in \mathcal{W}_1$ *and* $w_2 \in \mathcal{W}_2$*, let* $n \geq 0$ *and for all* $c_1, \ldots, c_n \in \delta_1(w_1)$ *and* $d_1, \ldots, d_n \in \delta_2(w_2)$ *we say that* $(\mathcal{M}_1, w_1, c_1, \ldots, c_n)$ is elementarily equivalent to $(\mathcal{M}_2, w_2, d_1, \ldots, d_n)$ *if for all formula PTML formulas* $\varphi(x_1, \ldots, x_n)$ *we have* $\mathcal{M}_1, w_1 \models \varphi(c_1, \ldots, c_n)$ *iff* $\mathcal{M}_2, w_2 \models \varphi(d_1, \ldots, d_n)$.

We denote this by $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \equiv (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$.

If $n = 0$, we have $(\mathcal{M}_1, w_1) \equiv (\mathcal{M}_2, w_2)$ which is the same as saying for all PTML sentence φ we have $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$.

As in ML^n, we now prove that bisimulation implies elementary equivalence for PTML.

Theorem 6.3 (bisimulation preserves elementary equivalence)**.** *Given two PTML models $\mathcal{M}_1 = (\mathcal{W}_1, \mathcal{D}_1, \delta_1, \mathcal{R}_1, \rho_1)$ and $\mathcal{M}_2 = (\mathcal{W}_2, \mathcal{D}_2, \delta_2, \mathcal{R}_2, \rho_2)$, for all $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$ and for all $c_1, \ldots, c_n \in \delta_1(w_1)$ and $d_1, \ldots, d_n \in \delta_2(w_2)$:*

if $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \rightleftharpoons (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$ then

$(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \equiv (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$.

Proof. We prove this for all models and for all bisimilar pairs simultaneously by induction on the structure of the formula.

In the base case we have propositions and the claim follows by condition 1. The $\neg$ and $\wedge$ cases follow by routine applications of the induction hypothesis.

For the case $\Diamond_{x_j}\psi$, pick any $\mathcal{M}_1, \mathcal{M}_2$ with $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$ and pick any $c_1, \ldots, c_n \in \delta_1(w_1)$ and $d_1, \ldots, d_n \in \delta_2(w_2)$ such that $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \rightleftharpoons (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$. Let (G, H) be the corresponding bisimulation which implies $(w_1, w_2) \in \mathrm{G}$ and $\{(c_i, d_i) \mid i \leq n\} \subseteq \mathrm{H}_{(w_1,w_2)}$. We need to prove that $\mathcal{M}_1, w_1 \models \Diamond_{c_j}\psi(c_1, \ldots, c_n)$ iff $\mathcal{M}_2, w_2 \models \Diamond_{d_j}\psi(d_1, \ldots, d_n)$.

Suppose $\mathcal{M}_1, w_1 \models \Diamond_{c_j}\psi(c_1, \ldots, c_n)$. Then by semantics there is some $u_1 \in \mathcal{W}_1$ such that $w_1 \xrightarrow{c_j} u_1 \in \mathcal{R}_1$ and $\mathcal{M}_1, u_1 \models \psi(c_1, \ldots, c_n)$. By condition *world-forth*, there is some $u_2 \in \mathcal{W}_2$ such that $w_2 \xrightarrow{d_j} u_2 \in \mathcal{R}_2$ and $(u_1, u_2) \in \mathrm{G}$ and $\mathrm{H}_{(w_1,w_2)} \subseteq \mathrm{H}_{(u_1,u_2)}$. Now since $\{(c_i, d_i) \mid i \leq n\} \subseteq \mathrm{H}_{(w_1,w_2)}$, we also have $\{(c_i, d_i) \mid j \leq n\} \subseteq \mathrm{H}_{(u_1,u_2)}$. Thus $(\mathcal{M}_1, u_1, c_1, \ldots, c_n) \rightleftharpoons (\mathcal{M}_2, u_2, d_1, \ldots, d_n)$ and by induction hypothesis we have $\mathcal{M}_2, u_2 \models \psi(d_1, \ldots, d_n)$. Hence $\mathcal{M}_2, w_2 \models \Diamond_{d_j}\psi(d_1, \ldots, d_n)$. The proof of the other direction is symmetric using *world-back* condition .

For the case $\varphi = \exists y\ \psi(x_1, \ldots, x_n, y)$, again pick any $\mathcal{M}_1, \mathcal{M}_2$ with $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$ and $c_1, \ldots, c_n \in \delta_1(w_1)$ and $d_1, \ldots, d_n \in \delta_2(w_2)$ such that $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \rightleftharpoons (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$.

Let (G, H) be the corresponding bisimulation which implies $(w_1, w_2) \in \mathrm{G}$ and $\{(c_i, d_i) \mid i \leq n\} \subseteq \mathrm{H}_{(w_1,w_2)}$. We need to prove that $\mathcal{M}_1, w_1 \models \exists y\ \psi(c_1, \ldots, c_n, y)$ iff $\mathcal{M}_2, w_2 \models \exists y\ \psi(d_1, \ldots, d_n, y)$

Suppose $\mathcal{M}_1, w_1 \models \exists y\ \psi(c_1, \ldots, c_n, y)$. By semantics, there is some $\mathbf{c} \in \delta_1(w_1)$ such that $\mathcal{M}_1, w_1 \models \psi(c_1, \ldots, c_n, \mathbf{c})$. Now by *agent-forth* condition, there exists some $\mathbf{d} \in \delta_2(w_2)$ such that $(\mathbf{c}, \mathbf{d}) \in \mathrm{H}_{(w_1,w_2)}$ and thus we have $(\mathcal{M}_1, w_1, c_1, \ldots, c_n, \mathbf{c}) \rightleftharpoons (\mathcal{M}_2, w_2, d_1, \ldots, d_n, \mathbf{d})$.

Now by induction hypothesis, $\mathcal{M}_2, w_2 \models \psi(d_1, \ldots, d_n, \mathbf{d})$ and hence $\mathcal{M}_2, w_2 \models \exists y\ \psi(d_1, \ldots, d_n, y)$. The proof of the other direction is symmetric, using the condition *agent-back*.

□

In ML^n, the converse holds for image-finite models. For PTML, this notion is name-specific.

Definition 6.4 (Image finite models)**.** *A PTML model $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ is* image-finite *if for all $u \in \mathcal{W}$, $\delta(u)$ is finite and for all $d \in \delta(u)$, the set $\{v \mid u \xrightarrow{d} v\}$ is finite.*

Thus image-finite models have finitely many names associated with every world and is finitely branching on each name. For classical propositional modal logic (ML^n), bisimilarity over image-finite models [1] coincides with formula preservation. As one may expect, for PTML also, formula preservation characterizes bisimilarity over image finite models.

[1]For ML^n, an image finite model corresponds to having finitely many successors at every world for every $i \in \mathsf{Ag}$.

Theorem 6.5. *Let $\mathcal{M}_1 = (\mathcal{W}_1, \mathcal{D}_1, \delta_1, \mathcal{R}_1, \rho_1)$ and $\mathcal{M}_2 = (\mathcal{W}_2, \mathcal{D}_2, \delta_2, \mathcal{R}_2, \rho_2)$ be two image-finite* PTML *models. For all $w_1 \in \mathcal{W}_1$ and for all $w_2 \in \mathcal{W}_2$ and for all $c_1, \ldots, c_n \in \delta_1(w_1)$ and $d_1, \ldots, d_n \in \delta_2(w_2)$ we have*

$(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \rightleftharpoons (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$ *iff* $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \equiv (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$.

Proof. ($\Rightarrow$) follows from Theorem 6.3. Thus it is sufficient to prove ($\Leftarrow$).

Define $\mathrm{G} = \{(u_1, u_2) \mid$ for all PTML sentence φ we have $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi\}$. For every $(u_1, u_2) \in \mathrm{G}$ define $\mathrm{H}_{(u_1,u_2)}$ to be the smallest set such that and for all $n \geq 1$ and $c_1, \ldots, c_n \in \delta_1(u_1)$ and $d_1, \ldots, d_n \in \delta_2(u_2)$ if $(\mathcal{M}_1, u_1, c_1, \ldots, c_n) \equiv (\mathcal{M}_2, u_2, d_1, \cdots d_n)$ then $\{(c_i, d_i) \mid i \leq n\} \subseteq H_{(u_1,u_2)}$.

It is sufficient to show that (G, H) is indeed a bisimulation. For this, we verify all the conditions. Pick any $(u_1, u_2) \in \mathrm{G}$.

Since u_1 and u_2 satisfy the same sentences, in particular they agree on propositions, Hence condition 1 holds.

For condition *agent-forth*, suppose it does not hold[2]. Then there is some $\mathbf{c} \in \delta_1(u_1)$ such that for all $d \in \delta_2(u_2)$ we have $(\mathcal{M}_1, u_1, \mathbf{c}) \not\equiv (\mathcal{M}_2, u_2, d)$. Since $\mathcal{M}_2$ is image-finite, let $\delta_2(u_2) = \{d_1, \ldots d_m\}$. Thus, for every $d_i \in \delta_2(u_2)$ we have a formula $\varphi_i(x)$ such that $\mathcal{M}_1, u_1 \models \varphi_i(\mathbf{c})$ and $\mathcal{M}_2, u_2 \models \neg\varphi_i(d_i)$. This implies $\mathcal{M}_1, u_1 \models \exists x\ (\bigwedge\limits_{i \leq m} \varphi_i)$ and $\mathcal{M}_2, u_2 \models \forall x\ (\bigvee\limits_{i \leq m} \neg\varphi_i)$. This contradicts the assumption that $(u_1, u_2) \in \mathrm{G}$.

Condition *agent-back* is proved analogously using the fact that $\mathcal{M}_1$ is image-finite.

For condition *world-forth*, suppose it does not hold. Then there is some $(\mathbf{c}, \mathbf{d}) \in H_{(u_1,u_2)}$ and some $u_1 \xrightarrow{\mathbf{c}} \mathbf{u}' \in \mathcal{R}_1$ for which the condition fails.

[2]Note that $\mathbf{c}, d_i$ are not in the syntax and they are not needed.

Now since $(\mathbf{c}, \mathbf{d}) \in H_{(u_1,u_2)}$ it means there is some $i, n \geq 0$ and some live agents $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots c_n \in \delta_1(u_1)$ and $d_1, \ldots, d_{i-1}, d_{i+1}, \ldots d_n \in \delta_2(u_2)$ such that $(\mathcal{M}_1, u_1, c_1, \ldots, c_{i-1}, \mathbf{c}, c_{i+1}, \ldots c_n) \equiv (\mathcal{M}_2, u_2, d_1, \ldots, d_{i-1}, \mathbf{d}, d_{i+1}, \ldots d_n)$.

Let $S(\mathbf{d}) = \{v \mid u_2 \xrightarrow{d} v \in \mathcal{R}_2\}$. Note that $S(\mathbf{d})$ is non-empty: otherwise, $\mathcal{M}_1, u_1, [x_i \mapsto c] \models \Diamond_{x_i}\top$ and $\mathcal{M}_2, u_2, [x_i \mapsto d] \models \Box_{x_i}\bot$ and this contradicts $(\mathcal{M}_1, u_1, c_1, \ldots, c_{i-1}, \mathbf{c}, c_{i+1}, \ldots c_n) \equiv (\mathcal{M}_2, u_2, d_1, \ldots, d_{i-1}, \mathbf{d}, d_{i+1}, \ldots d_n)$.

The model $\mathcal{M}_2$ is image-finite and hence let $S(\mathbf{d}) = \{v_1, \ldots v_k\}$. Define $S_1(\mathbf{d}) = \{v_j \mid v_j \in S(\mathbf{d}) \text{ and } (\mathbf{u}', v_j) \in \mathrm{G}\}$ and $S_2(\mathbf{d}) = \{v_j \mid v_j \in S(\mathbf{d}) \text{ and } (\mathbf{u}', v_j) \notin \mathrm{G}\}$. First we observe that $S_1(\mathbf{d})$ is non empty. If not, then for all $v_j \in S(\mathbf{d})$ there is some sentence ψ_j such that $\mathcal{M}_1, \mathbf{u}' \models \psi_j$ and $\mathcal{M}_2, v_i \models \neg\psi_j$. Thus we have $\mathcal{M}_1, u_1, [x_i \mapsto \mathbf{c}] \models \Diamond_{x_i}(\bigwedge_{j=1}^{k} \psi_j)$ and $\mathcal{M}_2, u_2, [x_i \mapsto \mathbf{d}] \models \Box_{x_i}(\bigvee_{j=1}^{k} \neg\psi_j)$ which contradicts $(\mathcal{M}_1, u_1, c_1, \ldots, c_{i-1}, c, c_{i+1}, \ldots c_n) \equiv (\mathcal{M}_2, u_2, d_1, \ldots, d_{i-1}, d, d_{i+1}, \ldots d_n)$.

Hence $S_1(\mathbf{d})$ is non-empty. Let $S_1(\mathbf{d}) = \{v'_1, \ldots v'_{k'}\}$. Now if there is some $\mathbf{v} \in S_1(\mathbf{d})$ such that $\mathrm{H}_{(u_1,u_2)} \subseteq \mathrm{H}_{(\mathbf{u}',\mathbf{v})}$ then we are done. Suppose not, then for every $v'_j \in S_1(\mathbf{d})$, there is some $(e^j, f^j) \in H_{(u_1,u_2)}$ and $(e^j, f^j) \notin \mathrm{H}_{(\mathbf{u}',v_j)}$. In particular, we have $(\mathcal{M}_1, \mathbf{u}', c_1, \ldots, c_{i-1}, c, c_{i+1}, \ldots c_n, e^j) \not\equiv (\mathcal{M}_2, v_j, d_1, \ldots, d_{i-1}, d, d_{i+1}, \ldots d_n, f^j)$. Hence for every $v_j \in S_1(\mathbf{d})$ there is some PTML formula $\alpha_j(x_1, \cdots, x_n, y_j)$ such that $\mathcal{M}_1, \mathbf{u}' \models \alpha_j(c_{i-1}, c, c_{i+1}, \ldots c_n, e^j)$ and $\mathcal{M}_2, v_j \models \neg\alpha_j(d_1, \ldots, d_{i-1}, d, d_{i+1}, \ldots d_n, f^j)$. Recall that for all $v_j \in S_2(\mathbf{d})$ there is a sentence φ_j such that $\mathcal{M}_1, \mathbf{u}' \models \varphi_j$ and $\mathcal{M}_2, v_j \models \neg\varphi_j$.

Thus we have $\mathcal{M}_1, u_1, [x_1 \mapsto c_1, \ldots x_n \mapsto c_n] \models (\bigwedge_{v_j \in S_1(\mathbf{d})} \Diamond_{x_i}\alpha_j \wedge \bigwedge_{v_j \in S_2(\mathbf{d})} \Diamond_{x_i}\varphi_j)$ and $\mathcal{M}_2, u_2, [x_1 \mapsto d_1, \ldots x_n \mapsto d_n] \models \Box_{x_i}(\bigvee_{v_j \in S_1(\mathbf{d})} \neg\alpha_j \vee \bigvee_{v_j \in S_2(\mathbf{d})} \neg\varphi_j)$. This contradicts $(\mathcal{M}_1, u_1, c_1, \ldots, c_{i-1}, c, c_{i+1}, \ldots c_n) \equiv (\mathcal{M}_2, u_2, d_1, \ldots, d_{i-1}, d, d_{i+1}, \ldots d_n)$.

Condition *agent-back* is argued symmetrically.

$\square$

As an application of bisimulation, we prove that PTML has the tree model property: every satisfiable formula has a tree model. For this, we prove that for every PTML model $\mathcal{M}$ and every $w \in W$, is bisimilar to some tree model. The canonical tree model corresponding to $(\mathcal{M}, w)$ is the standard unravelling of the model $\mathcal{M}$ starting from w.

For any PTML model $\mathcal{M}$ an $\mathcal{R}$-path is a sequence of the from $w_0 d_1 w_1 d_2 \ldots d_n w_n$ where $w_i \in \mathcal{W}$ and $d_j \in \mathcal{D}$ such that $(w_i, d_{i+1}, w_{i+1}) \in \mathcal{R}$.

Definition 6.6 (Tree models)**.** *A* tree model *for PTML is given by* $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ *rooted at* $r \in W$ *such that for every* $w \in \mathcal{W}$*, there is a unique* $\mathcal{R}$*-path from* r *to* w*.*

Theorem 6.7. *For any PTML formula* φ *we have* φ *is satisfiable iff* φ *is satisfiable in a tree model.*

Proof. It is enough to prove ($\Rightarrow$). Let φ be a PTML formula and let $\mathcal{M}$ be a model such that $\mathcal{M}, r, \sigma \models \varphi$ where $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$.

Let $\Pi = \Pi' \cup \{\lambda\}$ where Π' denotes the set of all $\mathcal{R}$-paths in $\mathcal{M}$ starting from r and λ is the empty path. The tree unravelling of the model starting from r is given by $\mathcal{M}' = (\mathcal{W}', \mathcal{D}, \delta', \mathcal{R}', \rho')$ where:

- $\mathcal{W}' = \Pi'$.
- For all $\pi u \in \mathcal{W}'$ define $\delta'(\pi u) = \delta(u)$.
- $\mathcal{R}' = \{(\pi_1, d, \pi_2) \mid \pi_1 = \pi u, \pi_2 = \pi u d v \text{ for some } \pi \in \Pi\}$.
- For all $\pi u \in \mathcal{W}'$ define $\rho'(\pi u) = \rho(u)$.

Define the bisimulation (G, H) where $\mathrm{G} = \{(u, \pi u) \mid u \in \mathcal{W} \text{ and } \pi u \in \mathcal{W}'\}$ and for every $(u, \pi u) \in \mathrm{G}$ define $H_{(u,\pi u)} = \{(d, d) \mid d \in \delta(u)\}$.

Clearly condition $[Val]$ holds since $\rho(u) = \rho'(\pi u)$ and *agent forth-back* holds since $\delta(u) = \delta'(\pi u)$ and $\mathrm{H}_{(u,\pi u)}$ is identity relation. For *world forth-back* if $(w, d, u) \in \mathcal{R}$ then $(\pi w, d, \pi w d u) \in \mathcal{R}'$ which satisfies the required conditions. On the other hand, if $(\pi w, d, \pi w d u) \in \mathcal{R}'$ then by construction we have $(w, d, u) \in \mathcal{R}$ which satisfies the required conditions.

Thus by Theorem 6.3, since $\mathcal{M}, r, \sigma \models \varphi$ we have $\mathcal{M}', r, \sigma \models \varphi$.

□

Like in ML^n, we can coarsen the definition of bisimulation to *k-bisimilarity* where condition 3 is modified appropriately. This is useful to characterize the set of models that satisfy the same set of formulas upto modal depth k.

A sequence of pairs $\{(\mathrm{G}^j, \mathrm{H}^j) \mid 0 \leq j \leq k\}$ called a *k-bisimulation* where every $\mathrm{G}^j \subseteq (\mathcal{W}_1 \times \mathcal{W}_2)$ is a non-empty binary relation over worlds and $\mathrm{H}^j = \{\mathrm{H}^j_{(w_1,w_2)} \mid (w_1, w_2) \in G^j\}$ where each $\mathrm{H}^j_{(w_1,w_2)} \subseteq (\delta_1(w_1) \times \delta_2(w_2))$. The conditions are modified as follows:

For all $(w_1, w_2) \in G^j$ the conditions of valuation (1) and agent back and forth property (2 a,b) remain the same. The world back and forth are specialized to respect the modal depth. For instance, if $j > 0$ then the forth property (condition 3a) is defined by: for all $u_1 \in \mathcal{W}_1$ if $w_1 \xrightarrow{d_1} u_1 \in \mathcal{R}_1$ there exists some $u_2 \in \mathcal{W}_2$ such that $w_2 \xrightarrow{d_2} u_2 \in \mathcal{R}_2$ and $(u_1, u_2) \in \mathrm{G}^{j-1}$ and $\mathrm{H}^j_{(w_1,w_2)} \subseteq \mathrm{H}^{j-1}_{(u_1,u_2)}$. Similarly the back condition (3b) is also modified.

For any two PTML models $\mathcal{M}_1$ and $\mathcal{M}_2$ and two worlds w_1 and w_2 in $\mathcal{W}_1$ and $\mathcal{W}_2$ respectively and for all $c_1, \ldots c_n \in \delta_1(w_1)$ and $d_1, \ldots, c_n \in \delta_2(w_2)$, we denote $(\mathcal{M}_1, w_1, c_1, \ldots c_n) \rightleftharpoons_k (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$ if there is some k-bisimulation $\{(\mathrm{G}^j, \mathrm{H}^j) \mid j \leq k\}$ such that $(w_1, w_2) \in \mathrm{G}^k$ and $\{(c_l, d_l) \mid l \leq n\} \subseteq \mathrm{H}^k_{(w_1,w_2)}$.

Similarly, we can also specialize elementary equivalence by modal depth. Define $(M_1, w_1, c_1 \ldots, c_n) \equiv_k (M_2, w_2, d_1, \ldots, d_n)$ if for every formula $\varphi(x_1, \ldots, x_n)$ with modal depth at most k, we have: $M_1, w_1 \models \varphi(c_1, \ldots, c_n)$ iff $M_2, w_2 \models \varphi(d_1, \ldots, d_n)$.

Lemma 6.8. *Let $\mathcal{M}_1 = (\mathcal{W}_1, \mathcal{D}_1, \delta_1, \mathcal{R}_1, \rho_1)$ and $\mathcal{M}_2 = (\mathcal{W}_2, \mathcal{D}_2, \delta_2, \mathcal{R}_2, \rho_2)$ be two PTML models. For all $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$ and for all $c_1, \ldots, c_n \in \delta_1(w_1)$ and $d_1, \ldots, d_n \in \delta_2(w_2)$. If $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \rightleftharpoons_k (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$ then $(\mathcal{M}_1, w_1, c_1, \ldots, c_n) \equiv_k (\mathcal{M}_2, w_2, d_1, \ldots, d_n)$.*

The proof of the lemma follows along the same lines as the proof of Theorem 6.3. Thus, we have the following theorem.

Theorem 6.9. *For any PTML formula φ of modal depth h, φ is satisfiable iff φ is satisfiable in a tree model of depth at most h.*

6.1.1 Deciding PTML bisimulation

Having defined the notion of bisimulation, a natural algorithmic question arises: Given two finite PTML models $\mathcal{M}_1$ and $\mathcal{M}_2$ with $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$, decide whether $(\mathcal{M}_1, w_1) \rightleftharpoons (\mathcal{M}_2, w_2)$ or not. We assume that the underlying set of propositions is finite, so that the valuation function ρ specified in the input is finite. Algorithm 1 describes the procedure to decide PTML bisimulation.[3] The procedure is similar to the one used for ML^n [BdRV01].

To see that the algorithm terminates, note that initially $|\mathrm{G}| + |\mathrm{H}| \leq (|\mathcal{W}_1| \cdot |W_2| \cdot |\mathcal{D}_1| \cdot |\mathcal{D}_2|)$. Since the algorithm reduces the size of $|\mathrm{G}| + |\mathrm{H}|$ by at least 1 in every iteration and every condition can be checked in polynomial time, the algorithm terminates in polynomial time.

[3]This kind of refinement technique with obvious adjustments also works to decide bisimulation for first order modal logic without equality.

algorithm 1 Deciding PTML bisimulation

$G \leftarrow \{(u_1, u_2) \mid \rho_1(u_1) = \rho_2(u_2)\}$; $H \leftarrow \{H_{(u_1,u_2)} \mid (u_1, u_2) \in G\}$;
where each $H_{(u_1,u_2)} \leftarrow (\delta_1(u_1) \times \delta_2(u_2))$
repeat
  if there is some $(u_1, u_2) \in G$ and $c \in \delta_1(u_1)$ such that there is no $(c, d) \in H_{(u_1,u_2)}$ **then**
    Remove (u_1, u_2) from G and Remove $H_{(u_1,u_2)}$ from H
  end if
  if there is some $(u_1, u_2) \in G$ and $d \in \delta_2(u_2)$ such that there is no $(c, d) \in H_{(u_1,u_2)}$ **then**
    Remove (u_1, u_2) from G and Remove $H_{(u_1,u_2)}$ from H
  end if
  if there is some $(u_1, u_2) \in G$ and some $(c, d) \in H_{(u_1,u_2)}$ such that for some $u_1 \xrightarrow{c} v_1$ there is no $u_2 \xrightarrow{d} v_2$ such that $(v_1, v_2) \in G$ and $H_{(u_1,u_2)} \not\subseteq H_{(v_1,v_2)}$ **then**
    Remove (c, d) from $H_{(u_1,u_2)}$
  end if
  if there is some $(u_1, u_2) \in G$ and some $(c, d) \in H_{(u_1,u_2)}$ such that for some $u_2 \xrightarrow{d} v_2$ there is no $u_1 \xrightarrow{c} v_1$ such that $(v_1, v_2) \in G$ and $H_{(u_1,u_2)} \not\subseteq H_{(v_1,v_2)}$ **then**
    Remove (c, d) from $H_{(u_1,u_2)}$
  end if
until No more deletion is possible
return *yes* if $(w_1, w_2) \in G$ and *no* otherwise

To see that the algorithm is correct, if the algorithm returns *yes*, then it can be verified that the final (G, H) obtained by the algorithm satisfies all properties of bisimulation. On the other hand, if $(\mathcal{M}_1, w_1) \rightleftharpoons (\mathcal{M}_2, w_2)$ then let (G', H') be the corresponding bisimulation. If the algorithm returns *no* then consider the first member of G' or H' that is being removed by the algorithm. If this tuple is of the form (u_1, u_2) then it has to be removed in case 1 or 2. But this is not possible since (G', H') is a bisimulation and hence we can always find a corresponding (c, d) pair. Similarly if the removed tuple is of the form (c, d) then it would have been removed in case 3 or 4 and this again is not possible since we can always find an appropriate $(v_1, v_2) \in G' \subseteq G$ which satisfies the required conditions (and by assumption this pair is not yet removed).

Theorem 6.10. *Given two finite PTML models $\mathcal{M}_1$ and $\mathcal{M}_2$ with $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$ deciding whether $(\mathcal{M}_1, w_1) \rightleftharpoons (\mathcal{M}_2, w_2)$ is in* PTIME.

6.2 Bisimulation for $\mathsf{PTML}_=$

When we have equality in the language, the logic gains more expressivity. In particular, the formulas can count the number of agents at every world.

For any two PTML models $\mathcal{M}_1$ and $\mathcal{M}_2$ with w_1 and w_2 say $(\mathcal{M}_1, w_1)$ is *equaility-PTML bisimilar* to $(\mathcal{M}_2, w_2)$ if there is some bisimulation (G, H) such that $(w_1, w_2) \in \mathrm{G}$ and for all $(u_1, u_2) \in \mathrm{G}$ the corresponding $\mathrm{H}_{(u_1,u_2)}$ is a bijection.

With this additional condition, analogous to Theorem 6.3, we can show that equality-PTML bisimulation implies elementary equivalence over $\mathsf{PTML}_=$ formulas.

The only extra condition to verify is the case $x = y$. Let $\mathcal{M}_1, w_1, \sigma_1 \models (x = y)$ and suppose $(w_1, w_2) \in \mathrm{G}$ and $\{(\sigma_1(x), \sigma_2(x)), (\sigma_1(y), \sigma_2(y))\} \subseteq \mathrm{H}_{(w_1,w_2)}$. Now since $\mathrm{H}_{(w_1,w_2)}$ is a bijection and $\sigma_1(x) = \sigma_1(y)$ we also have $\sigma_2(x) = \sigma_2(y)$. Hence it follows that $\mathcal{M}_2, w_2, \sigma_2 \models (x = y)$.

Similarly we can argue that if $\mathcal{M}_2, w_2, \sigma_2 \models (x = y)$ then $\mathcal{M}_1, w_1, \sigma_1 \models (x = y)$.

Having equality makes it harder to decide whether the given models are bisimilar or not. First we note that NP suffices.

This is because we can guess (G, H) and check that it satisfies all the properties in polynomial time. Is the problem NP-hard? Perhaps not. We will prove that for *equality-PTML bisimulation* is as hard as the *graph isomorphism problem*. This hardness is because of the demand that $\mathrm{H}_{(u_1,u_2)}$ should be a bijection for every $(u_1, u_2) \in \mathrm{G}$.

Definition 6.11 (Graph isomorphism)**.** *Given two finite simple undirected graphs $\mathcal{G}_1 = (V_1, E_1)$ and $\mathcal{G}_2 = (V_2, E_2)$ we say that $\mathcal{G}_1$ is isomorphic to $\mathcal{G}_2$ if there is a bijective mapping $f : V_1 \mapsto V_2$ such that for all $a, b \in V_1$ we have $(a, b) \in E_1$ iff $(f(a), f(b)) \in E_2$.*

Theorem 6.12 (Babai [Bab16]). *Deciding whether two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic is in $O(2^{\log^k(n)})$ for some k where $n = |V_1| + |V_2|$.*

For the reduction, given two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ we construct the corresponding PTML tree models $\mathcal{M}_1$ and $\mathcal{M}_2$ (of polynomial size) rooted at r_1 and r_2 respectively such that $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic iff $(\mathcal{M}_1, r_1)$ is equality-bisimilar to (M_2, r_2).

Theorem 6.13. *Deciding whether two PTML models are equality-bisimilar is at least as hard as graph isomorphism problem.*

Proof. For any simple indirected graph $\mathcal{G} = (V, E)$ define the corresponding PTML model $\mathcal{M}_\mathcal{G} = (\mathcal{W}, V, \delta, \mathcal{R}, \rho)$ where $\mathcal{W} = \{r\} \cup \{u_{ab}, v_{ab} \mid a, b \in V \text{ and } (a, b) \in E\}$. For all $w \in \mathcal{W}$ define $\delta(w) = V$ and $\rho(w) = \emptyset$. Define $\mathcal{R} = \{(r, a, u_{ab}), (u_{ab}, b, v_{ab}) \mid u_{ab}, v_{ab} \in \mathcal{W}\}$. Clearly the size of $\mathcal{M}_\mathcal{G}$ is linear in the size of G.

Now given any two simple indirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the corresponding PTML models rooted at r_1 and r_2 respectively. We claim that $\mathcal{G}_1$ is isomorphic to $\mathcal{G}_2$ iff $(\mathcal{M}_1, r_1)$ is equality-bisimilar to $(\mathcal{M}_2, r_2)$.

To prove this, suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic, let $f : V_1 \mapsto V_2$ be the isomorphism mapping.

Define $\mathrm{G} = \{(r_1, r_2),\ (u_{ab}, u_{f(a)f(b)}),\ (v_{ab}, v_{f(a)f(b)}) \mid a, b \in V_1\}$. For every pair $(w, w') \in \mathrm{G}$ define $\mathrm{H}_{(w,w')} = \{(a, f(a)) \mid a \in V_1\}$. Since $f : V_1 \mapsto V_2$ is a bijection, every $\mathrm{H}_{(w,w')}$ is also a bijection. Now we verify all the conditions for bisimulation. Clearly condition *Val* holds since valuation function ρ is same at all worlds. Conditions *agent back and forth* hold since f is a bijection. Finally condition *world back and forth* holds since $r_1 \xrightarrow{a} u_{ab} \xrightarrow{b} v_{ab}$ iff $(a, b) \in E_1$ iff $(f(a), f(b)) \in E_2$ iff $r_2 \xrightarrow{f(a)} u_{f(a)f(b)} \xrightarrow{f(b)} v_{f(a)f(b)}$.

For the other direction, suppose that $(\mathcal{M}_1, r_1)$ is equality bisimilar to $(\mathcal{M}_2, r_2)$. Let (G, H) be the corresponding equality bisimulation. Define $f : V_1 \mapsto V_2$ where for all $a \in V_1$ we have $f(a) = a'$ such that $(a, a') \in \mathrm{H}_{(r_1,r_2)}$. Note that f is a bijection since $\mathrm{H}_{(r_1,r_2)}$ is a bijective mapping. To see that f satisfies isomorphism property, note that $(a, b) \in E_1$ iff $r_1 \xrightarrow{a} u_{ab} \xrightarrow{b} v_{ab}$ iff $r_2 \xrightarrow{f(a)} u_{f(a)f(b)} \xrightarrow{f(b)} v_{f(a)f(b)}$ iff $(f(a), f(b)) \in E_2$.

□

Note that there is a gap in the complexity. The upper bound for deciding equality-PTML is NP and the lower bound is graph isomorphism. We believe that this gap can be closed by reducing equality-PTML bisimulation to graph isomorphism problem. But the reduction seems to be more involved and is not taken up in this thesis. Further investigation is needed in this direction.

6.3 Bisimulation for IQML

In the previous chapter, we introduced the *implicitly quantified modal logic* (IQML) as the variable free fragment of PTML which has two modalities $[\forall]$ and $[\exists]$ which represents $\forall x \Box_x$ and $\exists x \Box_x$ respectively. Recall that an IQML model is given by $\mathcal{M} = (\mathcal{W}, \mathcal{I}, \mathbb{R}, \delta, \rho)$ where $\mathcal{I}$ is a non-empty countable index set and $\mathbb{R} = \{\mathcal{R}_i \mid i \in \mathcal{I}\}$ where each $\mathcal{R}_i \subseteq (\mathcal{W} \times \mathcal{W})$ and $\delta : W \mapsto 2^{\mathcal{I}}$ such that whenever $(w, u) \in \mathcal{R}_i$ we have $i \in \delta(w)$. Also, since we do not have variables in the syntax, recall that we do not impose the monotonicity condition on δ function.[4]

To simplify the notations, given any model $\mathcal{M}$, $w \in \mathcal{W}$ and a formula of the form $[\exists]\varphi$, if $\mathcal{M}, w \models [\exists]\varphi$ and $i \in \mathcal{I}$ is the corresponding witness then we write $\mathcal{M}, w \models \Box_i \varphi$ (similarly we have $\mathcal{M}, w \models \Diamond_i \varphi$ for $\langle\exists\rangle\varphi$).

[4]The monotonicity condition can be relaxed since we do not have variables in the syntax and hence the problem of evaluating free variables disappears.

Now we introduce the notion of bisimulation specialized for IQML. Note that since there are no explicit mention of agents in this logic, we can get away with dropping the H component of the bisimulation. All we need to ensure is that every index in one structure has a *corresponding index* in the other. The following definition of bisimulation formalizes the notion of 'corresponding index'.

Definition 6.14. *Given two IQML models $\mathcal{M}_1 = (\mathcal{W}_1, \mathcal{I}_1, \mathbb{R}_1, \delta_1, \rho_1)$ and $\mathcal{M}_2 = (\mathcal{W}_2, \mathcal{I}_2, \mathbb{R}_2, \delta_2, \rho_2)$, an IQML-bisimulation on them is a non-empty relation $\mathrm{G} \subseteq (\mathcal{W}_1 \times \mathcal{W}_2)$ such that for all $(w_1, w_2) \in \mathrm{G}$ the following conditions hold:*

Val. $\rho_1(w_1) = \rho_2(w_2)$.

$[\exists]$*forth.* *For all $i \in \delta_1(w_1)$ there is some $j \in \delta_2(w_2)$ such that for all u_2 if $w_2 \xrightarrow{j} u_2$ then there is some u_1 such that $w_1 \xrightarrow{i} u_1$ and $(u_1, u_2) \in \mathrm{G}$.*

$[\exists]$*back.* *For all $j \in \delta_1(w_1)$ there is some $i \in \delta_1(w_1)$ such that for all u_1 if $w_1 \xrightarrow{i} u_1$ then there is some u_2 such that $w_2 \xrightarrow{j} u_2$ and $(u_1, u_2) \in \mathrm{G}$.*

$\langle\exists\rangle$*forth.* *For all $i \in \delta_1(w_1)$ and for all u_1 if $w_1 \xrightarrow{i} u_1$ then there is some $j \in \delta_2(w_2)$ and some u_2 such that $w_2 \xrightarrow{j} u_2$ and $(u_1, u_2) \in \mathrm{G}$.*

$\langle\exists\rangle$*back.* *For all $j \in \delta_2(w_2)$ and for all u_2 if $w_2 \xrightarrow{j} u_2$ then there is some $i \in \delta_1(w_1)$ and some u_1 such that $w_1 \xrightarrow{i} u_1$ and $(u_1, u_2) \in \mathrm{G}$.*

Note that these conditions are different from typical back and forth conditions.

Given two models $\mathcal{M}_1$ and $\mathcal{M}_2$ we say that w_1, w_2 are IQML bisimilar if there is some IQML bisimulation G on the models such that $(w_1, w_2) \in \mathrm{G}$ and denote it $(\mathcal{M}_1, w_1) \sim (\mathcal{M}_2, w_2)$. The elementary equivalence for IQML is denoted by $(\mathcal{M}_1, w_1) \triangleq (\mathcal{M}_2, w_2)$ if for all $\varphi \in$ IQML, $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$.

Now we restate the theorems discussed in the previous section for IQML.

Theorem 6.15. *For any two IQML models $\mathcal{M}_1$ and $\mathcal{M}_2$ and any $w_1 \in \mathcal{W}_1$ and $w_2 \in \mathcal{W}_2$, if $(\mathcal{M}_1, w_1) \sim (\mathcal{M}_2, w_2)$ then $(\mathcal{M}_1, w_1) \triangleq (\mathcal{M}_2, w_2)$.*

Proof. Let G be an IQML bisimulation such that $(w_1, w_2) \in \mathrm{G}$. We can show that for all $(v_1, v_2) \in \mathrm{G}$ we have $\mathcal{M}_1, w_1 \models \varphi$ iff $\mathcal{M}_2, w_2 \models \varphi$, by induction on structure of φ. The theorem follows along standard lines. We highlight only the modal cases.

For the case $\varphi := [\exists]\psi$: Suppose $\mathcal{M}_1, v_1 \models [\exists]\psi$, we need to prove that $\mathcal{M}_2, v_2 \models [\exists]\psi$. Since $\mathcal{M}_1, v_1 \models [\exists]\psi$, there is some $i \in \mathcal{I}_1$ such that $\mathcal{M}_1, v_1 \models \Box_i\psi$. Now let $j \in \mathcal{I}_2$ be the witness for i for condition ([∃]forth). We claim that $\mathcal{M}_2, v_2 \models \Box_j\psi$. Suppose not; then $\mathcal{M}_2, v_2 \models \Diamond_j\neg\psi$ and hence there is some $v_2 \xrightarrow{j} u_2$ such that $\mathcal{M}_2, u_2 \not\models \psi$. Since j was the witness for i for ([∃]forth) condition, there is some $v_1 \xrightarrow{i} u_1$ such that $(u_1, u_2) \in \mathrm{G}$. By induction hypothesis, $\mathcal{M}_1, u_1 \not\models \psi$ which contradicts $\mathcal{M}_1, v_1 \models \Box_i\psi$. The other direction is proved symmetrically using ([∃]back) condition.

For the case $\langle\exists\rangle\psi$: Suppose $\mathcal{M}_1, v_1 \models \langle\exists\rangle\psi$ then there is some $i \in \mathcal{I}_1$ and some $u_1 \in W_1$ such that $v_1 \xrightarrow{i} u_1$ and $\mathcal{M}_1, u_1 \models \psi$. By condition (⟨∃⟩forth) there is some $j \in \mathcal{I}_2$ and some $v_2 \xrightarrow{j} u_2$ such that $(u_1, u_2) \in G$. By induction hypothesis $M_2, u_2 \models \psi$ and hence $\mathcal{M}_2, v_2 \models \langle\exists\rangle\psi$. The other direction is symmetrically argued using (⟨∃⟩back) condition.

□

Again, the converse holds over image finite models with finite index set ($\mathcal{I}$). An IQML model $\mathcal{M}$ is said to be image-finite if $\mathcal{I}$ is finite and $N^i(w) = \{u \mid (w, u) \in R_i\}$ is finite for all $w \in \mathcal{W}$ and $i \in \mathcal{I}$.

Theorem 6.16. *Suppose $\mathcal{M}_1$ and $\mathcal{M}_2$ are image-finite IQML models then $(\mathcal{M}_1, w_1) \sim (\mathcal{M}_2, w_2)$ iff $(\mathcal{M}_1, w_1) \triangleq (\mathcal{M}_2, w_2)$.*

Proof. (⇒) follows from Theorem 6.15. For (⇐), define $\mathrm{G} = \{(v_1, v_2) \mid \mathcal{M}_1, v_1 \triangleq \mathcal{M}_2, v_2\}$. It suffices to show that G is indeed an IQML bisimulation. For this, choose any $(v_1, v_2) \in G$. Clearly $[Val]$ holds since v_1, v_2 agree on all IQML propositions.

Now suppose that the ([∃]forth) condition does not hold. Let $\delta_2(v_2) = \{j_1 \cdots j_n\}$. If the condition does not hold, then there is some $\mathbf{i} \in \delta_1(v_1)$ such that for all $j_l \in \mathcal{I}_2$ there is some $u_l \in \mathcal{W}_2$ such that $v_2 \xrightarrow{j_l} u_l$ and for all $v_1 \xrightarrow{\mathbf{i}} w$ we have $(w, u_l) \notin$ G. Let **i**-successors of v_1 be $N_{\mathbf{i}}(v_1) = \{w \mid (v_1, w) \in \mathcal{R}_{\mathbf{i}}\}$. Since $\mathcal{M}_1$ is image finite, let $N_{\mathbf{i}}(v_1) = \{w_1 \cdots w_m\}$. By above argument, for all $l \leq n$ and $d \leq m$ we have $(w_d, u_l) \notin$ G . Hence for every $l \leq n$ and every $d \leq m$ there is a formula φ_d^l such that $\mathcal{M}_1, w_d \models \varphi_d^l$ but $\mathcal{M}_2, u_l \models \neg\varphi_d^l$.

Now consider the formula $\alpha = [\exists](\bigwedge_l \bigvee_d \varphi_d^l)$. It can be verified that $\mathcal{M}_1, v_1 \models \Box_{\mathbf{i}}(\bigwedge_l \bigvee_d \varphi_d^l)$ but $\mathcal{M}_2, v_2 \models \langle\forall\rangle(\bigvee_l \bigwedge_d \neg\varphi_d^l)$ which contradicts $(v_1, v_2) \in$ G.

The ([∃]back) condition is argued symmetrically.

Suppose that the (⟨∃⟩back) condition does not hold. Then there is some $\mathbf{j} \in \mathcal{I}_2$ and some $v_2 \xrightarrow{j} \mathbf{u_2}$ such that for all $i \in \delta_1(w_1)$ and for all $w \in \mathcal{W}_2$ if $v_1 \xrightarrow{i} w$ then $(\mathbf{u_2}, w) \notin$ G. Let $N(v_1) = \{w \mid (v_1, w) \in \mathcal{R}_i$ for some $i \in \mathcal{I}_1\}$. Since $\mathcal{M}_1$ is image-finite, let $N(v_1) = \{w_1, \ldots w_m\}$. By above argument, for every $w_d \in N(v_1)$ there is a formula ψ_d such that $\mathcal{M}_1, w_d \models \psi_d$ and $\mathcal{M}_2, \mathbf{u_2} \models \neg\psi_d$. Hence $\mathcal{M}_2, v_2 \models \Diamond_{\mathbf{j}}(\bigwedge_d \neg\psi_d)$ but $\mathcal{M}_1, v_1 \models [\forall](\bigvee_d \psi_d)$ which contradicts $(v_1, v_2) \in$ G.

The (⟨∃⟩forth) is argued symmetrically.

□

6.3.1 Characteristic formula

As in PTML, we can define n-bisimulation that preserves n modal depth IQML formulas. Moreover, we can give a formula that characterises n-bisimilarity. Formally, over a finite set of propositions, for every IQML model $\mathcal{M}$ and every $w \in \mathcal{W}$ there is a formula $\chi^n_{[\mathcal{M},w]}$ such that for any IQML model $\mathcal{N}$ and $u \in \mathcal{W}^{\mathcal{N}}$ if $\mathcal{M}, u \models \chi^n_{[\mathcal{M},w]}$ then $\mathcal{M}, w$ is n-IQML bisimilar to $\mathcal{M}, u$.

Definition 6.17. *Given two* IQML *models* $\mathcal{M}_1$ *and* $\mathcal{M}_2$*, a sequence* $\{\mathrm{G}^0, \ldots, \mathrm{G}^n\}$ *is called an* n-IQML *bisimulation where each* $G^i \subseteq (\mathcal{W}_1 \times \mathcal{W}_2)\}$ *is a non-empty relation such that for all* $k \leq n$ *and* $(w_1, w_2) \in \mathrm{G}^k$ *the following holds:*

val. $\rho_1(w_1) = \rho_2(w_2)$.

If $k > 0$

n*-[∃]forth.* *For all* $i \in \mathcal{I}_1$ *there is some* $j \in \mathcal{I}_2$ *such that for all* $w_2 \xrightarrow{j} u_2$ *there is some* $w_1 \xrightarrow{i} u_1$ *such that* $(u_1, u_2) \in \mathrm{G}^{k-1}$.

n*-[∃]back.* *For all* $j \in \mathcal{I}_2$ *there is some* $i \in \mathcal{I}_1$ *such that for all* $w_1 \xrightarrow{i} u_1$ *there is some* $w_2 \xrightarrow{j} u_2$ *such that* $(u_1, u_2) \in \mathrm{G}^{k-1}$.

n*-⟨∃⟩forth.* *For all* $i \in \mathcal{I}_1$ *and for all* $w_1 \xrightarrow{i} u_1$ *there is some* $j \in \mathcal{I}_2$ *and some* $w_2 \xrightarrow{j} u_2$ *such that* $(u_1, u_2) \in \mathrm{G}^{k-1}$.

n*-⟨∃⟩back.* *For all* $j \in \mathcal{I}_2$ *and for all* $w_2 \xrightarrow{j} u_2$ *there is some* $i \in \mathcal{I}_1$ *and some* $w_1 \xrightarrow{i} u_1$ *such that* $(u_1, u_2) \in \mathrm{G}^{k-1}$.

We write $(\mathcal{M}_1, w_1) \sim_n (\mathcal{M}_2, w_2)$ if there is some n-IQML bisimulation $\{G^0, \ldots, \mathrm{G}^n\}$ such that $(q_1, w_2) \in \mathrm{G}^n$. Similarly we have $(\mathcal{M}_1, w_1) \triangleq_n (\mathcal{M}_2, w_2)$ if they agree on all IQML formulas of modal depth at most n.

Along the lines of Theorem 6.15, we can show that if $(\mathcal{M}_1, w_1) \sim_n (\mathcal{M}_2, w_2)$ then $(\mathcal{M}_1, w_1) \triangleq_n (\mathcal{M}_2, w_2)$. Now we prove that n-IQML bisimulation can be characterized by an IQML formula of modal depth n over a finite set of propositions. The proof is along the standard lines as in ML^n (refer Goranko and Otto, [GO07]).

Note that, along the lines of Theorem 6.7, we can show that any IQML model is IQML-bisimilar to the tree unravelling. Hence we restrict our attention to tree models. Given an IQML tree model $\mathcal{M}$ we define its restriction to level n in the obvious manner: $\mathcal{M}|n$ is simply the same as $\mathcal{M}$ upto level n and the remaining nodes in $\mathcal{M}$ are 'thrown away'.

Lemma 6.18. *Let the set of propositions $\mathcal{P}^0$ be a finite set. For all n and for all* IQML *model $\mathcal{M}$ and for all $w \in \mathcal{W}$ there is a formula $\chi^n_{[\mathcal{M},w]} \in \mathsf{IQML}$ of modal depth n such that for any model $\mathcal{M}'$ and $w' \in \mathcal{W}'$ we have $(\mathcal{M}', w') \models \chi^n_{[\mathcal{M},w]}$ iff $(\mathcal{M}', w') \rightleftharpoons_n (\mathcal{M}, w)$.*

Proof. Note that ($\Leftarrow$) follows from Theorem 6.15 specialized to n-IQML bisimulation. For the other direction, for all $u \in \mathcal{W}$ we construct $\chi^n_{[\mathcal{M},u]}$ by induction on n. For $n = 0$, since $\mathcal{P}^0$ is finite, $\chi^0_{[\mathcal{M},u]} = \bigwedge\limits_{p \in \rho(u)} p \wedge \bigwedge\limits_{q \notin \rho(u)} \neg q$ is the required formula.

By induction, for all $u \in \mathcal{W}$ we have $\chi^n_{[\mathcal{M},u]}$. Let $\mathcal{R} = \bigcup \mathcal{R}_i$ and let $\Gamma^n_{\mathcal{M}} = \{\chi^n_{[\mathcal{M},u]} \mid u \in \mathcal{W}\}$. Note that inductively $\Gamma^n_{\mathcal{M}}$ is finite. For any $S \subseteq \Gamma^n_{\mathcal{M}}$ let $\overset{\vee}{S}$ denote the disjunction $\bigvee\limits_{\varphi \in S} \varphi$. For the induction step, the characteristic formula is given by:

$$\chi^{n+1}_{[\mathcal{M},u]} = \overbrace{\chi^0_{[\mathcal{M},u]}}^{Val.} \wedge \overbrace{\bigwedge_{i \in \mathcal{I}} [\exists](\bigvee_{(u,v) \in \mathcal{R}_i} \chi^n_{[\mathcal{M},v]})}^{n-[\exists]\text{forth}} \wedge \overbrace{\bigwedge_{S \subseteq \Gamma^n_{\mathcal{M}}} ([\exists](\overset{\vee}{S}) \to \bigvee_{i \in \mathcal{I}} \bigwedge_{(u,v) \in \mathcal{R}_i} [\forall](\chi^n_{[\mathcal{M},v]} \to \overset{\vee}{S}))}^{n-[\exists]\text{back}}$$
$$\underbrace{\bigwedge_{(u,v) \in \mathcal{R}} \langle \exists \rangle \chi^n_{[M,v]}}_{n-\langle \exists \rangle\text{forth}} \wedge \underbrace{[\forall](\bigvee_{(w,v) \in \mathcal{R}} \chi^n_{[M,v]})}_{n-\langle \exists \rangle\text{back}}$$

The formula remains finite even if $\mathcal{I}$ is infinite or the number of successors of u is infinite, since inductively there are only finitely many characteristic formulas of depth n. We now prove that the formula $\chi^n_{[\mathcal{M},u]}$ indeed captures n-bisimulation.

Note that for any finite set of formulas T, if $\alpha \in T$ then the formula $\alpha \to (\bigvee\limits_{\psi \in T} \psi)$ is a propositional validity. Also, for all n if $\chi \in \Gamma^n_{\mathcal{M}}$ such that $\mathcal{M}', u' \models \chi$ then for all other characteristic formulas $\chi' \in \Gamma^n_{\mathcal{M}}$ we have $\mathcal{M}', w' \not\models \chi'$.

First we verify that the formula $\chi^n_{[\mathcal{M},u]}$ holds at $\mathcal{M}, u$:

- $\mathcal{M}, u \models \chi^0_{[\mathcal{M},w]}$ follows from the definition of ρ.

- For the $n - [\exists]$ forth part, for every $i \in \mathcal{I}$ we have $\mathcal{M}, u \models \Box_i(\bigvee_{(u,v)\in\mathcal{R}_i} \chi^n_{[\mathcal{M},v]})$ and hence the claim follows.

- For the $n - [\exists]$back part, let $S \subseteq \Gamma^n_{\mathcal{M}}$. Suppose $\mathcal{M}, u \models [\exists]\overset{\vee}{S}$, let $\mathbf{j}$ be the witness. Hence we have $\mathcal{M}, u \models \Box_{\mathbf{j}}\overset{\vee}{S}$. Now observe that for all $(u, v) \in \mathcal{R}_{\mathbf{j}}$ we have $\chi^n_{[\mathcal{M},v]} \in S$, otherwise there is some $(u, v) \in R_{\mathbf{j}}$ such that $\mathcal{M}, v \models \bigwedge_{\varphi\in S} \neg\varphi$ which is a contradiction to $\mathcal{M}, u \models \Box_{\mathbf{j}}\overset{\vee}{S}$.

  Now we need to show that $\mathcal{M}, u \models \bigvee_{i\in\mathcal{I}} \bigwedge_{(u,v)\in\mathcal{R}_i} [\forall](\chi^n_{[\mathcal{M},u]} \to \overset{\vee}{S})$. For this, set $i = \mathbf{j}$ and pick any $(u, v) \in \mathcal{R}_{\mathbf{j}}$. From the above argument, $\chi^n_{[\mathcal{M},v]} \in S$ and hence the formula $(\chi^n_{[\mathcal{M},v]} \to \overset{\vee}{S})$ is a validity. Thus, $\mathcal{M}, u \models \bigwedge_{(u,v)\in\mathcal{R}_{\mathbf{j}}} [\forall](\chi^n_{[\mathcal{M},u]} \to \overset{\vee}{S})$.

- For $n - \langle\exists\rangle$forth, let $(u, v) \in \mathcal{R}$ which means for some $i \in \mathcal{I}$ we have $(u, v) \in R_i$ and by induction hypothesis $\mathcal{M}, v \models \chi^n_{[\mathcal{M},v]}$. Hence $\mathcal{M}, u \models \langle\exists\rangle\chi^n_{[\mathcal{M},u]}$.

- For $n - \langle\exists\rangle$back, for any $i \in \mathcal{I}$ and any $(u, v) \in R_i$ we have $\mathcal{M}, v \models \chi^n_{[\mathcal{M},v]}$ and hence $\mathcal{M}, u \models [\forall](\bigvee_{(u,v)\in\mathcal{R}} \chi^n_{[\mathcal{M},v]})$.

Thus for all $u \in \mathcal{W}$ and for all n we have $\mathcal{M}, u \models \chi^n_{\mathcal{M},u]}$.

Now suppose $\mathcal{M}', w' \models \chi^n_{[\mathcal{M},w]}$, then we need to prove that $(\mathcal{M}, w) \sim_n (\mathcal{M}', w')$. Define $\{\mathrm{G}^0, \ldots, \mathrm{G}^n\}$ where every $G^k = \{(u, u') \mid \mathcal{M}', u' \models \chi^k_{[\mathcal{M},u]}\}$. Now we verify all the required conditions for all $k \leq n + 1$ and all $(u, u') \in \mathrm{G}^k$.

The (Val) condition holds since $\mathcal{M}', u' \models \chi^0_{[\mathcal{M},u]}$. If $k > 0$ then we need to verify the rest of the conditions.

- For $(n - [\exists]$forth$)$, pick any $\mathbf{i} \in \mathcal{I}$. We have $\mathcal{M}', u' \models \chi^k_{[\mathcal{M},u]}$. By ($[\exists]$forth) part of $\chi^k_{\mathcal{M},u]}$ we have $\mathcal{M}', u' \models [\exists](\bigvee_{(u,v)\in\mathcal{R}_i} \chi^l_{[\mathcal{M},v]})$. Let $\mathbf{j}' \in \mathcal{I}'$ be the witness such that $\mathcal{M}', u' \models \Box_{\mathbf{j}'}(\bigvee_{(u,v)\in\mathcal{R}_i} \chi^l_{[\mathcal{M},v]})$. Hence for every $(u', v') \in \mathcal{R}'_{\mathbf{j}'}$ there is some $(u, v) \in R_{\mathbf{i}}$ such that $\mathcal{M}', v' \models \chi^{l-1}_{[\mathcal{M},v]}$ and by definition we have $(v, v') \in \mathrm{G}^{l-1}$.

- For condition $(n-[\exists]\text{back})$, let $\mathbf{j}' \in \mathcal{I}'$.

  Define $S = \{\chi^{k-1}_{[\mathcal{M},v]} \mid \text{for some } (u',v') \in \mathcal{R}_{\mathbf{j}'} \text{ we have } \mathcal{M}',v' \models \chi^{k-1}_{[\mathcal{M},v]}\}$. Now clearly, $\mathcal{M}',u' \models [\exists](\overset{\vee}{S})$. Hence by ($n$-$[\exists]$back) part of $\chi^{l}_{[\mathcal{M},u]}$, there is some $\mathbf{i} \in \mathcal{I}_1$ such that $\mathcal{M}',u' \models \bigwedge_{(u,v)\in\mathcal{R}_{\mathbf{i}}} [\forall](\chi^{l-1}_{[\mathcal{M},v]} \to \overset{\vee}{S})$. In particular, $\mathcal{M}',u' \models \bigwedge_{(u,v)\in\mathcal{R}_{\mathbf{i}}} \Box_{\mathbf{j}'}(\chi^{l-1}_{[\mathcal{M},v]} \to \overset{\vee}{S})$ but we also have $\mathcal{M}',u' \models \Box_{\mathbf{j}'}(\overset{\vee}{S})$. Thus, by definition of S we have $\mathcal{M}',u' \models \Box_{\mathbf{j}'}(\bigvee_{(u,v)\in\mathcal{R}_{\mathbf{i}}} \chi^{l-1}_{[\mathcal{M},v]})$. Let $\mathbf{i}$ be the ($[\exists]$back) witness for $\mathbf{j}'$.

  Hence, for every $(u,v) \in \mathcal{R}_{\mathbf{i}}$ there is some $(u',v') \in \mathcal{R}_{\mathbf{j}'}$ such that $\mathcal{M}',v' \models \chi^{l-1}_{\mathcal{M},v]}$ and hence $(v,v') \in \mathrm{G}^{l-1}$.

- For the $n-\langle\exists\rangle$forth condition, let $i \in \mathcal{I}$ and $(u,v) \in \mathcal{R}_i$. By $n-\langle\exists\rangle$forth part of the formula, $\mathcal{M}',u' \models \langle\exists\rangle\chi^{l-1}_{[\mathcal{M},v]}$ and hence we have a corresponding $i' \in \mathcal{I}'$ and $(u',v') \in \mathcal{R}_{i'}$ such that $\mathcal{M}',v' \models \chi^{n-1}_{[\mathcal{M},v]}$ and hence $(v,v') \in \mathrm{G}^{l-1}$.

- Finally for $n-\langle\exists\rangle$back, suppose $i' \in \mathcal{I}$ and $(w',u') \in \mathcal{R}_{i'}$ then by $n-\langle\exists\rangle$back part of the formula, $\mathcal{M}',u' \models \chi^{n-1}_{[\mathcal{M},u]}$ for some $i \in \mathcal{I}$ and $(w,u) \in R_i$. Thus we obtain the required witness.

□

6.3.2 Bisimulation games and invariance theorem

There is a natural translation of IQML into a fragment of first order logic over two sorted domain: one for agents and the other for worlds.

In the syntax we have two sorts of variables $\mathcal{V}_X$ and $\mathcal{V}_Y$ for worlds and agents respectively. Further, we have a ternary predicate R to encode the accessibility relation. To encode propositions, for every $p \in \mathcal{P}^0$ we have a corresponding monadic predicate Q_p.

Definition 6.19 (2Sor.FO syntax)**.** *Let $\mathcal{V}_X$ and $\mathcal{V}_Y$ be two countable and disjoint sorts of variables. Let R be ternary predicate and for every $p \in \mathcal{P}^0$ let Q_p is the corresponding monadic predicate . The two sorted FO* (2Sor.FO)*, corresponding to IQML is given by:*

$$\alpha ::= Q_p(x) \mid R(x, y, x') \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists y\ \alpha \mid \exists x\ \alpha$$

where $x, x' \in \mathcal{V}_X$ and $y \in \mathcal{V}_Y$.

A 2Sor.FO structure is given by $\mathfrak{M} = [(\mathcal{W}, \mathcal{I}), (\hat{R}, \hat{\rho})]$ where $(\mathcal{W}, \mathcal{I})$ is the two sorted domain and $(\hat{R}, \hat{\rho})$ are interpretations with $\hat{R} \subseteq (\mathcal{W} \times \mathcal{I} \times \mathcal{W})$ and $\hat{\rho} : \mathcal{W} \mapsto 2^{Q_\mathcal{P}}$ where $Q_\mathcal{P} = \{Q_p \mid p \in \mathcal{P}\}$. The semantics $\Vdash$ is defined for 2Sor.FO in the standard way where the variables in $\mathcal{V}_X$ range over the first sort $(\mathcal{W})$ and variables of $\mathcal{V}_Y$ range over second $(\mathcal{I})$.

Given an IQML structure $\mathcal{M} = (\mathcal{W}, \mathcal{I}, \mathcal{R}, \delta, \rho)$ the corresponding 2Sor.FO structure is given by $\mathfrak{M} = [(\mathcal{W}, \mathcal{I}), (\hat{R}, \hat{\rho})]$ where $(w, i, v) \in \hat{R}$ iff $(w, v) \in \mathcal{R}_i$ and $Q_p \in \hat{\rho}(w)$ iff $p \in \rho(w)$. Similarly given any 2Sor.FO structure, it can be interpreted as an IQML structure. Thus there is a natural correspondence between IQML structures and 2Sor.FO structures. For any IQML structure $\mathcal{M}$ let the corresponding 2Sor.FO structure be denoted by $\mathfrak{M}$.

Definition 6.20 (IQML to 2Sor.FO translation)**.** *The translation of $\varphi \in$ IQML into a* 2Sor.FO *parametrized by $x \in \mathcal{V}_X$ is given by:*

$Tr_6(p\ : x) = Q_p(x)$

$Tr_6(\neg\varphi\ : x) = \neg\, Tr_6(\varphi\ : x)$

$Tr_6(\varphi \wedge \psi :\ x) = Tr_6(\varphi :\ x) \wedge Tr_6(\psi\ : x)$

$Tr_6([\exists]\varphi :\ x) = \exists\tau\forall y\ (R(x, \tau, y) \rightarrow Tr_6(\varphi :\ y))$

$Tr_6([\forall]\varphi :\ x) = \forall\tau\forall y\ (R(x, \tau, y) \rightarrow Tr_6(\varphi :\ y))$

Note that this translation can be achieved using two variables of each sort.

Proposition 6.21. *For any formula $\varphi \in$ IQML and any IQML structure $\mathcal{M}$*
$\mathcal{M}, w \models \varphi$ iff $\mathfrak{M}, [x \mapsto w] \Vdash Tr_6(\varphi : \ x)$.

Hence IQML can be translated into 2Sor.FO with 2 variables of $\mathcal{V}_X$ sort and one variable of $\mathcal{V}_Y$ sort. Given two IQML models $\mathcal{M}_1$ and $\mathcal{M}_2$, the notion of IQML bisimulation naturally translates to bisimulation over the corresponding 2Sor.FO models $\mathfrak{M}_1$ and $\mathfrak{M}_2$.

Given that every IQML formula can be translated to 2Sor.FO, a natural question arises: When does a 2Sor.FO formula have an equivalent IQML formula? van Benthem studied this question for propositional modal logic for its corresponding translation into first order logic and proved that every first order logic formula is bisimulation invariant iff there is an equivalent modal formula [BdRV01, Ben10b].

We prove a similar theorem for IQML: every IQML bisimulation invariant 2Sor.FO formula has an equivalent IQML formula. Define $\alpha(x) \in$ 2Sor.FO to be *IQML bisimulation invariant* if for all $(\mathcal{M}_1, w_1) \sim (\mathcal{M}_2, w_2)$ we have $\mathfrak{M}_1, [x \mapsto w_1] \Vdash \alpha(x)$ iff $\mathfrak{M}_2, [x \mapsto w_2] \Vdash \alpha(x)$. We can similarly speak of $\alpha(x)$ being n-IQML bisimulation invariant as well. Also, $\alpha(x)$ is equivalent to some IQML formula if there is some formula $\varphi \in$ IQML such that for all $\mathcal{M}$ we have $\mathfrak{M}, [x \mapsto w] \Vdash \alpha(x)$ iff $\mathcal{M}, w \models \varphi$.

Theorem 6.22. *Let $\alpha(x) \in$ 2Sor.FO with one free variable $x \in \mathcal{V}_X$. Then $\alpha(x)$ is IQML bisimulation invariant iff $\alpha(x)$ is equivalent to some IQML formula.*

Note that $\Leftarrow$ follows from Theorem 6.15. To prove $(\Rightarrow)$ it suffices to show that if $\alpha(x)$ is bisimulation invariant then, for some n the formula $\alpha(x)$ is n-IQML bisimulation invariant. The theorem follows since, by Lemma 6.18, n-bisimulation classes have a characteristic formula and the equivalent formula of $\alpha(x)$ is the disjunction of the characteristic formulas of all the n-bisimulation classes which are identified by $\alpha(x)$.

We follow the proof strategy similar to that for propositional modal logic as described in (Goranko and Otto [GO07]). Towards proving this, we introduce a notion of locality for 2Sor.FO formulas. As usual, we restrict our attention to rooted tree models. For any IQML tree model $\mathcal{M}$ and let $\mathcal{M}|n$ be the restriction of $\mathcal{M}$ to nodes at depth at most n and the 2Sor.FO model corresponding to $\mathcal{M}|n$ is given by $\mathfrak{M}|n$.

Definition 6.23. *We say that a formula $\alpha(x)$ is n-local if for any IQML tree model $\mathcal{M}$ rooted at w we have $\mathfrak{M} \Vdash \alpha(w)$ iff $\mathfrak{M}|n \Vdash \alpha(w)$.*

Lemma 6.24. *For any $\alpha(x) \in$ 2Sor.FOwith $x \in \mathcal{V}_X$, if $\alpha(x)$ is bisimulation invariant then $\alpha(x)$ is n-local for $n = 2^q$ where q is the number of quantifications in $\alpha(x)$ (including both $\mathcal{V}_X$ and $\mathcal{V}_Y$ sorts).*

Assuming this lemma, consider a 2Sor.FO formula $\alpha(x)$ which is bisimulation invariant. It is n-local for a syntactically determined n. We now claim that $\alpha(x)$ is n-bisimulation invariant. To prove this, consider any $(\mathcal{M}_1, w_1) \rightleftharpoons_n (\mathcal{M}_2, w_2)$. We need to show that $\mathfrak{M}_1, [x \mapsto w_1] \Vdash \alpha(x)$ iff $\mathfrak{M}_2, [x \mapsto w_2] \Vdash \alpha(x)$. Suppose that $\mathfrak{M}_1, [x \mapsto w_1] \Vdash \alpha(x)$. By locality, $\mathfrak{M}_1|n, [x \mapsto w_1] \Vdash \alpha(x)$. Now observe that $(\mathcal{M}_1|n, w_1) \rightleftharpoons (\mathcal{M}_2|n, w_2)$. By bisimulation invariance of $\alpha(x)$, $\mathfrak{M}_2|n, [x \mapsto w_2] \Vdash \alpha(x)$. But then again by locality, $\mathfrak{M}_2, [x \mapsto w_2] \Vdash \alpha(x)$. The other direction is symmetric.

Thus it only remains to prove the locality lemma. For this, it is convenient to consider the *Ehrenfeucht-Fraisse* (EF) game for 2Sor.FO. In this game we have two types of pebbles, one for $\mathcal{W}$ and the other for $\mathcal{I}$. The game is played between two players Spoiler (**Sp**) and Duplicator (**Dup**) on two 2Sor.FO structures.

A configuration of the game is given by $[(\mathfrak{M}, \bar{s}); (\mathfrak{M}', \bar{t})]$ where $\bar{s} \in (\mathcal{W} \cup \mathcal{I})^*$ is a finite string $(\mathcal{W} \cup \mathcal{I})$ and similarly $\bar{t} \in (\mathcal{W}' \cup \mathcal{I}')^*$. The game has two kinds of rounds, namely $\mathcal{W}$ round and $\mathcal{I}$ round.

Suppose the current configuration is $[(\mathfrak{M}, \bar{s}); (\mathfrak{M}', \bar{t})]$. In a $\mathcal{W}$ round, **Sp** places a $\mathcal{W}$ pebble on some $\mathcal{W}$ sort in one of the structures and **Dup** responds by placing a $\mathcal{W}$ pebble on a $\mathcal{W}$ sort in the other structure. Similarly, in a $\mathcal{I}$ round, **Sp** picks one structure and places an $\mathcal{I}$ pebble on some $\mathcal{I}$ sort and **Dup** responds by placing an $\mathcal{I}$ pebble on some $\mathcal{I}$ sort in the other structure. In both cases, the new configuration is updated to $[(\mathfrak{M}, \bar{s}s); (\mathfrak{M}', \bar{t}t)]$ where s and t are the new elements (either $\mathcal{W}$ or $\mathcal{I}$ sort) picked in the corresponding structures.

A (q_x, q_y) round game is one where q_x many pebbles of type $\mathcal{W}$ are used and q_y many pebbles of type $\mathcal{I}$ is used. Suppose after (q_x, q_y) rounds, the final configuration is $[(\mathfrak{M}, \bar{s}); (\mathcal{M}', \bar{t})]$. Player **Dup** wins if the mapping $f(s_i) = t_i$ forms a partial isomorphism over $\mathfrak{M}$ and $\mathfrak{M}'$. Otherwise **Sp** wins.

It can be shown along standard lines that **Dup** has a winning strategy in the (q_x, q_y) round game over two structures iff they agree on all formulas with quantifier rank of $\mathcal{V}_X$ sort $\leq q_x$ and quantifier rank of $\mathcal{V}_Y$ sort $\leq q_y$ (refer books [BdRV01, PG92]).

Let $\mathcal{M}, w$ be any tree structure. To prove Lemma 6.24, we need to show that $\mathfrak{M}, w \models \alpha(x)$ iff $\mathfrak{M}|n \models \alpha(x)$ where n is as described in the lemma.

Note that *inclusion* relation G over $\mathfrak{M}$ and $\mathfrak{M}|n$ forms an n-IQML bisimulation. Let $q = q_x + q_y$ and $\mathfrak{N}$ be q disjoint copies of $\mathfrak{M}$ and $\mathfrak{M}|n$. Also note that G continues to be an n-IQML bisimulation over the disjoint union of $(\mathfrak{N} \uplus \mathfrak{M}, w)$ and $(\mathfrak{N} \uplus \mathfrak{M}|n,\ w)$. Moreover, notice that $(\mathfrak{M}, w)$ is n-IQML bisimilar to $(\mathfrak{N} \uplus \mathfrak{M}, w)$ and further $(\mathfrak{M}|n,\ w)$ is n-IQML bisimilar to $(\mathfrak{N} \uplus \mathfrak{M}|n,\ w)$.

Now since $\alpha(x)$ is bisimulation invariant, it is enough to show that **Dup** has a winning strategy in the game starting from $[(\mathfrak{N} \uplus \mathfrak{M}, w),\ (\mathfrak{N} \uplus \mathfrak{M}|n,\ w)]$.

To describe the winning strategy for **Dup**, we introduce the notion of distance between any two worlds in an IQML model which is given by the length of path between the worlds. Suppose u_1 is at a distance of m from w_1 then we need a formula of modal depth m to access u_1 from w_1. Hence for **Dup** to win, in $\mathcal{W}$ round, she just needs to ensure that at every round m the m-distance neighbours around a worlds marked by pebbles forms a partial isomorphism and in $\mathcal{I}$ round, she can just play according to the *identity* mapping since the index set $\mathcal{I}$ is the same on both sides.

In particular, if **Sp** places $\mathcal{W}$ pebble on a $\mathcal{W}$ sort which is within m distance of an already pebbled $\mathcal{W}$ pebble, **Dup** plays according to a local isomorphism in the m- neighbourhoods of previously pebbled elements (such move exists since $n = 2^q$ and $m < q$); if **Sp** places a $\mathcal{W}$ pebble somewhere beyond 2^{q-m} distance from all $\mathcal{W}$ pebbles previously used, then, **Dup** responds in a fresh isomorphic copy of type $\mathfrak{M}$ or $\mathfrak{M}|n$ correspondingly (again, this move is guaranteed to exist since previously at most $m-1(< q)$ copies would have been used).

If **Sp** decides to use an $\mathcal{I}$ pebble and places it on some $\mathcal{I}$ sort i in one structure, then **Dup** responds by placing an $\mathcal{I}$ pebble on i in the mirror copy in the other structure, where by mirror copy we mean: for $\mathfrak{M}$ or $\mathfrak{M}|n$ in $\mathfrak{N}$ then the mirror copy in the other structure is itself and the original $\mathfrak{M}$ and $\mathfrak{M}|n$ are mirror copies of each other.

This completes the proof of locality Lemma 6.24 and thus of Theorem 6.22.

6.3.3 IQML and 1-variable fragment of PTML

The formulas in IQML can be translated into 1-variable fragment of PTML by inductively replacing $[\exists]\varphi$ by $\exists x \Box_x \varphi$ and $\langle\exists\rangle\varphi$ by $\exists x \Diamond_x \varphi$.

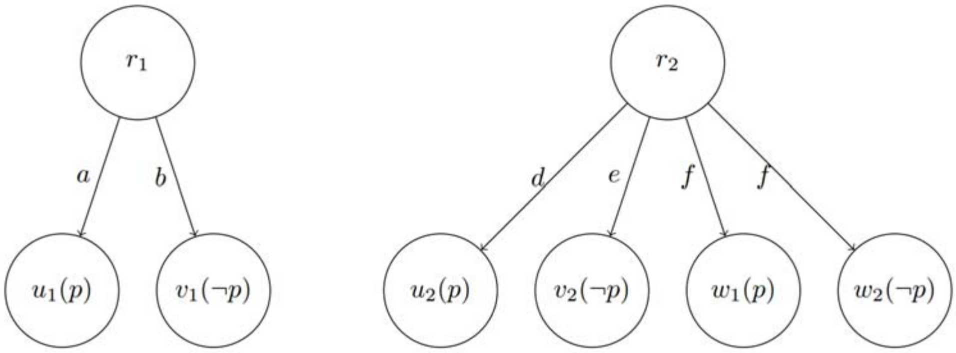


Figure 6.1: Two models which are IQML bisimilar but can be distinguished by 1-variable formula of PTML. Both are constant agent models with $\{a, b\}$ being the agents in $\mathcal{M}_1$ and $\{d, e, f\}$ as agents in $\mathcal{M}_2$.

Now we prove that the 1-variable fragment of PTML is strictly more expressive than IQML. For this we give two models which are IQML bisimilar but there is a 1-variable PTML formula that can distinguish the models.

Consider the models $\mathcal{M}_1$ and $\mathcal{M}_2$ described in Fig. 6.1.
Note that $(\mathcal{M}_1, r_1) \sim (\mathcal{M}_2, r_2)$ with $\mathrm{G} = \{(r_1, r_2), (u_1, u_2), (v_1, v_2), (w_1, u_1), (w_2, v_1)\}$.
On the other hand $\mathcal{M}_1, r_1 \models \exists x\ (\Diamond_x p \wedge \Diamond_x \neg p)$ but $\mathcal{M}_2, r_2 \not\models \exists x\ (\Diamond_x p \wedge \Diamond_x \neg p)$.

6.4 Discussion

We introduced the notion of bisimulation for PTML and $\mathsf{PTML}_=$. Note that the definition of bisimulation can be lifted to TML (and $\mathsf{TML}_=$) by modifying the first condition to say that the predicate type are same at bisimilar worlds. Formally, if $(w_1, w_2) \in \mathrm{G}$ then for all $c_1, \ldots, c_n \in \delta_1(w_1)$ and for all $(c_1, d_1), \ldots, (c_n, d_n) \in \mathrm{H}_{(w_1, w_2)}$ and for all predicate P of arity n, $(c_1, \ldots, c_n) \in \rho_1(w_1, P)$ iff $(d_1, \ldots, d_n) \in \rho_2(w_2, P)$ and vice-versa. With this, all the theorems discussed in this section go through.

Similarly, we can also specialize the notion of bisimulation to capture exactly the *monodic fragment of PTML*.

Recall that φ is monodic fragment if every modal subformula of the form $\Delta_x\psi$ has $\mathsf{FV}(\psi) \subseteq \{x\}$ where $\Delta \in \{\Box, \Diamond\}$. Thus, the formulas can carry an agent into a world only if it is the incoming edge label. Thus, the conditions *agent forth and back* of bisimulation (Def. 6.1) can be specialized for monodic PTML by replacing $H_{(w_1,w_2)} \subseteq H_{(u_1,u_2)}$ with $(c, d) \in H_{(u_1,u_2)}$. This relaxed condition suffices for monodic formulas since they cannot access any other agent using modal formulas except the agent which corresponds to the incoming label edge. With this modification, we can prove that monodic bisimulation implies monodic elementary equivalence.

If we have constants $\mathbf{C}$ in the vocabulary, then we have an additional condition in bisimulation that for all $(u_1, u_2) \in \mathrm{G}$ and for all constants $\mathbf{c} \in \mathbf{C}$ if interpretations of $\mathbf{c}$ are d_1 and d_2 in u_1 and u_2 respectively, then $(d_1, d_2) \in \mathrm{H}_{(u_1,u_2)}$. Note that this condition is enough to capture elementary equivalence for both rigid and non-rigid interpretations of constants.

We discussed van Benthem type theorem for IQML. Similarly, PTML and TML can be translated to 2Sor.FO over the appropriate vocabulary. For these fragments also, a similar van Benthem characterization needs to be worked out. For instance, can we always get the characterization theorem using finite models or do we have to use omega saturated models?

Chapter 7

Model checking

For any logic $\mathcal{L}$, given a formula φ in $\mathcal{L}$ and some $\mathcal{L}$-structure $\mathfrak{A}$, a natural question is to check whether $\mathfrak{A} \models \varphi$. For this to be an algorithmic problem, both the inputs should be finite. This is known as the model checking problem or the verification problem for $\mathcal{L}$.

In particular, for TML, the model checking problem is to take a TML formula φ and a finite structure $\mathcal{M}$ with $w \in \mathcal{W}$ and an interpretation σ relevant at w as inputs and decide whether $\mathcal{M}, w, \sigma \models \varphi$.

In this chapter, we will first discuss the complexity issues for model checking TML and PTML. Note that the interest of TML is that it can specify properties about infinitely branching structures. So if we present such infinite models in a finite fashion, model checking over such structures could be of interest. Towards this, we introduce a finite specification of infinite models using regular expressions and prove that model checking problem for PTML over such input specification is decidable.

Logic	Combined Complexity	Expression Complexity	Data Complexity
FO	PSPACE	PSPACE	PTIME
ML	PTIME	PTIME	PTIME
TML	PSPACE	PSPACE	PTIME
PTML	PSPACE	PSPACE	PTIME

Table 7.1: Summary of complexity results for the model checking problem over finite structures.

7.1 Finite structures

First we consider the model checking problem for TML. Note that the model checking algorithm has two parameters, one of them is a TML formula φ and the other is a finite TML structure $\mathcal{M}$ with a designated $w \in \mathcal{W}$ and an interpretation $\sigma : \mathsf{FV}(\varphi) \mapsto \delta(w)$, which is together represented as a tuple $(\mathcal{M}, w, \sigma)$ which we call a pointed TML structure. The output is *yes* or *no* depending upon whether $\mathcal{M}, w, \sigma \models \varphi$ or not respectively. Thus, we have three different cases depending on which parameters are fixed and which are provided by inputs. In the literature they are called **Combined Complexity** when we consider both parameters as inputs; **Expression Complexity** when the pointed TML structures are fixed (formulas are inputs) and **Data Complexity** when formulas are fixed (pointed TML structures are inputs).

Table 7.1 gives a summary of complexity results for model checking FO, ML, TML and PTML. Note that since FO is the modal free fragment of TML, the different variants of model checking problem for TML are at least as hard as that for FO. Further, model checking for TML also reduces to model checking over FO since there is a natural translation of TML to an appropriate 2-sorted FO similar to what we discussed in Chapter 6, Def. 6.20 for IQML. Thus, the **Combined Complexity**, **Expression Complexity** and **Data Complexity** of TML are in PSPACE, PSPACE and PTIME respectively.

On the other hand for PTML we can prove that the **Expression Complexity** (and hence **Combined Complexity**) continues to be PSPACE hard for PTML. To prove this, we use the reduction form quantified boolean formulas (QBF).

In fact, this follows from the translation of FO model checking to PTML model checking using the translation Tr_1 (Def. 4.1) described while proving $\mathsf{PTML}^{\top}$ is undecidable, where we translate every FO formula into $\mathsf{PTML}^{\top}$ formula (Def. 4.1, $\mathsf{Tr}_1(\varphi)$). In the proof of Theorem 4.2, we have a translation of FO model $\mathfrak{A}$ to a PTML model $\mathcal{M}$ rooted at r such that $\mathfrak{A}, \sigma \models \varphi$ iff $\mathcal{M}, r, \sigma \models \mathsf{Tr}_1(\varphi)$. The model translation and formula translation, both can be computed in PTIME. Hence we can prove that the **Expression Complexity** (and hence **Combined Complexity**) continues to be PSPACE hard for PTML. Here we give a direct proof by reduction from TQBF problem.

Definition 7.1. *(QBF) Let V be a countable set of variables. The boolean formulas (α) and QBF formulas (φ) over V are defined by:*

$$\begin{aligned} \alpha &:= s_i \mid \neg s_i \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \\ \varphi &:= \alpha \mid \exists s_i\ \varphi \mid \forall s_i\ \varphi \end{aligned} \qquad \textit{where } s_i \in V.$$

To keep the presentation simple, we assume that every variable is quantified at most once in any given QBF formula. Given a QBF formula φ, the free variables of φ is defined in the standard way where $\mathsf{FV}(s) = \mathsf{FV}(\neg s) = \{s\}$ and $\mathsf{FV}(\alpha \vee \beta) = \mathsf{FV}(\alpha \wedge \beta) = \mathsf{FV}(\alpha) \cup \mathsf{FV}(\beta)$ and $\mathsf{FV}(\exists s\ \varphi) = \mathsf{FV}(\forall s\ \varphi) = FV(\varphi) \setminus \{s\}$. We say that φ is a totally quantified boolean sentence (TQBF sentence) if $\mathsf{FV}(\varphi) = \emptyset$.

For any TQBF formula φ, any function $\sigma : \mathsf{FV}(\varphi) \mapsto \{\mathtt{T}, \mathtt{F}\}$ is called a *valuation*.

Definition 7.2. *For any QBF formula φ given a valuation $\sigma : FV(\varphi) \mapsto \{\mathtt{T}, \mathtt{F}\}$, we define $\sigma \models \varphi$ where*

$\sigma \models s_i$	$\Leftrightarrow$	$\sigma(s_i) = \mathtt{T}$
$\sigma \models \neg s_i$	$\Leftrightarrow$	$\sigma(s_i) = \mathtt{F}$
$\sigma \models \alpha \wedge \beta$	$\Leftrightarrow$	$\sigma \models \alpha$ *and* $\sigma \models \beta$
$\sigma \models \alpha \vee \beta$	$\Leftrightarrow$	$\sigma \models \alpha$ *or* $\sigma \models \beta$
$\sigma \models \exists s\ \varphi$	$\Leftrightarrow$	*there is some valuation* σ_s *such that* $\sigma_s \models \varphi$
$\sigma \models \forall s\ \varphi$	$\Leftrightarrow$	*for every valuation* σ_s *we have* $\sigma_s \models \varphi$

where $\sigma_s : \mathsf{FV}(\varphi) \cup \{s\} \mapsto \{\mathtt{T}, \mathtt{F}\}$ *is a mapping that extends* σ*.*

Thus for a TQBF sentence φ we can say whether $\models \varphi$ or not which corresponds to φ evaluating to $\mathtt{T}$ or $\mathtt{F}$ respectively.

Theorem 7.3 ([AB09]). *Deciding whether a given TQBF sentence evaluates to* $\mathtt{T}$ *or* $\mathtt{F}$ *is* PSPACE *complete.*

Definition 7.4 (QBF to PTML translation). *For every* $s_i \in V$*, let* $x_i \in \mathcal{V}$ *be the corresponding variable in the vocabulary of PTML. For any QBF formula* φ*, the translation to PTML formula is defined inductively as follows:*

- $\mathit{Tr}_5(s_i) = \Diamond_{x_i} \top$ *and* $\mathit{Tr}_5(\neg s_i) = \Box_{x_i} \bot$
- $\mathit{Tr}_5(\alpha \wedge \beta) = \mathit{Tr}_5(\alpha) \wedge \mathit{Tr}_5(\beta)$ *and* $\mathit{Tr}_5(\alpha \vee \beta) = \mathit{Tr}_5(\alpha) \vee \mathit{Tr}_5(\beta)$
- $\mathit{Tr}_5(\exists s_i \psi) = \exists x_i\ \mathit{Tr}_5(\psi)$ *and* $\mathit{Tr}_5(\forall s_i \psi) = \forall x_i\ \mathit{Tr}_5(\psi)$

To prove that **Expression Complexity** of PTML is PSPACE hard, first we need to fix the model. Define $\mathcal{M} = (\mathcal{W}, \mathcal{D}, \delta, \mathcal{R}, \rho)$ where $\mathcal{W} = \{u, v\}$ and $\mathcal{D} = \{0, 1\}$ with $\delta(u) = \delta(v) = \mathcal{D}$ and $\mathcal{R} = \{(u, 1, v)\}$ and also $\rho(w) = \emptyset$ for all $w \in W$. Note that $\mathcal{M}$ is simply a model with two worlds u, v both having agent set $\{0, 1\}$ and there is a 1-labelled edge from u to v.

For any valuation $\sigma : \{s_1, s_2 \ldots, s_n\} \mapsto \{\mathtt{T}, \mathtt{F}\}$ define $\hat{\sigma} : \{x_1, x_2, \ldots, x_n\} \mapsto \{0, 1\}$ such that $\hat{\sigma}(x_i) = 1$ iff $\sigma(s_i) = \mathtt{T}$.

Theorem 7.5. *Let $\mathcal{M}$ be the model defined above. For any QBF formula φ for any valuation $\sigma : \mathsf{FV}(\varphi) \mapsto \{\mathtt{T}, \mathtt{F}\}$ we have $\sigma \models \varphi$ iff $\mathcal{M}, u, \hat{\sigma} \models \mathsf{Tr}_5(\varphi)$.*

Proof. The proof is by induction on the structure of φ. In the base case φ is of the form s_i and $\sigma \models s_i$ iff $\sigma(s_i) = \mathtt{T}$ iff $\hat{\sigma}(x_i) = 1$ iff $\mathcal{M}, u, \hat{\sigma} \models \Diamond_{x_i}\top$. Similarly, $\sigma \models \neg s_i$ iff $\sigma(s_i) = \mathtt{F}$ iff $\hat{\sigma}(x_i) = 0$ iff $\mathcal{M}, u, \hat{\sigma} \models \Box_{x_i}\bot$.
The $\wedge$ and $\vee$ cases are standard.

For $\exists s_i\ \psi$, we have $\sigma \models \exists x_i\ \psi$ iff there is some assignment σ' that extends σ such that $\sigma' \models \psi$ iff $\mathcal{M}, u, \hat{\sigma'} \models \mathsf{Tr}_5(\psi)$ iff $\mathcal{M}, u, \hat{\sigma} \models \exists x_i\ \mathsf{Tr}_5(\psi)$.
The case of $\forall s_i\ \psi$ is analogous. $\square$

Corollary 7.6. *The* **Expression Complexity** *for PTML is* PSPACE *hard (and hence* **Combined Complexity** *is also* PSPACE *hard).*

7.2 Finitely specified structures

When we consider models which are potentially infinite, first we need a finite representation of such models which can be provided as input for the model checking algorithm. We motivate the finite representation with an example.

Example 7.7. *Consider an operating system which can execute many processes at a time. A configuration of the system is given by the states of its active processes. Any active process can change the system state by making a move. Any move by a process can create one or more new processes (threads), thus making the active set dynamic and potentially unbounded. In this setting, consider the following assertions:*

- *There is at least one process active which can potentially change the system state:*
  $\exists x\ \Diamond_x\top$.

- *For all possible next configurations, property p holds:*
  $\forall x(\Box_x p)$.

- *There are at least two active processes*[1]*:*
  $\exists x \Box_x p \wedge \exists y \ \Diamond_y \neg p$.

- *There is a process such that it can change to a configuration in which none of the processes can make a move (system halts):*
  $\exists x \Diamond_x \forall y \Box_y \bot$.

In such multi-thread dynamic systems, the names of the processes (id) can be thought of as strings over a finite alphabet. We assume that the processes come from a *regular set* and thus, can be specified as a *transition system* with finitely many states and edges between the states labelled by regular expressions. Further, every state also comes with its own *regular expression* which provides a pool of potential new threads that can arise.

7.2.1 Model specification

We assume that the reader is familiar with the notion of *regular languages* and *finite automata* [Sip06].

Let Σ be a finite alphabet and $Reg(\Sigma)$ be the set of all regular expressions over Σ. For all $r \in Reg(\Sigma)$ let L_r denote the regular language generated by the expression r. If $s, t \in \Sigma^*$ then $s \cdot t$ denotes the concatenation of strings s and t, often written as st. We say that a string $s \in \Sigma^*$ matches regular expression $r \in Reg(\Sigma)$ if $s \in L_r$.

Definition 7.8 (Regular agent transition system)**.** *Let $\mathcal{P}$ be a countable set of propositions. A* regular agent transition system *is given by $\mathcal{T} = (\mathcal{Q}, \Lambda, \gamma, \mu, \rho)$ where:*

[1] x and y cannot have same witness and hence at least two processes are required.

- *$\mathcal{Q}$ is a finite set of states.*

- *$\Lambda \subseteq_{fin} Reg(\Sigma)$ is a finite set of regular expressions.*

- *$\gamma \subseteq (\mathcal{Q} \times \Lambda \times \mathcal{Q})$ is the set of transitions labelled by regular expressions.*

- *$\mu : \mathcal{Q} \mapsto \Lambda$ where $\mu(q)$ describes the potential set of new processes that are created at q.*

- *$\rho : \mathcal{Q} \to 2^{\mathcal{P}}$ is the valuation of propositions at every state.*

From the regular agent transition system, we can obtain the configuration space of the process system with threads. The configuration of the system is given by a state along with the set of processes that are currently active. For any regular expression r if there is an r edge from q to q' in the transition system, then it means any process with id s which matches the regular expression r that is currently active can change the system configuration with the state being updated from q to q'.

Further, the new configuration will carry all previously active processes (strings) as active ids[2] along with a finite (unbounded) number of new threads created of the form $s \cdot t \in L_{\mu(q')}$. Each new string of the from st in the updated configuration indicates a new child thread created when the parent process s makes a transition. Note that even though the number of threads newly created is finite, we cannot bound the size beforehand. Hence, the number of new processes added is finite but unbounded [3]. The language of regular expressions is rich enough to consider tree structures of concurrent and sequential threads with forking, as well as process threads created within loops, perhaps while waiting for an external event to occur.

The input also needs to specify the initial configuration given by a state q_0 and a finite set of strings $s_1, \ldots, s_n \in \Sigma^*$ which are the process that are present (alive) at the start of the system.

[2]this corresponds having the condition that no thread can be reused, which corresponds to the monotonicity condition when viewed as PTML model

[3]this corresponds to non-deterministic fork of threads

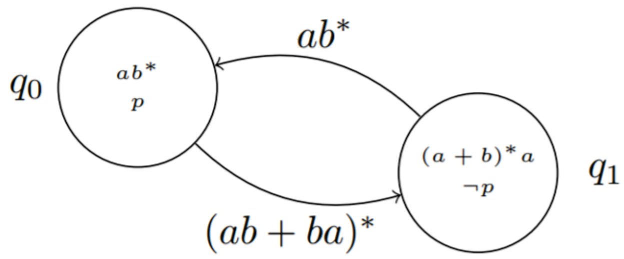


Figure 7.1: Illustration of a regular agent transition system. The valuation function ρ is not highlighted.

Definition 7.9. *Given regular agent transition system* $\mathcal{T} = (\mathcal{Q}, \Lambda, \gamma, \mu, \rho)$ *and an initial state* (q_0, A_0) *where* $q_0 \in \mathcal{Q}$ *and* $A_0 = \{s_1, \ldots, s_n\} \subseteq \Sigma^*$*. Define the configuration graph* $C_{\mathcal{T}} = (\mathcal{W}, \mathcal{R})$ *rooted at* (q_0, A_0) *to be the smallest tree such for every* $(q, A) \in \mathcal{W}$ *and every* $(q, r, q') \in \gamma$ *and* $s \in A$ *we have* $(q', A') \in \mathcal{W}$ *and* $\big((q, A),\ t,\ (q', A \cup A')\big) \in \mathcal{R}$ *where* $A' \subseteq_{fin} \{st \mid st \in L_{\mu(q')}\}$

For any configuration graph $C_{\mathcal{T}} = (\mathcal{W}, \mathcal{R})$ define the corresponding induced PTML model $\mathcal{M}_{\mathcal{T}} = (\mathcal{W}, \Sigma^*, \mathcal{R}, \delta, \rho)$ where for all $(q, A) \in \mathcal{W}$ define $\delta((q, A)) = A$ and $\rho((q, A)) = \rho(q)$.

7.2.2 Example

Consider the regular agent transition system $\mathcal{T} = (\mathcal{Q}, \Lambda, \gamma, \mu, \rho)$ defined in Fig. 7.1. The regular expressions on the edges denote the set of processes that can change the corresponding system state. The regular expressions inside a state q given by $\mu(q)$ states denote the potential pool of new processes that can be created when the system enters the state q.

Suppose the system starts at initial state q_0 and active processes $\{a, ab, ba\}$, then the corresponding configuration graph of the system is described in Fig. 7.2.

Note that ab is the only process that can make a move from the root since it is the only string that matches ab^*. However, ab itself can have unboundedly many

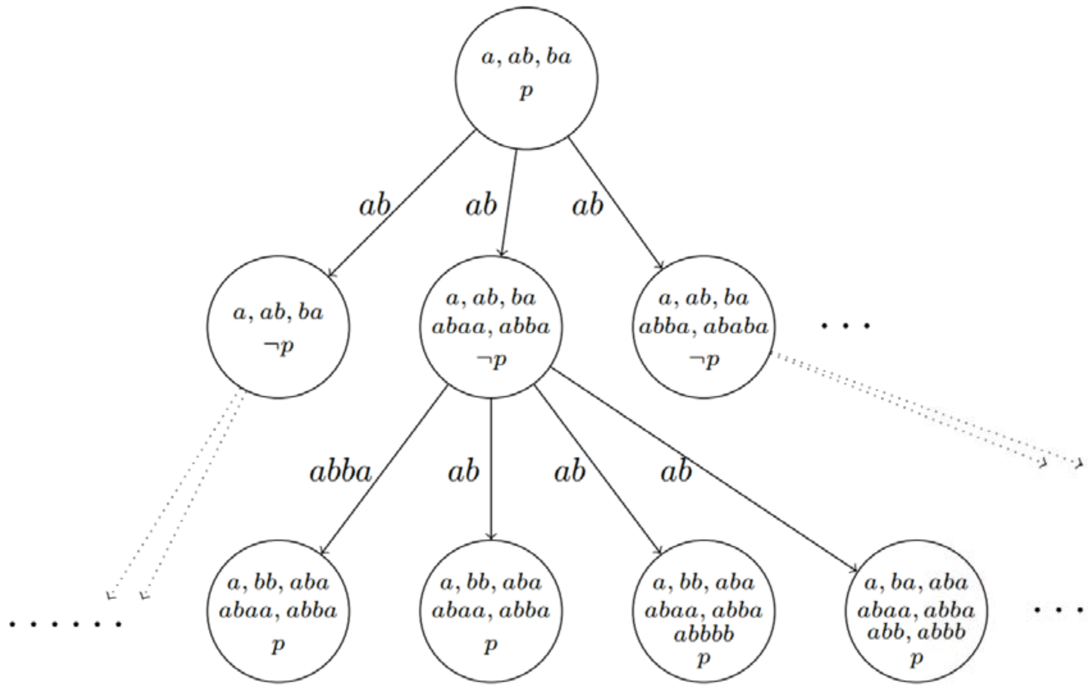


Figure 7.2: Configuration graph corresponding to transition system in Fig. 7.1 with $(q_0, \{a, ab, ba\})$ as the initial states. The dotted lines indicate more successors.

branches. In every such new world along with the old process set $\{a, ab, ba\}$ there are some finitely many new processes added which are of the form $ab \cdot s$ for some $s \in \Sigma^*$ such that $ab \cdot s$ matches $(a + b)^*a$. The figure highlights 3 such possible branches (one of them includes a state where there are no new processes created).

Now at the second level, consider the state where the set of live processes is $\{a, ab, ba, abaa, abba\}$ (the middle state). The possible processes that can make a move from this state are ab and $abba$ since these are the only strings that match $(ab+ba)^*$. Further, in the successors of $abba$, there are no new processes since there cannot be a string of the form $abba \cdot s$ that matches ab^*. On the other hand, the successors of ab are unbounded since there are infinitely many extensions of ab that matches ab^*.

Consider the formula $\forall x \forall y\ (\Diamond_x \neg p \to \Box_y \Diamond_x p)$. We will verify that this formula is true at the root.

First note that the precondition holds only for $[x \mapsto ab]$. For y, we need to verify only the case when $[y \mapsto ab]$ since other processes do not have any successors. But note all successors of ab from the root, have unboundedly many successors of ab.

On the other hand the formula $\forall x\ (\Box_x p \vee \Box_x \Box_x p)$ is false. In particular when $[x \mapsto ab]$, all ab successors we have at least one ab successor to $\neg p$. Hence the formula does not hold at the root.

7.2.3 Model checking

Note that for any regular agent transition system $\mathcal{T}$ the corresponding $\mathcal{M}_{\mathcal{T}}$ is an unbounded branching model that describes the configuration space of the run of the system starting from some initial configuration. In [PR17], we consider this model specification and prove that model checking monodic PTML formulas over such specification is decidable. In fact, we can prove that the model checking problem is decidable for PTML without monodic restriction.

Given an input $(\mathcal{T}, q_0, A_0)$ and a PTML sentence φ as an input, the regular model checking problem is the check whether $\mathcal{M}_{\mathcal{T}}, (q_0, A_0) \models \varphi$. Let $\mathcal{M}^n_{\mathcal{T}}$ denote the PTML induced by $\mathcal{T}$ but is restricted to height at most n.

Observation 7.10. *For any PTML formula φ and for any $\mathcal{M}_{\mathcal{T}}$ rooted at (q, A) we have $\mathcal{M}_{\mathcal{T}}, (q, A), \sigma \models \varphi$ iff $\mathcal{M}^{\mathsf{md}(\varphi)}_{\mathcal{T}}, (q, A), \sigma \models \varphi$.*

Thus, for any given PTML formula φ, it is enough to consider $\mathcal{M}^{\mathsf{md}(\varphi)}_{\mathcal{T}}$. To prove that the model checking problem problem is decidable, we prove that every induced model $\mathcal{M}^n_{\mathcal{T}}$ is bisimilar to a some model $\mathcal{N}^{\mathsf{md}(\varphi)}_{\mathcal{T}}$ of bounded size.

Now, by Theorem 6.3 (which states bisimulation implies elementary equivalence), it is enough to check if the formula φ is true in $\mathcal{N}^{\mathsf{md}(\varphi)}_{\mathcal{T}}$ which reduces to the model checking problem over finite models. Towards this, first we define an equivalence induced over agents and the worlds of $\mathcal{M}_{\mathcal{T}}^{n}$ and using this we define a filtration model.

Before defining the filtration, we recall some results about regular languages and finite automata. For any $r \in Reg(\Sigma)$ let A_r be the minimal deterministic automaton for L_r. For any finite set of regular expressions $\Lambda \subseteq Reg(\Sigma)$, and for all strings $s, t \in \Sigma^*$, we can define an equivalence $s \equiv_\Lambda t$ if for all $r \in \Lambda$, run of s in A_r ends in the same state as that of the run of t in A_r. For all $s \in \Sigma^*$, define $[\![s]\!]_\Lambda = \{t \mid \ s \equiv_\Lambda t\}$ and $[\![\Sigma^*]\!]_\Lambda = \{[\![s]\!]_\Lambda \mid s \in \Sigma^*\}$. Similarly, for any sets of strings $A, B \subseteq \Sigma^*$ define $A \equiv_\Lambda B$ if $\{[\![s]\!]_\Lambda \mid s \in A\} = \{[\![t]\!]_\Lambda \mid t \in B\}$.

Definition 7.11 (Filtration model)**.** *Let $\mathcal{T} = (\mathcal{Q}, \Lambda, \gamma, \mu, \rho)$ be any regular agent transition system and let (q_0, A_0) be the initial configuration and $\mathcal{M}_{\mathcal{T}}^{n} = (\mathcal{W}, \Sigma^*, \mathcal{R}, \delta, \rho)$ be the* PTML *model induced by $\mathcal{T}$ rooted at (q_0, A_0) restricted to height at most n.*

- *For all $(q, A),\ (q, B) \in \mathcal{W}$ define $(q, A) \simeq_{\mathcal{T}} (q', B)$ if (q, A) and (q', B) are at same height and $q = q'$ and $A \equiv_\Lambda B$.*

  For all $(q, A) \in \mathcal{W}$ define $[\![(q, A)]\!]_{\mathcal{T}} = \{(q, A') \mid (q, A) \simeq (q, A')\}$ and $[\![W]\!]_{\mathcal{T}} = \{[\![(q, A)]\!]_{\mathcal{T}} \mid (q, A) \in \mathcal{W}\}$.

  For all $A \subseteq \Sigma^*$ define $[\![A]\!]_{\mathcal{T}} = \{[\![s]\!]_\Lambda \mid s \in A\}$.

- *The filtration model of $\mathcal{M}_{\mathcal{T}}^{n}$ is given by $\mathcal{N}_{\mathcal{T}}^{n} = (W', D', R', \delta', \rho')$ as follows:*

  - $\mathcal{W}' = [\![W]\!]_{\mathcal{T}}$ *and* $\mathcal{D}' = [\![\Sigma^*]\!]_{\mathcal{T}}$.
  - $\mathcal{R}' = \{\big([\![(q, A)]\!]_{\mathcal{T}}, [\![s]\!]_\Lambda, [\![(q, B)]\!]_{\mathcal{T}}\big) \mid \big((q, A), s, (Q', B))\big) \in \mathcal{R}\}$.
  - $\delta'([\![(q, A)]\!]_{\mathcal{T}}) = [\![A]\!]_{\mathcal{T}}$.
  - $\rho'([\![(q, A)]\!]_{\mathcal{T}}) = \rho((q, A))$.

Note that $\mathcal{N}_{\mathcal{T}}^{n}$ is well defined. Now we prove that the filtration preserves bisimilarity.

Theorem 7.12. *Let $\mathcal{T} = (\mathcal{Q}, \Lambda, \gamma, \mu, \rho)$ be any regular agent transition system and let (q_0, A) be the initial configuration and $\mathcal{M}^n_{\mathcal{T}} = (\mathcal{W}, \Sigma^*, \mathcal{R}, \delta, \rho)$ be the PTML model induced by $\mathcal{T}$ rooted at (q_0, A) restricted to height at most n and $\mathcal{N}^n_{\mathcal{T}} = (W', D', R', \delta', \rho')$ be the corresponding filtration model. Then $(\mathcal{M}^n_{\mathcal{T}},\ (q_0, A))$ is bisimilar to $(\mathcal{N}^n_{\mathcal{T}},\ [\![(q_0, A)]\!]_{\mathcal{T}})$.*

Proof. First we define the bisimulation relation (G, H) where
$\mathrm{G} = \{\big((q, A), [\![(q, A)]\!]_{\mathcal{T}}\big) \mid (q, A) \in \mathcal{W}\}$ and for every $\big((q, A), [\![(q, A)]\!]_{\mathcal{T}}\big) \in \mathrm{G}$ define $\mathrm{H}_{\big((q,A),[\![(q,A)]\!]_{\mathcal{T}}\big)} = \{(s, [\![s]\!]_{\Lambda}) \mid s \in \delta((q, A))\}$.

We now show that (G, H) is a bisimulation by verifying all the required properties. Pick any $\big((q, A), [\![(q, A)]\!]_{\mathcal{T}}\big) \in \mathrm{G}$.

The condition [Val] holds since ρ depends only on q.

For condition (*agent-forth*), if $s \in \delta(q, A)$ then by definition $s \in A$ and hence $[\![s]\!]_{\Lambda} \in [\![A]\!]_{\mathcal{T}} = \delta'([\![(q, A)]\!]_{\mathcal{T}})$.

For condition (*agent-back*), suppose $[\![s]\!]_{\Lambda} \in \delta'([\![(q, A)]\!]_{\mathcal{T}})$ then first note that there is some $(q, A) \simeq_{\mathcal{T}} (q, A')$ and $s_1 \equiv_{\Lambda} s$ such that $[\![A_1]\!]_{\mathcal{T}} = [\![A]\!]_{\mathcal{T}}$ and $s_1 \in A_1$. This implies $[\![s_1]\!]_{\Lambda} \in [\![A]\!]_{\mathcal{T}}$ and hence $[\![s_1]\!]_{\Lambda} \in [\![A]\!]_{\mathcal{T}}$. Thus there is some $s_2 \in A$ such that $s_2 \equiv_S s$ and we have $(s_2, [\![s]\!]_{\Lambda}) \in \mathrm{H}_{\big((q,A),[\![(q,A)]\!]_{\mathcal{T}}\big)}$.

To verify *world forth and back* properties, pick any $(s, [\![s]\!]_{\Lambda}) \in H_{\big((q,A),[\![(q,A)]\!]_{\mathcal{T}}\big)}$.

For (*world-forth*), if $\big((q, A),\ s,\ (q', B)\big) \in \mathcal{R}$ then $([\![(q, A)]\!]_{\mathcal{T}}, [\![s]\!]_{\Lambda}, [\![(q', B)]\!]_{\mathcal{T}}) \in \mathcal{R}'$. Also, since $A \subseteq B$ we have $\mathrm{H}_{\big((q,A),[\![(q,A)]\!]_{\mathcal{T}}\big)} \subseteq \mathrm{H}_{\big((q',B),[\![(q',B)]\!]_{\mathcal{T}}\big)}$. Finally, by definition of G we have $\big((q', B), [\![(q', B)]\!]_{\mathcal{T}}\big) \in G$ and we are done.

For (*world-back*) condition, suppose $([\![(q, A)]\!]_{\mathcal{T}}, [\![s]\!]_{\Lambda}, [\![(q', B)]\!]_{\mathcal{T}}) \in R'$, then we need to find some $\big((q', B_1), [\![(q', B)]\!]_{\mathcal{T}}\big) \in G$ such that $(q, A) \xrightarrow{s} (q', B_1)$ and also $H_{\big((q,A),[\![(q,A)]\!]_{\mathcal{T}}\big)} \subseteq H_{\big((q',B_1),[\![(q',B)]\!]_{\mathcal{T}}\big)}$.

Now, since $(\llbracket(q,A)\rrbracket_{\mathcal{T}}, \llbracket s\rrbracket_\Lambda, \llbracket(q',B)\rrbracket_{\mathcal{T}}) \in R'$, by construction there is some transition $\big((q,A_1), s_1, (q',B_1)\big) \in \mathcal{R}$ such that $(q,A) \simeq (q,A_1)$ and $(q',B) \simeq (q',B_1)$ and $\llbracket s\rrbracket_\Lambda = \llbracket s_1\rrbracket_\Lambda$. Note that B_1 is of the form $A \cup B_2$ where $B_2 = \{s_1t_1, s_1t_2, \ldots, s_1t_n \mid s_1t_i \in L_{\mu(q')}\}$. Define $B_2' = \{st_1, s_1t_2, \ldots, st_n\}$ which is obtained by replacing s_1 by s (as prefix) in every string occurring in B_2. Let $A_2' = A \cup B_2'$.

First we verify that $\big((q',B), \llbracket(q',A_2')\rrbracket_{\mathcal{T}}\big) \in G$. For this it is enough to prove that $\llbracket B\rrbracket_\Lambda = \llbracket A_2'\rrbracket_\Lambda$. Pick any $t' \in A_1' \cup A_2'$. If $t' \in A$ then $\llbracket t'\rrbracket_\Lambda \in \llbracket A_1'\rrbracket_\Lambda \cap \llbracket A_2'\rrbracket_\Lambda$. Further if $t' \in B_1$ then t' is of the form s_1t_i and we have $st_i \in B_2$ such that $s_1t_i \equiv_\Lambda st_i$. On the other hand if $t' \in B_2$ then t' is of the form st_i and we have $s_1t_i \in B_1$ such that $s_1t_i \equiv_\Lambda st_i$. Thus, $\llbracket B\rrbracket_\Lambda = \llbracket A_2'\rrbracket_\Lambda$ and hence $\big((q',B), \llbracket(q',A_2')\rrbracket_{\mathcal{T}}\big) \in G$.

Now since $s_1 \equiv s$ and every $st_i \in L_{\mu(q')}$ we have $\big((q,A), s, (q',A_2')\big) \in \mathcal{R}$. Finally since $A \subseteq A_2'$we have $H_{\big((q,A),\llbracket(q,A)\rrbracket_{\mathcal{T}}\big)} \subseteq H_{\big((q',A_1'),\llbracket(q',A_1')\rrbracket_{\mathcal{T}}\big)}$.

□

Corollary 7.13. *Regular Model checking problem has a non-deterministic algorithm with $O(l \cdot 2^{2^{|\Lambda|}} \cdot |\mathcal{Q}|)$ time complexity where l is the length of φ and $\mathcal{Q}$ and Λ are the states and regular expressions mentioned in $\mathcal{T}$ respectively.*

Proof. Note that for any regular expression $r \in Reg(\Sigma)$ of length n the corresponding minimal deterministic automaton A^r has at most 2^n states. Hence, $\simeq_\Lambda$ and $\equiv_\Lambda$ induce a *bounded* partition where the size of $|D'| \leq 2^{|\Lambda|}$ and $|W'| \leq 2^{|\Lambda|} \cdot |\mathcal{Q}|$ where $\mathcal{Q}$ and Λ are the finite set of states and finite set of regular expressions specified in $\mathcal{T}$ respectively. Now the problem reduces to the case of model checking a formula over finite structures and the claim follows.

□

7.3 Discussion

We considered one candidate for finite specification for PTML models based on regular expressions. If we can encode predicate interpretation in states (again via automata) then the same results can be lifted to model checking TML.

One obvious question is to what happens to **Expression Complexity** and **Data Complexity** for the finitely specified models. Note that the filtration gives us a model whose size is exponential in the given regular agent transition system. Thus, it follows that the **Expression Complexity** and **Data Complexity** for PTML over this specification are in PTIME and 2-NEXPTIME respectively. However, the lower bound is open for all these model checking variants and needs to be pinned down.

To specify reachability properties, we need a transitive modality or temporal operators, which are not considered in this thesis. In [AAA+16], Abdulla et al introduce a new framework for dynamic database systems, whose reachability properties can be expressed in FOML with transitive modal operators. This provides a specification to represent infinite state relational transition systems and database with updates. TML can be similarly used as a candidate logic to express properties to be model checked.

Chapter 8

Conclusion

In this thesis we have studied propositional term modal logic (PTML). We showed that it is an undecidable logic even when restricted further syntactically. This motivated the study of decidable fragments for PTML and we identified a few: the monodic, bundled and 2-variable fragments.

As PTML is a modal logic, its bisimulation is of interest. We characterized elementary equivalence of PTML via bisimulation and studied associated algorithmic questions. We also gave a van Benthem like characterization for the implicitly quantified fragment of PTML.

We also studied model checking for PTML, though perhaps in less detail than what might be possible. For instance, one could add transitive modalities $\Box^*$ and $\Diamond^*$ to express reachability properties and the model checking problem remains decidable for PTML with the finite specification discussed in the thesis. There might be other way of finite representations which might be of interest in other unbounded agent systems which needs to be explored.

We extended many of the results to term modal logic and its fragments and discussed how having constants in the vocabulary influences these results.

There are some natural questions that arise from the results considered in the thesis for PTML. For instance, there is a gap between the known lower bound and upper bound for satisfiability problem for PTML^2. Also, the decidability status of $\mathsf{PTML}^2_=$ is open. Similarly, the decidability status of $\forall x \Box_x$ bundled fragment of TML over constant agent models is open. It is also not clear how equality affects the decidability results for the bundled fragments.

Term modal logic seems to be more expressive that FO but less expressive than FOML (since TML^2 is decidable but FOML^2 is undecidable). On the other hand, TML can be easily encoded in *second order logic*. Pinning down the exact expressiveness of TML is an interesting and open ended project to explore.

8.1 Future directions

Decidable fragments. There are many directions to look for in terms of decidable fragments for TML. For instance, the analogue of the fluted fragment of FO, fragments with bounded quantifier alternation and restricted quantifier prenex form etc are all interesting.

Correspondence theory. An important branch of modal logic is correspondence theory, by which frame conditions are characterized by modal formulas. Exploring various fragments of TML to see what frame conditions can they characterize would help us better understand the expressiveness of TML.

Finite model theory. Most of finite model theory literature is studied by fixing the vocabulary. A less studied aspect is when the finite structure comes with its own vocabulary. In such a setting we would like to assert properties like *there is some relation R such that* $\alpha(R)$ where α is a formula that mentions R as a predicate.

Note that this is different from second order logic; since here, we are not allowed to pick arbitrary interpretation for the quantified predicates. The structure fixes the interpretation for all the relations that come with it.

This formalization is closer to the quantification of modalities used in term modal logic. We believe that the results and proof techniques discussed in this thesis gives us some tools to study finite model theory of unbounded vocabulary.

Also note that there is a way to induce a linear order over the underlying agent set in TML using only *equality* (refer Lemma 4.8). Thus TML and its fragments can be explored as candidate logics to capture various complexity classes over unordered structures.

Database theory. *Unordered unranked trees* arise in the context of XML documents and relational database with updates. TML is a natural logic to express properties on *unranked, unordered trees* with edge labels (where every tree comes with its own set of potential labels). We believe that TML and its monadic second order variants can be used to study the notion of *regular languages* over such trees.

Infinite state systems. In [AAA+16], Abdulla *et al* introduce a new framework for dynamic database systems. Term modal logic is a natural candidate to specify properties in such dynamic database systems. The freedom of having edge labels being specified by the structure may help us to better model the system where edge labels can encode information about who updates the database. Adding temporal operators to TML or having transitive frame restrictions over TML can particularly lead to interesting applications for verification of infinite state systems and for use in database query languages.

SUMMARY

Modal logic has been ubiquitously used in many fields of computer science including verification, epistemic logic etc. Typically we have two modal operators $\Box$ and $\Diamond$ which in a broad sense refers to *necessity* and *possibility* respectively. For instance, $\Box_i\alpha$ in an epistemic setting means that "Reasoner i knows that alpha". Similarly, $\Diamond_i\alpha$ in the context of a system of processes is interpreted as "Process i can possibly change the system configuration to a state where α holds". These reasoners or process index are referred to as agents in general.

Classically, the number of agents is assumed to be fixed and finite. But in many settings like multi-process systems / client-server systems / systems with unboundedly many reasoners, we cannot fix the agent set beforehand. The active agents change not only from one model to the other but also from one state to the other in the same model. For instance, in multi-process systems, when the system configuration changes, some processes may be terminated and some new ones may be created.

Term modal logic introduced by Fitting et.al is suitable to study such settings, where we can state properties like $\exists x \forall y \, \Box_x \Box_y \alpha$ which in the epistemic setting translates to "there exists some agent who knows that everybody knows that α".

In this thesis we will explore three main aspects for term modal logic:
(1) Satisfiability problem (2) Bisimulation and (3) Model checking problem.

Satisfiability problem Surprisingly restriction to propositional fragment is of no help. In fact we prove that TML satisfiability problem is undecidable even when $(\top, \bot)$ are used as atoms. Using reductions of tiling problems, we strengthen the result further that the FinSat, UnSat and InfAx are mutually recursively inseparable for TML with atoms restricted only to $e$quality.

These undecidability results motivate us to identify decidable fragments. In this thesis, we identify some decidable fragments of term modal logic: the monodic fragment, the bundled fragment and the two variable fragment.

Bisimulation characterizes modal logics model theoretically. We introduce bisimulation for propositional term modal logic, and prove that it preserves elementary equivalence and the converse holds over image finite models.

Further, we discuss van-benthem type invariance theorem for the variable free fragment called the *implicitly quantified modal logic*. We also tailor the bisimulation to different fragments of TML and use this to compare their expressiveness.

Model Checking When we consider the model checking problem for term modal logic, it is clear that it reduces to classical model checking of First order logic when models are finite, and only complexity issues are interesting. We present these, considering the variants where the model is fixed, or the formula is fixed, or when both are inputs.

When the model is infinite, we need a finite representation to provide input to the algorithm. We consider models where agents are specified as *regular expressions*. These specifications are motivated by consideration of how process identifiers are created in dynamic systems of processes. For such specification, we show that model checking is decidable.

Bibliography

[AAA+16] Parosh Aziz Abdulla, C Aiswarya, Mohamed Faouzi Atig, Marco Montali, and Othmane Rezine. Recency-bounded verification of dynamic database-driven systems. In *Proceedings of the 35th ACM Symposium on Principles of Database Systems*, 2016.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697. ACM, 2016.

[Bal14] Roberta Ballarin. Modern origins of modal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2014 edition, 2014.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic (Cambridge Tracts in Theoretical Computer Science)*. Cambridge University Press, 2001.

[Ben10a] Johan van Benthem. Frame correspondences in modal predicate logic. *Proofs, categories and computations: Essays in honor of Grigori Mints*, pages 1–14, 2010.

[Ben10b] Johan van Benthem. Modal logic for open minds. 2010.

[EFT13] H-D Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical logic*. Springer Science & Business Media, 2013.

[FHMV04] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Vardi. *Reasoning About Knowledge*. A Bradford Book, 2004.

[Fin75] Kit Fine. Normal forms in modal logic. *Notre Dame journal of formal logic*, 16(2):229–237, 1975.

[FL79] Michael J Fischer and Richard E Ladner. Propositional dynamic logic of regular programs. *Journal of computer and system sciences*, 18(2):194–211, 1979.

[FM99] Melvin Fitting and Richard L. Mendelsohn. *First-Order Modal Logic (Synthese Library)*. Springer, 1999.

[FTV01] Melvin Fitting, Lars Thalmann, and Andrei Voronkov. Term-modal logics. *Studia Logica*, 69(1):133–169, 2001.

[GH93] Adam J Grove and Joseph Y Halpern. Naming and identity in epistemic logics part i: the propositional case. *Journal of Logic and Computation*, 3(4):345–378, 1993.

[GK72] Yu Sh Gurevich and IO Koryakov. Remarks on berger's paper on the domino problem. *Siberian Mathematical Journal*, 13(2):319–321, 1972.

[GKV97] Erich Grädel, Phokion G Kolaitis, and Moshe Y Vardi. On the decision problem for two-variable first-order logic. *Bulletin of symbolic logic*, 3(1):53–69, 1997.

[GO99] Erich Grädel and Martin Otto. On logics with two variables. *Theoretical computer science*, 224(1-2):73–113, 1999.

[GO07] Valentin Goranko and Martin Otto. Model theory of modal logic. In *Studies in Logic and Practical Reasoning*, volume 3, pages 249–329. Elsevier, 2007.

[Göd33] Kurt Gödel. Zum entscheidungsproblem des logischen funktionenkalküls. *Monatshefte für Mathematik und Physik*, 40(1):433–443, 1933.

[Gro95] Adam J Grove. Naming and identity in epistemic logic part ii: a first-order logic for naming. *Artificial Intelligence*, 74(2):311–350, 1995.

[HC96] MJ Hughes and GE Cresswell. *A New Introduction to Modal Logic. Routledge. 1996.* Routledge, 1996.

[Koo07] Barteld Kooi. Dynamic term-modal logic. In *A Meeting of the Minds*, pages 173–186, 2007.

[KP18] Md Aquil Khan and Vineeta Singh Patel. A formal study of a generalized rough set model based on relative approximations. In *International Joint Conference on Rough Sets*, pages 502–510. Springer, 2018.

[Kri62] Saul A. Kripke. The undecidability of monadic modal quantification theory. *Mathematical Logic Quarterly*, 8(2):113–116, 1962.

[Mor75] Michael Mortimer. On languages with two variables. *Mathematical Logic Quarterly*, 21(1):135–140, 1975.

[OC17] Eugenio Orlandelli and Giovanna Corsi. Decidable term-modal logics. In *15th European Conference on Multi-Agent Systems*, 2017.

[OR94] Martin J Osborne and Ariel Rubinstein. *A course in game theory.* MIT press, 1994.

[PG92] Wolfram Pohlers and Thomas Glaß. *An introduction to mathematical logic.* Westf. Wilh.-Univ. Münster, Institut für Math. Logik u. Grundlagenforschung, 1992.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57. IEEE, 1977.

[PR14] Soumya Paul and R Ramanujam. Subgames within large games and the heuristic of imitation. *Studia Logica*, 102(2):361–388, 2014.

[PR17] Anantha Padmanabha and R. Ramanujam. Model checking a logic over systems with regular sets of processes. *Developmental Aspects of Intelligent Adaptive Systems (Innovations in Software Engineering), CEUR Workshop Proceedings*, Vol. 1819, 2017.

[PR19a] Anantha Padmanabha and R. Ramanujam. The monodic fragment of propositional term modal logic. *Studia Logica*, 107(3):533–557, Jun 2019.

[PR19b] Anantha Padmanabha and R. Ramanujam. Propositional modal logic with implicit modal quantification. In *Indian Conference on Logic and Its Applications*, pages 6–17. Springer, 2019.

[PR19c] Anantha Padmanabha and R. Ramanujam. Two variable fragment of Term Modal Logic. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[PRW18] Anantha Padmanabha, R. Ramanujam, and Yanjing Wang. Bundled fragments of first-order modal logic: (un)decidability. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, pages 43:1–43:20, 2018.

[RS17] 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.

[Sht18] Gennady Shtakser. Propositional epistemic logics with quantification over agents of knowledge. *Studia Logica*, 106(2):311–344, 2018.

[Sip06] Michael Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

[vDvdHK07] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic: 337 (Synthese Library)*. Springer Netherlands, 2007.

[Wan17] Yanjing Wang. A new modal framework for epistemic logic. In *Proceedings Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2017, Liverpool, UK, 24-26 July 2017.*, pages 515–534, 2017.

[WS18] Yanjing Wang and Jeremy Seligman. When names are not commonly known: Epistemic logic with assignments. In *Advances in Modal Logic Vol. 12 (2018): 611-628, College Publications*, 2018.

[WZ01] Frank Wolter and Michael Zakharyaschev. Decidable fragments of first-order modal logics. *The Journal of Symbolic Logic*, 66(3):1415–1438, 2001.