Available online at www.sciencedirect.com
ScienceDirect
Annals of Pure and Applied Logic 00 (2026) 1–34
Annals of Pure and Applied Logic

A modal approach towards substitutions

Yaxin Tu[a], Sujata Ghosh[b], Fenrong Liu[c,d], Dazhu Li[e,f,*]

[a]*Department of Computer Science, Princeton University, Princeton, USA*
[b]*Indian Statistical Institute, Chennai, India*
[c]*The Tsinghua-UvA JRC for Logic, Department of Philosophy, Tsinghua University, Beijing, China*
[d]*Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands*
[e]*Institute of Philosophy, Chinese Academy of Sciences, Beijing, China*
[f]*Department of Philosophy, University of Chinese Academy of Sciences, Beijing, China*

Abstract

Substitutions play a crucial role in a wide range of contexts, from analyzing the dynamics of social opinions and conducting mathematical computations to engaging in game-theoretical analysis. For many situations, considering one-step substitutions is often adequate. Yet, for more complex cases, iterative substitutions become indispensable. In this article, our primary focus is to study logical frameworks that model both single-step and iterative substitutions. We explore a number of properties of these logics, including their expressive strength, Hilbert-style proof systems, and satisfiability problems. Additionally, we establish connections between our proposed frameworks and relevant existing ones in the literature. For instance, we precisely delineate the relationship between single-step substitutions and the standard syntactic replacements commonly found in many classical logics. Moreover, special emphasis is placed on iterative substitutions. In this context, we compare our proposed framework with existing ones involving iterative reasoning, thereby highlighting the advantages of our proposal.

Keywords: Substitutions, replacements, iterative reasoning, modal logic, satisfiability problem

1. Substitutions as a ubiquitous mechanism

Many phenomena have substitutions as a core mechanism. Examples are abound in various fields, ranging from everyday social interactions to theoretical computations, including mathematical calculations, game-theoretical analysis as well as logic operations. In this article, instead of studying the concrete manifestations of substitutions in specific fields, we aim to explore logical frameworks to reason about substitutions themselves. This not only improves our understanding of a fundamental technique utilized across various fields, but also provides us with a uniform tool for these investigations. To set the groundwork for this discussion, let us first clarify our approach to substitutions.

There can be different kinds of substitutions. For illustration, let us take the logical operation $\varphi[\psi/p]$ as an example, which is common in a number of classical frameworks, including propositional logic, modal logic and first-order logic. With this operation, a new formula is obtained by replacing all occurrences of p in φ with ψ. An important feature of the operation is that it does not affect the semantic extension of a fixed formula: for instance, given a valuation function V and a propositional letter q, the extension $V(q)$ of q is never changed by the operation. So, the operation $\varphi[\psi/p]$ is essentially a syntactic update. However, depending on specific applications, it is equally natural to work with substitutions on a semantic level. It enables us to change the semantic extensions of formulas,

*Corresponding author
Email address: lidazhu@ucas.ac.cn (Dazhu Li)

especially when we need to encode dynamic information with fixed formulas.[1] To illustrate this and the ubiquity of substitutions, let us now present some concrete examples from different areas mentioned above.

Scenario 1: single-step substitutions in belief diffusion Consider the social community depicted in Figure 1, where three agents, a, b and c, are friends of each other. Friendships are represented by directed arrows. Also, we use atomic propositions to annotate beliefs or opinions: for instance, in Stage 1 of Figure 1, agents a and b hold the belief p, while c does not. As usual, modalities $\Box$ and $\Diamond$ are used to describe the properties of friends.[2] As suggested in e.g., [1, 2], beliefs of agents can be affected by their friends (due to conformity or peer pressure, for instance). Consider the following policy on the updates of beliefs:

An agent holds the belief p in the next stage if, and only if, (i) all her friends hold the belief p at the current stage or (ii) she believes p and some of her friends also have the same belief.

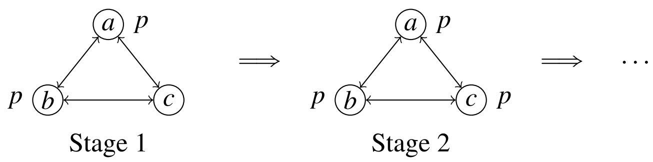

Figure 1. The belief status in a community that stabilizes.

Formally, the mechanism of the policy has an essence of substitutions, which can be written specifically as follows:

$$p := \Box p \vee (\Diamond p \wedge p)$$

This implies that *the agents who will believe p in the next stage are exactly those who currently has the property* $\Box p \vee (\Diamond p \wedge p)$. With the formal description, one can see that the set of agents who believe p at Stage 2 would become $\{a, b, c\}$, evolving from the initial set of $\{a, b\}$. Afterwards, the belief status among the agents stabilizes, which is exactly the transformation required by the update policy.[3] So, the diffusion of beliefs in this way is essentially an embodiment of the abstract process of substitutions.

Given the ubiquity of phenomena involving substitutions, the single-step updates designed above might appear somewhat limited. Indeed, many situations call for *iterative* reasoning. Let us consider the following case to illustrate this point:

Scenario 2: iterative substitutions in backward induction Consider a simple 2-player board game, denoted as $\mathcal{G} = (\mathcal{B}, s_0)$, involving *player* 0 and *player* 1. $\mathcal{B} = (W, R)$ is a finite board, consisting of finitely many nodes W and a binary relation R among nodes, and $s_0 \in W$ is a node indicating the current position of a *token* 't'. In each *round*, players move the token alternately, following the arrows in R, with player 0 acting first. Moreover, a player *wins* whenever the opponent cannot proceed with a move. When the game runs infinitely, player 0 wins.

Let us consider a concrete example with the game board $\mathcal{B}_1$ depicted in Figure 2. It is simple to see that when the token t is originally placed at a, player 0 has a winning strategy: whenever it is her turn, she just needs to move the token 'downwards'; while when t is originally placed at c, player 1 wins. More generally, on a finite $\mathcal{B} = (W, R)$, we can conduct backward induction to determine *winning positions* for players: a node s is a winning position for a player if, and only if, when the initial position of the token t is s, the player can win. For instance, for player 1, the procedure goes as follows:[4]

[1] In what follows, to distinguish between these two approaches, we will often call the syntactic substitutions '*replacements*' and the semantic ones '*substitutions*'.

[2] For instance, agent c has the property $\Box p$, i.e., *all c's friends* have the belief p, while agent a has the property $\Diamond p$, meaning that she has *at least one friend* with the belief p.

[3] Generally, the belief status of agents in a community may not always stabilize. To illustrate this, consider a community comprising two agents who are friends with each other, where one holds the belief p and the other does not. Under the given update policy, their belief statuses will enter a loop, continually alternating rather than reaching a stable state. [3] develops formal tools for the phenomena of oscillations. The substitution operations laid out above are important to understand these dynamics.

[4] By the *Gale-Stewart Theorem* [4], a node on a board is either a winning position of player 0 or a winning position of player 1. So, the complement of the set of winning positions of player 1 is the set of winning positions of player 0.

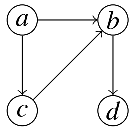

Figure 2. A game board $\mathcal{B}_1$

Starting from the set S_0 of dead ends on the board, determine the set S_1 such that along arrows in R, any state s_1 of S_1 can only reach states s' that have access to some state in S_0; then based on $S_0 \cup S_1$, determine the set S_2 such that S_2 is to $S_0 \cup S_1$ as S_1 is to S_0; and repeat this procedure up to some finite times.[5]

Again, this shares an obvious substitution core, but now it is more complex than that for Scenario 1 and has an iterative form as follows:

$$p := \Box\bot; (p := p \vee \Box\Diamond p)^*$$

reading that *first set p to be the dead ends and then iteratively re-define the p-states (of the next stage) as the states with the property $p \vee \Box\Diamond p$ (at the current stage) for finitely many times*. Such a form of iterated substitutions is first introduced in [5], and for an illustration of the match between this formal definition and the backward induction procedure, see Figure 3, and we defer the precise proof to Section 4.2.[6]

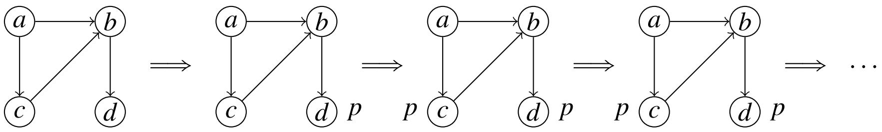

Figure 3. With the description $p := \Box\bot; (p := p \vee \Box\Diamond p)^*$, we have the transformation sequence $\varnothing \Rightarrow \{d\} \Rightarrow \{c, d\} \Rightarrow \{c, d\} \Rightarrow \ldots$ for the p-states, which converges to $\{c, d\}$. This is exactly the backward induction reasoning of the winning positions of player 1.

Although the above scenarios have been studied in different settings across existing literature, adopting a perspective centered on substitutions offers a unified approach to these diverse cases. To understand the properties of the substitution cores in these examples, we are going to explore logical frameworks with both styles of substitutions: the single-step and the iterative computations. These will be integrated into standard modal logic as additional modalities. Beyond the widespread application of substitutions, as illustrated in the examples, they also play a crucial role in numerous other logical systems. The remainder of this section will discuss related work and provide a concise summary of the contributions made by this article.

Related work Here are several lines of research that are closely relevant to our work:

Logics for the changes of local fact Generalizing the idea of propositional control in [6], [7] studies the logic of local fact change that extends the standard modal logic with an operator $\bigcirc\varphi$ stating that after replacing the propositional letters true at the current state with some set of propositional letters, φ is true at that state. As established in [7], the logic has an undecidable satisfiability problem. Notably, there are several distinctions between this approach and our current work. A key difference is that the updates in this logic are implicit; they are not explicitly described by an update operator. Additionally, the set of propositional letters involved in an update could be infinite, and in general, the updates may be undefinable using a formula within the logic itself.

Another relevant direction is the study on the dynamic logic of propositional assignments [8], which contains modalities $\langle p \leftarrow \varphi \rangle$ that reset the truth value of p at the current states to the truth value of φ. Different from our work, the truth values of propositional variables at other states cannot be changed by the modality.

[5]Here $2^{|W|}$ is a proper upper bound for the repetitions in the backward induction process ($|W|$ is *the cardinal number* of W), since after $2^{|W|}$ times we will repeat some previous step.

[6]One may observe that the backward induction described can also be formalized by the modal μ-calculus (see Section 4.2), and we will show that our proposal with the substitution operators are more capable to reason about the details of the game (Section 5).

Substitutions as syntactic modalities To avoid making substitutions a notion of the meta-language, [9] treats one-step substitutions as modalities and adds them to propositional logic, modal logic and first-order logic, respectively. For the resulting logics, [9] develops complete proof systems. This method contrasts with our framework, as the substitution modalities in [9] are syntactic; they function to alter the formulas themselves rather than to update the models. See [10] for a proceedings version of [9], which uses one-step substitutions as modalities in the setting of first-order logic.

Dynamic-epistemic logics Another important tradition is dynamic-epistemic logic DEL (e.g., [11, 12]) that removes states from models (e.g., the public announcement logic PAL [13]) or changes models with product updates [14], to capture epistemic (or doxastic) updates induced by informative changes or factual changes. Moreover, [15] studies a logic of iterative modal relativization that involves iterative generalizations of public announcement operators in PAL and [16] explores epistemic logic with a predicate language that contains operators to update values of terms.

It is important to emphasize that the notion of substitution has been explored in DEL, at both implicit level (using substitution as a way to define models) and the usual explicit level (incorporating expressions for substitution into logical languages). For instance, [14] considers an action for 'flip' that changes the truth set of a propositional variable in a model to its complement. More generally, [17, 18, 19] use substitution as a method to capture how an event changes the valuation for propositional variables in models describing the same. They also discuss the simultaneous version of substitution that changes the valuation for two or more propositional variables at the same time. More specifically, [20] adds explicit operators for single-step substitutions, called 'public assignments', to PAL and discusses operators for simultaneous single-step substitution. A sound and complete Hilbert-style proof system for the logic extending PAL with the simultaneous single-step substitution operators is given in [21], and [22] explores its further extension with various operators for common knowledge (with models in which knowledge is represented by arbitrary relations). Finally, we would like to point out that [21] and [20] provide further observations that are relevant to some of our results (cf. Proposition 2.3, Proposition 2.5, Theorem 3.6, Theorem 4.6). In the course of this paper, we will compare the relevant concepts/results to indicate the detailed connections and show how we improve or complement the results in [20, 21] (see Remark 2.6 and Remark 3.7).

Needless to say, in addition to the work mentioned above, there are quite a few milestone contributions that have close connections to our proposed frameworks, including the *modal μ-calculus* [23, 24], *the infinitary modal logic* [25], the *propositional dynamic logic* [26] and *iterative modal relativization* [15]. These logics also involve iterated and/or infinite computations. For a better understanding of our proposal, we would explore the precise connections between these frameworks and ours in the later sections.

Our main goal in the article is to explore the framework, *Modal Iterative Substitution Logic* MISL, with both single-step substitutions and their iterative generalizations. As a starting point, we will begin with a comparatively simpler setting, *Modal Substitution Logic* MSL. The rest of the article is organized as follows:

- Section 2 lays out the basics of MSL, whose language extends the standard modal language with single-step substitution operators $\langle p := \psi \rangle$. The resulting logic enjoys a number of desired properties. With the techniques of recursion axioms developed in dynamic-epistemic logic that enable us to reduce MSL-formulas to standard modal logic formulas, we offer a complete proof system for MSL. Based on this axiomatization, we also establish a decidability result for the satisfiability problem of the logic.

- Although our substitution operators are different from ordinary replacements $\varphi[\psi/p]$ on many levels, they are also closely related to each other, and a witness to this is given by the axiomatization in Section 2. In Section 3 we determine the precise condition under which they are modally equivalent.

- After having an understanding of the basic framework, in Section 4 we move on to the more intricate case MISL. Section 4.1 discusses various validities concerning the generalizations of the schematic principles of MSL. Section 4.2 analyzes the applications of the resulting logic to our motivating examples and other relevant game-theoretic notions in the literature.

- Next, motivated by the observation on the applications of MISL, we place our logic at a broader setting and study the formal connections between MISL and its cousins with the flavor of iterative computations in Section 5, which also demonstrates the merits of our proposal.

- In Section 6 we systematically investigate various properties of MISL, involving its expressive power and computational behavior.

  W.r.t. expressiveness, although MISL is much more powerful than the standard modal logic, Section 6.1 shows that it is still invariant under the notion of standard bisimulation for modal logic.

  On the other hand, the iterative substitution operators increase the computational complexity of MISL drastically. Section 6.2 shows that the logic does not have the finite model property, and develops suitable upper bound and lower bound for the satisfiability problem of MISL, which illustrate that the logic is Σ^1_1-complete (hence, undecidable). Moreover, Section 6.3 shows that the satisfiability problem of the logic is undecidable even when we confine ourselves to very simple classes of models (e.g., finite tree models).

- Finally, Section 7 concludes the article with several directions that deserve to be explored in future.

2. A basic logic for single-step substitutions: axiomatization and decidability

Let us now present a formal proposal to explore the nature of substitutions. This section focuses on a comparatively simple framework, *Modal Substitution Logic* (MSL), that only contains substitution operators for single-step updates. For the logic, we develop a complete Hilbert-style proof system and prove its decidability.

First of all, let us define the language of MSL, which extends *the language* $\mathcal{L}_{\mathsf{ML}}$ *of the standard modal logic* ML in the following manner:

Definition 2.1 (Language $\mathcal{L}_{\mathsf{MSL}}$)**.** *Let* $\mathbf{P}$ *be a countable set of propositional letters. The* language $\mathcal{L}_{\mathsf{MSL}}$ of MSL *is given by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \langle p := \varphi\rangle\varphi$$

where $p \in \mathbf{P}$*. Abbreviations* $\top$*,* $\bot$*,* $\vee$ *and* $\to$ *are as usual, so are the box operators* $\Box$ *and* $[p := \varphi]$ *for* $\Diamond$ *and* $\langle p := \varphi\rangle$ *respectively.*

The readings of basic modal formulas are as usual, and the formula $\langle p := \psi\rangle\varphi$ states that *after we replace the truth set of* p *with the truth set of* ψ*,* φ *is the case.* The truth set of a formula in a model consists of the states where the formula is true. This can be made precise after we introduce the semantics, and for now, let us first define some syntactic notions. Given a formula $\langle p := \psi\rangle\varphi$, the propositional letter p and formula φ are the *pivot* and the *scope* of the substitution operator $\langle p := \psi\rangle$, respectively. Similarly for the box version $[p := \psi]$ in a formula $[p := \psi]\varphi$. Also, we sometimes write $\langle p := \psi_1; q := \psi_2\rangle\varphi$ for $\langle p := \psi_1\rangle\langle q := \psi_2\rangle\varphi$, and $[p := \psi_1; q := \psi_2]\varphi$ for $[p := \psi_1][q := \psi_2]\varphi$. The notion of *subformulas* for MSL, written as $\mathsf{Sub}(\varphi)$, extends that for the standard modal logic with the following:

$$\mathsf{Sub}(\langle p := \psi\rangle\varphi) := \mathsf{Sub}(\psi) \cup \mathsf{Sub}(\varphi) \cup \{\langle p := \psi\rangle\varphi\}.$$

As in the case for ML, the formulas of $\mathcal{L}_{\mathsf{MSL}}$ are evaluated in *Kripke models* $\boldsymbol{M} = (W, R, V)$, where W is a non-empty set of states, $R \subseteq W \times W$ is a binary relation, and $V : \mathbf{P} \to 2^W$ is a valuation function. As usual, for any $w \in W$, $(\boldsymbol{M}, w)$ is a *pointed model*, and for simplicity, we often write $\boldsymbol{M}, w$. Moreover, *frames* $\boldsymbol{F} = (W, R)$ are pairs without valuations. We also write $w \in \boldsymbol{M}$ or $w \in \boldsymbol{F}$ when $w \in W$.

Definition 2.2 (Semantics)**.** *Let* $\boldsymbol{M} = (W, R, V)$ *be a model and* $w \in W$*.* Truth of MSL-formulas φ at $(\boldsymbol{M}, w)$*, written as* $\boldsymbol{M}, w \vDash_{\mathsf{MSL}} \varphi$*, is given recursively as follows:*

$$\begin{array}{rcl}
\boldsymbol{M}, w \vDash_{\mathsf{MSL}} p & \Leftrightarrow & w \in V(p) \\
\boldsymbol{M}, w \vDash_{\mathsf{MSL}} \neg\varphi & \Leftrightarrow & \boldsymbol{M}, w \nvDash_{\mathsf{MSL}} \varphi \\
\boldsymbol{M}, w \vDash_{\mathsf{MSL}} \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \boldsymbol{M}, w \vDash_{\mathsf{MSL}} \varphi_1 \text{ and } \boldsymbol{M}, w \vDash_{\mathsf{MSL}} \varphi_2 \\
\boldsymbol{M}, w \vDash_{\mathsf{MSL}} \Diamond\varphi & \Leftrightarrow & \boldsymbol{M}, v \vDash_{\mathsf{MSL}} \varphi \text{ for some } v \in W \text{ s.t. } Rwv \\
\boldsymbol{M}, w \vDash_{\mathsf{MSL}} \langle p := \psi\rangle\varphi & \Leftrightarrow & \boldsymbol{M}|_{p:=\psi}, w \vDash_{\mathsf{MSL}} \varphi
\end{array}$$

where $\boldsymbol{M}|_{p:=\psi} = (W, R, V|_{p:=\psi})$ *is obtained through model transformation from* $\boldsymbol{M}$ *and may disagree with* $\boldsymbol{M}$ *only on the valuation of* p*, where* $V|_{p:=\psi}(p) = \{v \in W : \boldsymbol{M}, v \vDash_{\mathsf{MSL}} \psi\}$*.*

For any model $\boldsymbol{M}$ and formula φ, the *truth set of φ in $\boldsymbol{M}$* is defined as $[\![\varphi]\!]^{\boldsymbol{M}} := \{w \in \boldsymbol{M} : \boldsymbol{M}, w \vDash_{\mathsf{MSL}} \varphi\}$, and we often omit the superscript for the model $\boldsymbol{M}$ when it is clear from the context. Also, notions such as *satisfiability*, *validity* and *logical consequence* are defined as usual [27]. We note that substitutions are, to some extent, unidirectional: we cannot trace back to the status of valuations before the substitution after the substitution is carried out. For instance, $\langle p := \Box q\rangle\Box q \to p$ is not a validity, and for a counterexample, see Figure 4.

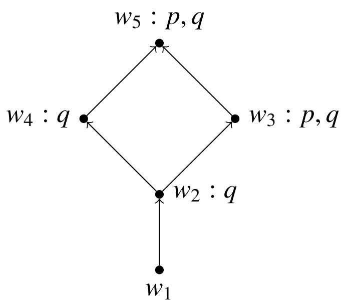

Figure 4. A model $\boldsymbol{M}$ in which $\langle p := \Box q\rangle\Box q \to p$ is false at w_1. In the model, formula $\Box q$ is true everywhere. So, after the substitution, p is true everywhere, and in particular, $\Box q$ is still true at w_1 in the new model. However, as we can see, w_1 does not have the property p in $\boldsymbol{M}$.

Conventions Many logical systems will be involved in the article. For each logical system, we will use its name as a subscript of $\vDash$ and $\mathcal{L}$, respectively, to express the satisfaction relation of the logic and its language: for instance, we wrote $\vDash_{\mathsf{MSL}}$ for the satisfaction relation of MSL and $\mathcal{L}_{\mathsf{MSL}}$ for its language.

MSL has many desired properties, among which the following is an important one:

Proof system Table 1 presents a calculus **MSL** for MSL. It is a direct extension of proof system **ML** for ML, which consists of (A1)-(A3), (Dual), ($\mathrm{K}_\Box$), (MP) and ($\mathrm{Nec}_\Box$). The axiom ($\mathrm{K}_{[\,]}$) shows that we can distribute an arbitrary substitution operator over an implication. Also, similar to the case for dynamic epistemic logic (DEL) [11, 12, 13, 14], we have the recursion axioms that enable us to reduce MSL-formulas to equivalent ML-formulas: (R1) and (R2) show how we can eliminate substitution operators when their scopes are propositional letters, (R3) indicates that substitution operators are self-dual, (R4) shows that we can distribute the operators over conjunction, and finally (R5) ensures that the substitution operators can be pushed inside the scope of a standard modality $\Box$. In the same convention, we will write $\vdash_{\mathbf{MSL}} \varphi$ if φ is provable in **MSL**, and write $\vdash_{\mathbf{ML}} \varphi$ if φ is provable in **ML**.

To deepen the understanding of how this calculus works, let us explore some technical properties of the device.

Proposition 2.3. *The following rule is derivable:*

$$\text{From } \chi_1 \leftrightarrow \chi_2, \text{ infer } [p := \varphi]\chi_1 \leftrightarrow [p := \varphi]\chi_2.$$

Proof. It suffices to show that $\vdash_{\mathbf{MSL}} \chi_1 \to \chi_2$ implies $\vdash_{\mathbf{MSL}} [p := \varphi]\chi_1 \to [p := \varphi]\chi_2$. The proof is routine, and it goes as follows:

(a)	$\chi_1 \to \chi_2$	Assumption
(b)	$[p := \varphi](\chi_1 \to \chi_2)$	($\mathrm{Nec}_{[\,]}$), (a)
(c)	$[p := \varphi](\chi_1 \to \chi_2) \to ([p := \varphi]\chi_1 \to [p := \varphi]\chi_2)$	($\mathrm{K}_{[\,]}$)
(d)	$[p := \varphi]\chi_1 \to [p := \varphi]\chi_2$	(MP), (b), (c)

This completes the proof. □

Next, as in the case for DEL, with the help of the recursion axioms (R1)-(R5), we can show the following:

Proposition 2.4. *For any $\varphi \in \mathcal{L}_{\mathsf{MSL}}$, there is a standard modal formula $\varphi_{\mathsf{ML}} \in \mathcal{L}_{\mathsf{ML}}$ such that $\vdash_{\mathbf{MSL}} \varphi \leftrightarrow \varphi_{\mathsf{ML}}$.*

Proof. Let φ be an MSL-formula. Starting from an innermost occurrence of a substitution operator, we can push it inside Boolean connectives and the standard modality $\Box$, which can finally give us a formula in which the substitution operator is eliminated. Repeating this procedure, we can finally get a standard modal formula φ_{ML} that is provably equivalent to the original φ, for which we may need to use propositional logic and the rule in Proposition 2.3. □

Proof system **MSL**	
Basic axioms:	
(A1)	$\varphi \to (\psi \to \varphi)$
(A2)	$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$
(A3)	$(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$
(Dual)	$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$
($\mathsf{K}_\Box$)	$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$
($\mathsf{K}_{[\,]}$)	$[p := \chi](\varphi \to \psi) \to ([p := \chi]\varphi \to [p := \chi]\psi)$
Recursion axioms:	
(R1)	$[p := \chi]q \leftrightarrow q$, given that p is not q
(R2)	$[p := \chi]p \leftrightarrow \chi$
(R3)	$[p := \chi]\neg\varphi \leftrightarrow \neg[p := \chi]\varphi$
(R4)	$[p := \chi](\varphi \wedge \psi) \leftrightarrow [p := \chi]\varphi \wedge [p := \chi]\psi$
(R5)	$[p := \chi]\Box\varphi \leftrightarrow \Box[p := \chi]\varphi$
Inference rules:	
(MP)	From φ and $\varphi \to \psi$, infer ψ
($\mathsf{Nec}_\Box$)	From φ, infer $\Box\varphi$
($\mathsf{Nec}_{[\,]}$)	From φ, infer $[p := \chi]\varphi$

Table 1. A proof system **MSL** for MSL

Then, let us note the following:

Proposition 2.5. *The following rule is derivable:*

$$\text{From } \chi_1 \leftrightarrow \chi_2, \text{ infer } [p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi.$$

Proof. It is by induction on φ.

(1) Formula φ is a propositional variable q. There are two different cases.

First, let us assume that q is distinct from p. By the axiom (R1), $[p := \chi_1]\varphi \leftrightarrow q$ and $[p := \chi_2]\varphi \leftrightarrow q$. Now it is easy to see that $[p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi$.

Next, assume that q is exactly p. By the axiom (R2), $[p := \chi_1]\varphi \leftrightarrow \chi_1$ and $[p := \chi_2]\varphi \leftrightarrow \chi_2$. By assumption, we have $\chi_1 \leftrightarrow \chi_2$, which then gives us $[p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi$, as needed.

(2) Formula φ is $\neg\psi$. By induction hypothesis, $[p := \chi_1]\psi \leftrightarrow [p := \chi_2]\psi$. So, $\neg[p := \chi_1]\psi \leftrightarrow \neg[p := \chi_2]\psi$. It follows from the axiom (R3) that $\neg[p := \chi_1]\psi \leftrightarrow [p := \chi_1]\neg\psi$ and $\neg[p := \chi_2]\psi \leftrightarrow [p := \chi_2]\neg\psi$. By propositional logic, $[p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi$.

(3) Formula φ is $\varphi_1 \wedge \varphi_2$. By induction hypothesis, for each $i \in \{1, 2\}$, we have $[p := \chi_1]\varphi_i \leftrightarrow [p := \chi_2]\varphi_i$. Using propositional logic, we can obtain $[p := \chi_1]\varphi_1 \wedge [p := \chi_1]\varphi_2 \leftrightarrow [p := \chi_2]\varphi_1 \wedge [p := \chi_2]\varphi_2$. It follows from (R4) that $[p := \chi_1]\varphi_1 \wedge [p := \chi_1]\varphi_2 \leftrightarrow [p := \chi_1]\varphi$ and that $[p := \chi_2]\varphi_1 \wedge [p := \chi_2]\varphi_2 \leftrightarrow [p := \chi_2]\varphi$. Consequently, $[p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi$, as desired.

(4) Formula φ is $\Box\psi$. By induction hypothesis, $[p := \chi_1]\psi \leftrightarrow [p := \chi_2]\psi$. Then, using propositional logic and the rule ($\mathsf{Nec}_\Box$), we can get $\Box([p := \chi_1]\psi \leftrightarrow [p := \chi_2]\psi)$. By ($\mathsf{K}_\Box$), (MP) and propositional logic, we have $\Box[p := \chi_1]\psi \leftrightarrow \Box[p := \chi_2]\psi$. It follows from axiom (R5) that $\Box[p := \chi_1]\psi \leftrightarrow [p := \chi_1]\Box\psi$ and that $\Box[p := \chi_2]\psi \leftrightarrow [p := \chi_2]\Box\psi$. So, $[p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi$.

(5) Formula φ is $[q := \psi_1]\psi_2$. Now, it follows from Proposition 2.4 that there is a $\theta \in \mathcal{L}_{\mathsf{ML}}$ with $[q := \psi_1]\psi_2 \leftrightarrow \theta$. For this θ, based on the arguments in (1)-(4), it holds that $[p := \chi_1]\theta \leftrightarrow [p := \chi_2]\theta$. We know from Proposition 2.3 that $[p := \chi_1]\theta \leftrightarrow [p := \chi_1]\varphi$ and that $[p := \chi_2]\theta \leftrightarrow [p := \chi_2]\varphi$. Then, $[p := \chi_1]\varphi \leftrightarrow [p := \chi_2]\varphi$, as needed. □

Remark 2.6. *As mentioned in Section 1, [21] extends* PAL *with substitution operators, and it proves that the two inference rules in Proposition 2.3 and Proposition 2.5 preserve* validity *of formulas. Our results can be seen as their syntactic counterparts.*

Now, it is easy to check the soundness of the proof system:

Proposition 2.7 (Soundness of **MSL**)**.** *The proof system* **MSL** *is sound for* MSL.

With the soundness, we can show the following:

Proposition 2.8. *For any $\varphi \in \mathcal{L}_{\mathsf{MSL}}$, there is a formula $\varphi_{\mathsf{ML}} \in \mathcal{L}_{\mathsf{ML}}$ such that for any $(\boldsymbol{M}, s)$, it holds that*

$$\boldsymbol{M}, s \vDash_{\mathsf{MSL}} \varphi \quad \textit{iff} \quad \boldsymbol{M}, s \vDash_{\mathsf{ML}} \varphi_{\mathsf{ML}}.$$

Proof. As indicated by Proposition 2.4, for any $\mathcal{L}_{\mathsf{MSL}}$, there is a standard modal formula φ_{ML} that is provably equivalent to the original φ. Now, by the soundness of the calculus **MSL**, formulas φ and φ_{ML} are equivalent, as required. □

Consequently, although MSL is a direct extension of ML, the two logics are as expressive as each other. With the result above, we can show the completeness of the proof system:

Theorem 2.9 (Completeness of **MSL**)**.** *The proof system* **MSL** *is complete for* MSL.

Proof. Consider an MSL-formula φ such that $\nvdash_{\mathbf{MSL}} \varphi$. With the help of the recursion axioms, we know that there is some $\varphi_{\mathsf{ML}} \in \mathcal{L}_{\mathsf{ML}}$ that is provably equivalent to φ. So, $\nvdash_{\mathbf{MSL}} \varphi_{\mathsf{ML}}$. Since **MSL** is a direct extension of **ML**, we have $\nvdash_{\mathbf{ML}} \varphi_{\mathsf{ML}}$. As **ML** is a complete calculus for ML, we have $\nvDash_{\mathsf{ML}} \varphi_{\mathsf{ML}}$. Now, using Proposition 2.8, it holds that $\nvDash_{\mathsf{MSL}} \varphi$. So, the calculus **MSL** is complete w.r.t. the class of models. □

Thus, **MSL** is a desired proof system for MSL. For more discussion on the axiomatization of DEL-style logics, we refer to [28].

The reasoning in the proof for Proposition 2.8 also illustrates that we can encode the satisfiability problem for MSL with that for ML, and since the latter is decidable [26], we have the following as a corollary:

Corollary 2.10 (Decidability of MSL)**.** *The satisfiability problem for* MSL *is decidable.*

So far, we have seen that MSL enjoys many desired properties. The techniques used to achieve them are inspired by those employed in DEL. Also, we showed that with respect to expressiveness, the logic is equivalent to ML, and so is equivalent to many landmark paradigms of DEL (e.g., the public announcement logic PAL). But it is worth noting that we do not necessarily need ML as a bridge to link them: as we will see, there is a natural and direct relation between our substitution operators and announcement operators of PAL (Definition 6.7). These suggest that our proposal could also be promising for modelling situations involving knowledge updates, but we leave this discussion for future.

3. Substitution operators v.s. ordinary replacements

In Section 1 we have clarified a conceptual difference between our notion of substitutions and the ordinary replacements $\varphi[\psi/p]$. On the other hand, it is also crucial to acknowledge the close relationship between these two approaches. An illustration of such connections is our recursion axioms in the proof system **MSL**: the replacements also have similar valid schemata [9]. To have a better understanding of the substitution operators, in this part we explore the precise relation between the two styles of substitutions. Along the way, we introduce many notions, and it is instructive to recognize the many applications of those concepts: for instance, they help in understanding the nature of the iterative setting MISL (cf. Sections 4.1 and 6.2).

Given a formula $\langle p := \psi\rangle\varphi$, let us first note that the pivot p would be 'bounded' by the operator, in that, very roughly speaking, the operator $\langle p := \psi\rangle$ would update the truth set of p with the states where ψ is true. To make this clear, we define the notions of *free variables* and *bound variables* in a formula:

Definition 3.1 (Free variables and bound variables in $\mathcal{L}_{\mathsf{MSL}}$)**.** *For any $\varphi \in \mathcal{L}_{\mathsf{MSL}}$, an occurrence of a propositional variable p is a* bound occurrence *if it is a pivot or appears in the scope of a substitution operator whose pivot is p. An occurrence of a propositional variable is a* free occurrence *if it is not bound. A propositional variable is called a* bound variable *if it has a bound occurrence. It is called a* free variable *if it has a free occurrence. We write $bv(\varphi)$ for* the set of bound variables *of φ, and $fv(\varphi)$ for* the set of free variables *of the formula.*

With the definition, a propositional variable in a formula can be both free and bound: for instance, in formula $\langle p := \Box p\rangle p$, the second occurrence of p is free, while the first and the third occurrences of the proposition are bound. Usually, it would be easier to consider the formulas in which no propositional letter is both free and bound:

Definition 3.2 (Clean $\mathcal{L}_{\mathsf{MSL}}$-formulas)**.** *A formula $\varphi \in \mathcal{L}_{\mathsf{MSL}}$ is* clean *if $fv(\varphi) \cap bv(\varphi) = \varnothing$.*

Fortunately, for many purposes it is enough to consider only the clean formulas, since we can turn an arbitrary formula to a clean one without changing its truth value. For instance, one can check that $\langle p := \Box p\rangle p$ is equivalent to $\langle q := \Box p\rangle q$. More generally, we have the following renaming rule for bound variables:

$$\langle p := \psi\rangle\varphi \leftrightarrow \langle q := \psi\rangle(\varphi[q/p]),\ \textit{where } q \textit{ is a fresh variable.} \qquad (\text{Renaming}_{\mathsf{MSL}})$$

It allows us to replace bound variables with fresh ones so that we can make a formula clean. Also, $\varphi[q/p]$ used in the rule is the formula obtained by replacing every *free* occurrence of the variable p in φ with an occurrence of q, which is formally given by the following:

Definition 3.3 (Replacements of variables in $\mathcal{L}_{\mathsf{MSL}}$)**.** *For any $\varphi, \psi \in \mathcal{L}_{\mathsf{MSL}}$, formula $\varphi[\psi/p]$ is given by the following:*

$$
\begin{aligned}
q[\psi/p] &:= \begin{cases} q & q \textit{ is distinct from } p \\ \psi & q \textit{ is } p \end{cases}\\
(\neg\varphi)[\psi/p] &:= \neg(\varphi[\psi/p])\\
(\varphi_1 \wedge \varphi_2)[\psi/p] &:= \varphi_1[\psi/p] \wedge \varphi_2[\psi/p]\\
(\Diamond\varphi)[\psi/p] &:= \Diamond(\varphi[\psi/p])\\
(\langle q := \chi\rangle\varphi)[\psi/p] &:= \begin{cases} \langle q := \chi[\psi/p]\rangle(\varphi[\psi/p]) & q \textit{ is distinct from } p \\ \langle q := \chi[\psi/p]\rangle\varphi & q \textit{ is } p \end{cases}
\end{aligned}
$$

We have the following:

Proposition 3.4. *The principle (Renaming$_{\mathsf{MSL}}$) is valid. As a consequence, every* MSL*-formula is equivalent to a clean formula.*

Proof. We show that (Renaming$_{\mathsf{MSL}}$) is valid. The second part of the proposition directly follows by recursively applying the renaming principle to a formula, as each application of the renaming principle strictly reduces the intersection size of free and bound variables.

Let $\langle p := \psi\rangle\varphi$ be a formula and $q \in \mathbf{P}$ be a propositional variable not occurring in $\langle p := \psi\rangle\varphi$. Let $\boldsymbol{F} = (W, R)$ be a frame. To achieve the goal, it is enough to show that for any valuations V_1, V_2 s.t. $V_1(p) = V_2(q)$ and for any $r \in \mathbf{P}$ distinct from p, q, $V_1(r) = V_2(r)$, it holds that

$$\boldsymbol{M}_1, w \vDash_{\mathsf{MSL}} \varphi \quad \text{iff} \quad \boldsymbol{M}_2, w \vDash_{\mathsf{MSL}} \varphi[q/p],$$

where $\boldsymbol{M}_1 = (W, R, V_1)$ and $\boldsymbol{M}_2 = (W, R, V_2)$.

We prove this by induction on φ. The cases for Boolean connectives $\neg$, $\wedge$ hold directly by induction hypothesis and Definition 3.3, and we merely consider other situations.

(1) Formula φ is a propositional variable r. There are two different cases.

(1.1) Formula r is p. Then, $\varphi[q/p]$ is q. By definition, $w \in V_1(p)$ iff $w \in V_2(q)$.

(1.2) Formula r is distinct from p. Then, $\varphi[q/p]$ is r. Again, by definition, $w \in V_1(r)$ iff $w \in V(r)$ iff $w \in V_2(r)$ (recall that q does not occur in $\langle p := \psi\rangle\varphi$, r must be different from q).

(2) Formula φ is $\Diamond\chi$. By Definition 3.3, $\varphi[q/p]$ is $\Diamond(\chi[q/p])$. By induction hypothesis, $[\![\chi]\!]^{\boldsymbol{M}_1} = [\![\chi[q/p]]\!]^{\boldsymbol{M}_2}$. With the truth condition for $\Diamond$, one can easily see that $\boldsymbol{M}_1, w \vDash_{\mathsf{MSL}} \varphi$ iff $\boldsymbol{M}_2, w \vDash_{\mathsf{MSL}} \varphi[q/p]$, as needed.

(3) We move to the case that φ is $\langle r := \theta\rangle\chi$. There are two different situations.

(3.1) Formula r is exactly p. Then, $\varphi[q/p]$ is $\langle p := \theta[q/p]\rangle\chi$. By induction hypothesis, $[\![\theta]\!]^{\boldsymbol{M}_1} = [\![\theta[q/p]]\!]^{\boldsymbol{M}_2}$. From this, it follows that $V_1|_{p:=\theta}(p) = V_2|_{p:=\theta[q/p]}(p)$. Now, $\boldsymbol{M}_1|_{p:=\theta}$ and $\boldsymbol{M}_2|_{p:=\theta[q/p]}$ may only differ on the truth set of q. Since q does not occur in χ, it holds that

$$\boldsymbol{M}_1|_{p:=\theta}, w \vDash_{\mathsf{MSL}} \chi \quad \text{iff} \quad \boldsymbol{M}_2|_{p:=\theta[q/p]}, w \vDash_{\mathsf{MSL}} \chi,$$

as desired.

(3.2) Formula r is different from p. Now, $\varphi[q/p]$ is $\langle r := \theta[q/p]\rangle(\chi[q/p])$. As the case above, by induction hypothesis, $V_1|_{r:=\theta}(r) = V_2|_{r:=\theta[q/p]}(r)$. For these two new valuations, $V_1|_{r:=\theta}(p) = V_1(p) = V_2(q) = V_2|_{r:=\theta[q/p]}(q)$ and for any other $u \in \mathbf{P}$, $V_1|_{r:=\theta}(u) = V_2|_{r:=\theta[q/p]}(u)$. So, we can still apply induction hypothesis to the setting involving $\boldsymbol{M}_1|_{r:=\theta}$ and $\boldsymbol{M}_2|_{r:=\theta[q/p]}$, which gives us the following:

$$\boldsymbol{M}_1|_{r:=\theta}, w \vDash_{\mathsf{MSL}} \chi \quad \text{iff} \quad \boldsymbol{M}_2|_{r:=\theta[q/p]}, w \vDash_{\mathsf{MSL}} \chi[q/p],$$

which completes the proof. □

Remark 3.5. *Generally the cleanness of formulas may* not *be preserved with respect to its subformulas. For instance, the formula $\langle p := q\rangle\langle p := p\rangle p$ is clean, but its subformula $\langle p := p\rangle p$ is not. But we can use (Renaming$_{\mathsf{MSL}}$) to obtain clean formulas such that their cleanness is preserved under their subformulas, e.g., by making the pivots of all substitution operators distinct from each other. With this understanding, in what follows, by clean* MSL*-formulas we actually mean that they are clean* MSL*-formulas such that the cleanness is preserved with respect to subformulas.*

With the help of these notions, we can establish the following principle connecting $\langle p := \psi\rangle\varphi$ and $\varphi[\psi/p]$:

Theorem 3.6. *$\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$ is a validity of* MSL*, given that $\langle p := \psi\rangle\varphi$ is clean.*

We will prove this in the remaining part of this section. Before introducing the details, let us note that when formula $\langle p := \psi\rangle\varphi$ is not clean, the equivalence may fail. For instance, given three different propositional letters p, q, r, consider formula $\langle p := q; q := r\rangle(p \wedge q)$ that is not clean: the formula is equivalent to $q \wedge r$, while formula $(\langle q := r\rangle(p \wedge q))[q/p]$, i.e., $\langle q := r\rangle(q \wedge q)$, is equivalent to r.

Remark 3.7. *[20] extends* PAL *with single-step substitution operators, and finds that $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$ is not valid. Although the result looks the same as our finding, we would like to emphasize that* the meaning of $\varphi[\psi/p]$ in [20] is very different from ours. *[20] does not distinguish between bound variables and free variables, and does not have the notion of clean formulas. In that work, $\varphi[\psi/p]$ denotes the formula obtained by replacing* all *occurrences of p in φ with ψ. As explained in [20], the failure of $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$ is caused by the interaction between substitution operators and public announcement operators, but one can see from the above counterexample that the equivalence does not hold even though public announcement operators are absent in our language. To make the equivalence hold, [20] imposes a precondition that the truth of ψ should be preserved under restriction to arbitrary submodels, and states that "a syntactic characterization of the rather semantic notion of 'preservation' is not available". But as Theorem 3.6 shows, we have a syntactic restriction to make the principle to hold, which is general enough (Proposition 3.4) and can also be transferred to the more complicated setting (Theorem 4.6).*

To prove Theorem 3.6, let us introduce the following order on clean formulas:

Definition 3.8 (Order on clean $\mathcal{L}_{\mathsf{MSL}}$-formulas)**.** *We define a binary relation $\ll$ on clean $\mathcal{L}_{\mathsf{MSL}}$-formulas as follows:*

$$\varphi \ll \neg\varphi, \quad \varphi \ll \varphi \wedge \psi, \quad \varphi \ll \psi \wedge \varphi, \quad \varphi \ll \Diamond\varphi, \quad \varphi[\psi/p] \ll \langle p := \psi\rangle\varphi.$$

We use $\gg$ for the converse of $\ll$.

This order is well-defined on clean formulas, since whenever the formula to the right-hand side of the order symbol $\ll$ is clean, then the formula on the left-hand side must be clean as well. Consequently, any descending $\ll$-chains spanned from a clean formula consist exclusively of clean formulas. In particular, replacements would not break cleanness, due to the following:

$$fv(\varphi[\psi/p]) = (fv(\varphi)\backslash\{p\}) \cup fv(\psi) = fv(\langle p := \psi\rangle\varphi)$$
$$bv(\varphi[\psi/p]) = bv(\varphi) \cup bv(\psi) \subseteq bv(\varphi) \cup bv(\psi) \cup \{p\} = bv(\langle p := \psi\rangle\varphi)$$

With respect to the order $\ll$, we have the following result:

Proposition 3.9. *Suppose $\langle p := \psi\rangle\varphi$ and $\langle p := \psi\rangle\chi$ are clean $\mathcal{L}_{\mathsf{MSL}}$-formulas, and φ is not atomic. Then, on the one hand, if $\varphi \gg \chi$ then $\varphi[\psi/p] \gg \chi[\psi/p]$. On the other hand, if $\varphi[\psi/p] \gg \chi$, then there is a formula χ' such that $\chi'[\psi/p]$ is the same formula as χ, and $\varphi \gg \chi'$.*

Proof. Since $\langle p := \psi\rangle\varphi$ and $\langle p := \psi\rangle\chi$ are clean, all formulas φ, ψ, χ, $\varphi[\psi/p]$ and $\chi[\psi/p]$ are clean. Now, assume that $\varphi \gg \chi$. Let us spit into cases according to the main connective of φ. When the main connective of φ is $\neg$, $\wedge$, or $\Diamond$, we prove the statement while setting χ' to be χ.

(1) The main connective of φ is $\neg$. Then the main connective of $\varphi[\psi/p]$ is also $\neg$. Now, $\varphi \gg \chi$ illustrates that the formula φ is $\neg\chi$. So, $\varphi[\psi/p]$ is the same as $\neg(\chi[\psi/p])$, which then gives us $\varphi[\psi/p] \gg \chi[\psi/p]$.

(2) The main connective of φ is $\wedge$. Then the main connective of $\varphi[\psi/p]$ is also $\wedge$. By $\varphi \gg \chi$, formula χ is either of the conjunctions of φ, which indicates that $\chi[\psi/p]$ is either of the conjunctions of $\varphi[\psi/p]$. Thus, $\varphi[\psi/p] \gg \chi[\psi/p]$.

(3) The main connective of φ is $\Diamond$. Then the main connective of $\varphi[\psi/p]$ is also $\Diamond$. It follows from $\varphi \gg \chi$ that formula φ is $\Diamond\chi$. So, $\varphi[\psi/p]$ is $\Diamond(\chi[\psi/p])$, which demonstrates that $\varphi[\psi/p] \gg \chi[\psi/p]$.

(4) The main connective of φ is a substitution operator. Suppose φ is $\langle q := \psi'\rangle\varphi'$. We set $\chi' = \varphi'[\psi'/q]$, and it follows from the definition of $\gg$ that χ' is the only only $\gg$-successor of φ. Next we show that $\chi'[\psi/p]$ is the only successor of $\varphi[\psi/p]$. To this end, we consider two different situations.

(4.1) Proposition q is different from p. Then, $\varphi[\psi/p]$ is the formula $\langle q := \psi'[\psi/p]\rangle(\varphi'[\psi/p])$. Note that $\chi'[\psi/p]$ is obtained by first replacing the free occurrences of q in φ' with ψ' and then replacing the free occurrences of p in $\varphi'[\psi'/q]$ with ψ. The free occurrences of p in $\varphi'[\psi'/q]$ are those in φ' and ψ'. The replacement process of $\chi'[\psi/p]$ is equivalent to first replacing the free occurrences of p in φ' with ψ, and then replacing the free occurrences of q in $\varphi'[\psi/p]$ with $\psi'[\psi/p]$. Since $q \notin fv(\psi)$, the free occurrences of q in φ' are the same as the free occurrences of q in $\varphi'[\psi/p]$. Hence, $\chi'[\psi/p]$ is the same formula as $(\varphi'[\psi/p])[(\psi'[\psi/p])/q]$. It follows that $\chi'[\psi/p]$ is the only successor of the formula $\langle q := \psi'[\psi/p]\rangle(\varphi'[\psi/p])$, and equivalently, $\chi'[\psi/p]$ is the only successor of the formula $\varphi[\psi/p]$.

(4.2) Proposition q is p. Then, $\varphi[\psi/p]$ is $\langle p := \psi'[\psi/p]\rangle\varphi'$, and $\chi'[\psi/p]$ is $(\varphi'[\psi'/p])[\psi/p]$. Following a similar argument as in (4.1), we can see that $\chi'[\psi/p]$ is the same formula as $\varphi'[(\psi'[\psi/p])/p]$, and therefore $\chi'[\psi/p]$ is the only successor of the formula $\varphi[\psi/p]$.

Therefore, it is always the case that $\chi'[\psi/p]$ is the only successor of $\varphi[\psi/p]$. On the one hand, if $\varphi \gg \chi$ then χ is χ', and hence $\varphi[\psi/p] \gg \chi[\psi/p]$. On the other hand, if $\varphi[\psi/p] \gg \chi[\psi/p]$, then $\chi[\psi/p]$ is the same formula as $\chi'[\psi/p]$, where $\varphi \gg \chi'$ holds. □

The above result implies that, on the one hand, given a finite chain

$$\varphi \gg \varphi_1 \gg \varphi_2 \gg \cdots \gg q_\varphi,$$

the following chain exists:

$$\varphi[\psi/p] \gg \varphi_1[\psi/p] \gg \varphi_2[\psi/p] \gg \cdots \gg q_\varphi[\psi/p].$$

On the other hand, any chain spanned by $\varphi[\psi/p]$ should start with

$$\varphi[\psi/p] \gg \varphi_1[\psi/p] \gg \varphi_2[\psi/p] \gg \cdots \gg q_\varphi[\psi/p],$$

where

$$\varphi \gg \varphi_1 \gg \varphi_2 \gg \cdots \gg q_\varphi$$

is a finite chain. If q_φ is not p, then the end of the above chain is exactly q_φ, which cannot have any $\gg$-successor. But if q_φ is p, then the end of the above chain is ψ. When the latter is the case, given a finite $\gg$-chain ε staring from ψ and ending at a propositional variable r, by adding ε to the chain above, we can obtain a finite $\gg$-chain starting from $\varphi[\psi/p]$ and ending at r, which cannot have further $\gg$-successor. With this, we can establish the following:

Lemma 3.10. *The order $\ll$ defined in Definition 3.8 is well-founded.*

Proof. We need to prove that any clean $\mathcal{L}_{\mathsf{MSL}}$-formula spans finite descending $\ll$-chains. It goes by induction on formulas. The base cases for propositional variables p are straightforward. The first four clauses in Definition 3.8 only allow a formula to descend to a strictly shorter formula. We now proceed to consider for $\langle p := \psi\rangle\varphi$.

Let $\langle p := \psi\rangle\varphi$ be a clean $\mathcal{L}_{\mathsf{MSL}}$-formula. Then, both ψ and φ are clean (Remark 3.5). By induction hypothesis, φ and ψ only span finite descending $\ll$-chains. By Proposition 3.9, any $\ll$-chain $\varphi[\psi/p]$ spans has an initial segment of the following form:

$$\varphi[\psi/p] \gg \varphi_1[\psi/p] \gg \varphi_2[\psi/p] \gg \cdots \gg q[\psi/p],$$

where

$$\varphi \gg \varphi_1 \gg \varphi_2 \gg \cdots \gg q$$

is a finite descending chain starting from φ with where $q \in \mathbf{P}$. Then, by the previous analysis, we know that $\varphi[\psi/p]$ can only span finite $\gg$-chains that end at propositional variables and so have no further $\gg$-successor. The proof is completed. □

We then define *the depth of a clean formula* $\mathsf{dep}(\varphi)$ to be the length of the longest descending $\ll$-chains spanned by φ. The details are as follows:

Definition 3.11 (Depth of a clean $\mathcal{L}_{\mathsf{MSL}}$-formula)**.** *For a clean $\varphi \in \mathcal{L}_{\mathsf{MSL}}$, its depth $\mathsf{dep}(\varphi)$ is given by the following:*

$$\mathsf{dep}(p) := 0 \quad \mathsf{dep}(\neg\varphi) := \mathsf{dep}(\varphi) + 1 \quad \mathsf{dep}(\varphi_1 \wedge \varphi_2) := \max\{\mathsf{dep}(\varphi_1), \mathsf{dep}(\varphi_2)\} + 1$$
$$\mathsf{dep}(\Diamond\varphi) := \mathsf{dep}(\varphi) + 1 \quad \mathsf{dep}(\langle p := \psi\rangle\varphi) := \mathsf{dep}(\varphi[\psi/p]) + 1$$

From the proof for Lemma 3.10, we can see that $\mathsf{dep}(\langle p := \psi\rangle\varphi)$ is upper-bounded by $\mathsf{dep}(\psi) + \mathsf{dep}(\varphi) + 1$. For any clean $\varphi \in \mathcal{L}_{\mathsf{MSL}}$, its depth is always finite. Now we are ready to prove Theorem 3.6.

Proof of Theorem 3.6. It goes by induction on $\mathsf{dep}(\langle p := \psi\rangle\varphi)$. The base case is that $\mathsf{dep}(\langle p := \psi\rangle\varphi) = 1$, which means that $\varphi[\psi/p]$ is a propositional variable. Clearly, $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$. Now, suppose the statement is true for all $\langle p := \psi\rangle\varphi$ such that $\mathsf{dep}(\langle p := \psi\rangle\varphi) < n$, where $n > 1$. We are going to show that it is also true for $\langle p := \psi\rangle\varphi$ such that $\mathsf{dep}(\langle p := \psi\rangle\varphi) = n$. To do so, let us consider the form of φ.

(1) Formula φ is a propositional variable. When φ is p, $\langle p := \psi\rangle\varphi$ is equivalent to ψ (by the axiom (R1)), and $\varphi[\psi/p]$ is exactly ψ. When φ is different from p, $\langle p := \psi\rangle\varphi$ is equivalent to φ (by the axiom (R2)), and $\varphi[\psi/p]$ is exactly φ. Therefore $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$.

(2) Formula φ is $\neg\chi$. By the axiom (R3), $\langle p := \psi\rangle\neg\chi \leftrightarrow \neg\langle p := \psi\rangle\chi$. Then,

$$\mathsf{dep}(\langle p := \psi\rangle\chi) = \mathsf{dep}(\chi[\psi/p]) + 1 = \mathsf{dep}(\neg\chi[\psi/p]) = \mathsf{dep}(\langle p := \psi\rangle\varphi) - 1$$

By induction hypothesis, it holds that $\langle p := \psi\rangle\chi \leftrightarrow \chi[\psi/p]$. Now, we have $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$.

(3) Formula φ is $\chi_1 \wedge \chi_2$. By the axiom (R4), $\langle p := \psi\rangle(\chi_1 \wedge \chi_2) \leftrightarrow \langle p := \psi\rangle\chi_1 \wedge \langle p := \psi\rangle\chi_2$. Also, for each $i \in \{1, 2\}$, it holds that

$$\mathsf{dep}(\langle p := \psi\rangle\chi_i) = \mathsf{dep}(\chi_i[\psi/p]) + 1 < \mathsf{dep}(\varphi[\psi/p]) + 1 = \mathsf{dep}(\langle p := \psi\rangle\varphi)$$

Then, by induction hypothesis, it holds immediately that $\langle p := \psi\rangle\chi_i \leftrightarrow \chi_i[\psi/p]$. Also, since $\chi_1[\psi/p] \wedge \chi_2[\psi/p]$ is exactly $\varphi[\psi/p]$, we can obtain $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$.

(4) Formula φ is $\Diamond\chi$. It follows from (R5) that $\langle p := \psi\rangle\Diamond\chi \leftrightarrow \Diamond\langle p := \psi\rangle\chi$. Moreover,

$$\mathsf{dep}(\langle p := \psi\rangle\chi) = \mathsf{dep}(\Diamond\chi[\psi/p]) = \mathsf{dep}(\langle p := \psi\rangle\varphi) - 1$$

By induction hypothesis, it holds that $\langle p := \psi\rangle\chi \leftrightarrow \chi[\psi/p]$. Now, we have $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$.

(5) Formula φ is $\langle q := \chi_1\rangle\chi_2$. Since $\langle p := \psi\rangle\varphi$ is clean, we can assume that φ is also clean (Remark 3.5). Now, it is enough to prove the following equivalences:

$$\begin{aligned}\langle p := \psi\rangle\langle q := \chi_1\rangle\chi_2 &\leftrightarrow \langle p := \psi\rangle(\chi_2[\chi_1/q])\\ &\leftrightarrow (\chi_2[\chi_1/q])[\psi/p]\\ &\leftrightarrow (\langle q := \chi_1\rangle\chi_2)[\psi/p]\end{aligned}$$

One can verify that $\mathsf{dep}(\langle q := \chi_1\rangle\chi_2) < \mathsf{dep}(\langle p := \psi\rangle\langle q := \chi_1\rangle\chi_2) = n$. Therefore, $\langle q := \chi_1\rangle\chi_2 \leftrightarrow \chi_2[\chi_1/q]$. Hence, the first and the last equivalences hold. To prove the second equivalence, it suffices to show that

$$\mathsf{dep}(\langle p := \psi\rangle(\chi_2[\chi_1/q])) < n.$$

By Definition 3.8, it holds that $\varphi \gg \chi_2[\chi_1/q]$. Then, it follows from Proposition 3.9 that $\varphi[\psi/p] \gg (\chi_2[\chi_1/q])[\psi/p]$. Hence, we have the following:

$$n = \mathsf{dep}(\langle p := \psi\rangle\varphi) = \mathsf{dep}(\varphi[\psi/p]) + 1 = \mathsf{dep}((\chi_2[\chi_1/q])[\psi/p]) + 2 = \mathsf{dep}(\langle p := \psi\rangle(\chi_2[\chi_1/q])) + 1$$

So, $\mathsf{dep}(\langle p := \psi\rangle(\chi_2[\chi_1/q])) < n$, and the second equivalence holds. The proof is completed. □

Now we have shown the main result of this section, which clarifies the relationship between $\langle p := \psi\rangle\varphi$ and $\varphi[\psi/p]$. It is important to emphasize that the said comparison has the language $\mathcal{L}_{\mathsf{MSL}}$ as a common basis, in that the replacement operation is defined for $\mathcal{L}_{\mathsf{MSL}}$ (Definition 3.3). It would also be meaningful to compare $\mathcal{L}_{\mathsf{MSL}}$ with the language extending $\mathcal{L}_{\mathsf{ML}}$ with the replacements (but not containing substitution operators) [9].

To end this section, let us note that Theorem 3.6 holds broader significance beyond its immediate context. For instance, it offers an alternative approach to prove Proposition 2.8 by induction on the depth of formulas, which we will leave as an exercise to the reader. Also, many of the notions that were used to prove the result are crucial, and we will revisit and generalize them in later sections. We are now ready to move to the more powerful setting containing the iterative generalizations of the single-step substitutions.

4. A modal logic for iterative substitutions

This part explores the augmented *Modal Iterative Substitution Logic* (MISL) with iterative substitution operators. The language of the logic is proposed in [5], and we will proceed to explore its formal properties. As we will see, the iterative computations make the logic much more complicated than MSL. This section will present various validities to show how the iterative operators work and formally show that the backward induction introduced in Scenario 2 of Section 1 can be characterized by MISL. In later sections, we will look into the properties of MISL, involving expressiveness and computational behavior, and study its connections with some relevant frameworks.

The *language* $\mathcal{L}_{\mathsf{MISL}}$ *for* MISL extends $\mathcal{L}_{\mathsf{MSL}}$ with iterative substitution operators

$$\langle (p := \psi)^*\rangle\varphi$$

expressing that *there is some* $n \in \mathbb{N}$ *such that after we iteratively substitute the truth set of p with that of* ψ *for n times,* φ *is the case*. Precisely, the truth condition is given by the following:

$$\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle (p := \psi)^*\rangle\varphi \;\Leftrightarrow\; \boldsymbol{M}|_{(p:=\psi)^n}, s \vDash_{\mathsf{MISL}} \varphi \textit{ for some } n \in \mathbb{N},$$

where $\boldsymbol{M}|_{(p:=\psi)^n} = (W, R, V|_{(p:=\psi)^n})$ disagrees with $\boldsymbol{M}$ only on the valuation of p, and $V|_{(p:=\psi)^n}(p)$ is defined in the following inductive manner:

$$V|_{(p:=\psi)^0}(p) := V(p)$$
$$V|_{(p:=\psi)^{n+1}}(p) := \{s \in W : \boldsymbol{M}|_{(p:=\psi)^n}, s \vDash_{\mathsf{MISL}} \psi\}$$

So, for any natural number $n \in \mathbb{N}$, $\boldsymbol{M}|_{(p:=\psi)^n}$ is obtained by n model transformations starting from $\boldsymbol{M}$.

We use $[(p := \psi)^*]$ as the dual of $\langle(p := \psi)^*\rangle$, and so its truth condition is as follows:

$$\boldsymbol{M}, s \vDash_{\mathsf{MISL}} [(p := \psi)^*]\varphi \;\Leftrightarrow\; \boldsymbol{M}|_{(p:=\psi)^n}, s \vDash_{\mathsf{MISL}} \varphi \textit{ for all } n \in \mathbb{N}.$$

Unlike the case of $\langle p := \psi\rangle\varphi$, iterative substitution operators are not self-dual, i.e., $[(p := \psi)^*]\varphi \leftrightarrow \langle(p := \psi)^*\rangle\varphi$ fails in general.

To close the part, let us introduce some syntactic notions. To distinguish from iterative substitution operators, those $\langle p := \psi\rangle$ and $[p := \psi]$ are termed as *simple substitution operators*. Moreover, both the iterative and the simple ones are called *substitution operators*. The notion of *subformulas for* MISL extends that for MSL with the following:

$$\mathsf{Sub}(\langle(p := \psi)^*\rangle\varphi) = \mathsf{Sub}(\psi) \cup \mathsf{Sub}(\varphi) \cup \{\langle(p := \psi)^*\rangle\varphi\}.$$

For a formula $\langle(p := \psi)^*\rangle\varphi$, p is the *pivot* of the iterative substitution operator, but in contrast to the case of simple substitution operators, we define the *scope* of operator $\langle(p := \psi)^*\rangle$ as the whole formula $\langle(p := \psi)^*\rangle\varphi$. Finally, let us introduce the following notion of 'depth of iteration' $\mathsf{DOI}(\varphi)$:

$$\mathsf{DOI}(p) = 0, \quad \mathsf{DOI}(\neg\varphi) = \mathsf{DOI}(\varphi), \quad \mathsf{DOI}(\varphi \wedge \psi) = \max\{\mathsf{DOI}(\varphi), \mathsf{DOI}(\psi)\}, \quad \mathsf{DOI}(\Diamond\varphi) = \mathsf{DOI}(\varphi),$$
$$\mathsf{DOI}(\langle p := \psi\rangle\varphi) = \max\{\mathsf{DOI}(\varphi), \mathsf{DOI}(\psi)\}, \quad \mathsf{DOI}(\langle(p := \psi)^*\rangle\varphi) = \max\{\mathsf{DOI}(\varphi), \mathsf{DOI}(\psi) + 1\},$$

This corresponds to the number of layers of iterative operators involved in a formula. In the next part, we explore more validities of the logic.

4.1. Validities of MISL

Since MISL is a direct extension of MSL, all validities of the latter still hold in the new setting, including the recursion axioms for the simple substitution operators. Also, as we shall see, the generalizations of the bound variable renaming rule ($\text{Renaming}_{\mathsf{MSL}}$) and Theorem 3.6 for MISL are also true. Moreover, the following schemata are valid:

Proposition 4.1. *The following formulas involving iterative substitution operators are valid:*

$$\langle(p := \psi)^*\rangle q \leftrightarrow q, \quad \textit{if } q \textit{ is not } p.$$
$$\langle(p := \psi)^*\rangle p \leftrightarrow p \vee \langle p := \psi; (p := \psi)^*\rangle p$$
$$\langle(p := \psi)^*\rangle\neg\varphi \leftrightarrow \neg[(p := \psi)^*]\varphi$$
$$\langle(p := \psi)^*\rangle\Diamond\varphi \leftrightarrow \Diamond\langle(p := \psi)^*\rangle\varphi$$
$$\langle(p := \psi)^*\rangle(\varphi_1 \wedge \varphi_2) \to \langle(p := \psi)^*\rangle\varphi_1 \wedge \langle(p := \psi)^*\rangle\varphi_2$$

Note that the last formula is only one-directional. Some patterns above are similar to that of the recursion axioms in the proof system **MSL**, whereas the others have typical forms for iterative operations like those in propositional dynamic logic PDL [29]. Let us now take a closer look at the iterative substitution operators.

As stated, an iterative substitution operator $\langle(p := \psi)^*\rangle$ leads to a sequence of transformations of the truth set of p, starting from the original $V(p)$ given by a model. Generally speaking, the proposition p in $\langle(p := \psi)^*\rangle$ functions as both the starting point of the transformation sequence and the pivot of each single-step transformation $\langle p := \psi\rangle$. Intuitively, when ψ contains p, the occurrences of p in ψ in the first $\langle p := \psi\rangle$ of the sequence are free, while those in the later stages are bound. To make this clear, let us first introduce the following:

Definition 4.2 (Normal $\mathcal{L}_{\mathsf{MISL}}$-formulas)**.** *An* MISL*-formula is* normal *if every iterative substitution* $(p := \psi)^*$ *follows immediately after a single-step initialization* $p := \psi_0$*.*

For instance, $\langle p := \bot; (p := \Box p)^*\rangle p$ is a normal formula, while $\langle p := \bot; q := \top; (p := \Box p)^*\rangle p$ is not. Given a normal formula, its subformulas might not be normal: for instance, formula $\langle p := \neg p; (p := \Box p)^*\rangle p$ is normal, but $\langle(p := \Box p)^*\rangle p$ is not. In a normal formula, each iterative substitution is a part of $p := \psi_0; (p := \psi)^*$. Now, we can generalize the notions of *free occurrences*, *free variables*, *bound occurrences* and *bound variables* for MSL (Definition 3.1) to *normal formulas*. For instance, in formula $\langle p := p \vee q; (p := \Box p)^*\rangle p$, only the second occurrence of p is free, while its other occurrences are bound. Moreover, an MISL-formula is *clean* if it is a normal formula not containing any propositional letter that is both free and bound.

Just as the case in MSL, we can turn a formula of $\mathcal{L}_{\mathsf{MISL}}$ to a clean one. For instance, $\langle (p := \Box p)^* \rangle p$ is equivalent to $\langle (q := p); (q := \Box q)^* \rangle q$. Given a formula of $\mathcal{L}_{\mathsf{MISL}}$, to obtain a desired clean formula, we can first apply the following normalization rule

$$\langle (p := \psi)^* \rangle \varphi \leftrightarrow \langle p := p; (p := \psi)^* \rangle \varphi \qquad \text{(Normalization)}$$

that transforms any MISL-formula to a normal one, and then apply the two bound variable renaming rules below to transform every normal MISL-formula to a clean formula:

$$\begin{array}{c} \langle p := \psi \rangle \varphi \leftrightarrow \langle q := \psi \rangle \varphi[q/p], \\ \langle p := \psi_0; (p := \psi_1)^* \rangle \varphi \leftrightarrow \langle q := \psi_0; (q := \psi_1[q/p])^* \rangle \varphi[q/p] \end{array} \qquad (\text{Renaming}_{\mathsf{MISL}})$$

where the main connective of φ in $\langle p := \psi \rangle \varphi$ in the first equivalence is not an iterative substitution operator,[7] the propositional variable q is a fresh variable in both the principles, and formula $\varphi[q/p]$ is obtained by replacing every free occurrence of the variable p in φ with an occurrence of q, which is defined in the following manner:

Definition 4.3 (Replacement of variables in a normal formula)**.** *For any normal formulas $\varphi, \psi \in \mathcal{L}_{\mathsf{MISL}}$, $\varphi[\psi/p]$ is defined by extending the clauses for propositional variables, $\neg, \wedge, \Diamond$ in Definition 3.3 with the following:*

$$(\langle q := \chi \rangle \varphi)[\psi/p] := \begin{cases} \langle q := \chi[\psi/p] \rangle (\varphi[\psi/p]) & q \text{ is different from } p \\ \langle q := \chi[\psi/p] \rangle \varphi & q \text{ is } p \end{cases}$$

$$(\langle q := \chi_0; (q := \chi_1)^* \rangle \varphi)[\psi/p] := \begin{cases} \langle q := \chi_0[\psi/p]; (q := \chi_1[\psi/p])^* \rangle (\varphi[\psi/p]) & q \text{ is different from } p \\ \langle q := \chi_0[\psi/p]; (q := \chi_1)^* \rangle \varphi & q \text{ is } p \end{cases}$$

where the main connective of φ in the first clause is not an iterative substitution operator.

With the notions, we now introduce the following:

Proposition 4.4. *The two principles of (Renaming$_{\mathsf{MISL}}$) are valid. As a direct consequence, every normal formula in* MISL *is equivalent to a clean formula.*

Proof. We show that the two principles of (Renaming$_{\mathsf{MISL}}$) are valid. The proof for the validity of the first principle, $\langle p := \psi \rangle \varphi \leftrightarrow \langle q := \psi \rangle \varphi[q/p]$, is by induction on φ, and the argument is the same as that in the proof for Proposition 3.4. We now move to showing the validity of the second principle.

Let $\langle p := \psi_0; (p := \psi_1)^* \rangle \psi_2$ be a formula and $q \in \mathbf{P}$ be a propositional variable not occurring in it. Let $\boldsymbol{F} = (W, R)$ be a frame. We are going to first claim and prove the following statement:

Claim: *Let V_1, V_2 be two valuations such that $V_1(p) = V_2(q)$ and $V_1(r) = V_2(r)$ for any $r \in \mathbf{P}$ distinct from p and q. Let $\boldsymbol{M}_1 = (W, R, V_1)$, $\boldsymbol{M}_2 = (W, R, V_2)$ be two arbitrary models with the same frame and valuations V_1, V_2 respectively. Then for any normal formula $\varphi \in \mathcal{L}_{\mathsf{MISL}}$,*

$$\boldsymbol{M}_1, w \vDash_{\mathsf{MISL}} \varphi \textit{ iff } \boldsymbol{M}_2, w \vDash_{\mathsf{MISL}} \varphi[q/p].$$

We prove this statement by an outer induction on $\mathsf{DOI}(\varphi)$ and an inner induction on the length of φ, where the length of a formula means the number of symbols occurring in it.

(1) We begin with the basic case that $\mathsf{DOI}(\varphi) = 0$, i.e., φ does not contain any iterative substitution operator. Again, this can be proved with the same argument as that in the proof for Proposition 3.4.

[7] Precisely, since $\langle p := \psi \rangle \varphi$ is a normal formula, the main connective of φ is either $\neg, \wedge, \Diamond$, a simple substitution operator, or an iterative substitution operator with pivot p. But the last case is dealt with by the second renaming rule, and this is why we require that the main connective of φ in $\langle p := \psi \rangle \varphi$ is *not* an iterative operator.

(2) For the induction step, let $1 \leqslant m \in \mathbb{N}$ and suppose that the claim is true for (i) every formula ψ with $\mathsf{DOI}(\psi) < m$ and (ii) every formula χ such that $\mathsf{DOI}(\chi) = m$ and the length is smaller than k. We proceed to consider the case that $\mathsf{DOI}(\varphi) = m$ and the length of φ is k. For this, we consider the form of φ.

Since $m > 1$, φ is not a propositional variable. The cases for Boolean connectives $\neg$, $\wedge$ and modality $\Diamond$ hold directly by induction hypothesis. Now consider the case where φ is $\langle r := \psi\rangle\chi$. The reasoning for the case where the main connective of χ is not an iterative substitution operator, is similar to that in the proof for Proposition 3.4. We need only consider the case where the main connective of χ is an iterative substitution operator, and since φ is a normal formula, φ must be $\langle r := \psi_0; (r := \psi_1)^*\rangle\chi$. We then split into two different cases according to whether r is p.

(2.1) Propositional variable r is p. By definition, $\varphi[q/p]$ is $\langle p := \psi_0[q/p]; (p := \psi_1)^*\rangle\chi$. Note that either (i) $\mathsf{DOI}(\psi_0) = m$ and the length of ψ_0 is smaller than k or (ii) $\mathsf{DOI}(\psi_0) < m$. So, by induction hypothesis, it holds that

$$\boldsymbol{M}_1, w \vDash_{\mathsf{MISL}} \psi_0 \quad \text{iff} \quad \boldsymbol{M}_2, w \vDash_{\mathsf{MISL}} \psi_0[q/p],$$

which indicates $[\![\psi_0]\!]^{\boldsymbol{M}_1} = [\![\psi_0[q/p]]\!]^{\boldsymbol{M}_2}$. Now, we just need to show that

$$\boldsymbol{M}_1|_{p:=\psi_0} \vDash_{\mathsf{MISL}} \langle(p := \psi_1)^*\rangle\chi \quad \text{iff} \quad \boldsymbol{M}_2|_{p:=\psi_0[q/p]} \vDash_{\mathsf{MISL}} \langle(p := \psi_1)^*\rangle\chi.$$

Since $\boldsymbol{M}_1|_{p:=\psi_0}$ and $\boldsymbol{M}_2|_{p:=\psi_0[q/p]}$ may only differ on the truth set of q that does not occur in $\langle(p := \psi_1)^*\rangle\chi$, the equivalence above holds.

(2.2) Propositional letter r is different from p. By definition, $\varphi[q/p]$ is $\langle r := \psi_0[q/p]; (r := \psi_1[q/p])^*\rangle(\chi[q/p])$. As the case above, we know that $[\![\psi_0]\!]^{\boldsymbol{M}_1} = [\![\psi_0[q/p]]\!]^{\boldsymbol{M}_2}$. Now, it holds that

$$\boldsymbol{M}_1|_{r:=\psi_0} \vDash_{\mathsf{MISL}} \langle(r := \psi_1)^*\rangle\chi \quad \text{iff} \quad \boldsymbol{M}_1|_{r:=\psi_0} \vDash_{\mathsf{MISL}} \langle(r := \psi_1)^n\rangle\chi \text{ for some } n \in \mathbb{N}.$$

When n is 0, by induction hypothesis it holds immediately that $\boldsymbol{M}_1|_{r:=\psi_0} \vDash_{\mathsf{MISL}} \chi$ iff $\boldsymbol{M}_2|_{r:=\psi_0[q/p]} \vDash_{\mathsf{MISL}} \chi[q/p]$. We now move to the case that $n > 0$. Since $\mathsf{DOI}(\psi_1) < m$, for any $0 \leqslant j \leqslant n-1$, $\boldsymbol{M}_1|_{r:=\psi_0;(r:=\psi_1)^j}, w \vDash_{\mathsf{MISL}} \psi_1$ iff $\boldsymbol{M}_2|_{r:=\psi_0[q/p];(r:=\psi_1[q/p])^j}, w \vDash_{\mathsf{MISL}} \psi_1[q/p]$. This indicates that $V_1|_{r:=\psi_0;(r:=\psi_1)^n}(r) = V_2|_{r:=\psi_0[q/p];(r:=\psi_1[q/p])^n}(r)$. By induction hypothesis, it holds that

$$\boldsymbol{M}_1|_{r:=\psi_0;(r:=\psi_1)^n}, w \vDash_{\mathsf{MISL}} \chi \quad \text{iff} \quad \boldsymbol{M}_2|_{r:=\psi_0[q/p];(r:=\psi_1[q/p])^n}, w \vDash_{\mathsf{MISL}} \chi[q/p].$$

This completes the proof of the claim.

Now given the claim, one can prove the following statement:

$$\forall n \in \mathbb{N}, \langle p := \psi_0; (p := \psi_1)^n\rangle\varphi \leftrightarrow \langle q := \psi_0; (q := \psi_1[q/p])^n\rangle\varphi[q/p],$$

It is enough to show that for all $n \in \mathbb{N}$, $V_1^{(n)} := V|_{p:=\psi_0;(p:=\psi_1)^n}$ and $V_2^{(n)} := V|_{q:=\psi_0;(q:=\psi_1[q/p])^n}$ satisfy the condition of the claim, then the statement follows since φ is a normal formula. To show that $V_1^{(n)}$ and $V_2^{(n)}$ agree on $r \notin \{p, q\}$ and $V_1^{(n)}(p) = V_2^{(n)}(q)$, one can prove by induction on n - note that the base case follows directly, and each induction step can be proved by combining the claim with the induction hypothesis. Summing up, the above implies that $\langle p := \psi_0; (p := \psi_1)^*\rangle\varphi \leftrightarrow \langle q := \psi_0; (q := \psi_1[q/p])^*\rangle\psi_2[q/p]$, which completes the proof. □

Remark 4.5. *Similar to the case in Remark 3.5, when saying that an* MISL*-formula φ is clean, we also assume that:*

- *When φ is $\langle p := \psi\rangle\chi$ and the main connective of χ is not an iterative substitution operator, formulas χ and ψ are also clean.*
- *When φ is $\langle p := \psi_1; (p := \psi_2)^*\rangle\chi$, all formulas ψ_1, ψ_2 and χ are clean.*

This can also be achieved by repeated applications of (Renaming$_{\mathsf{MISL}}$).

Now, we can show that under suitable condition, the principle $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$ stated in Theorem 3.6 also holds in the enlarged framework MISL:

Theorem 4.6. *$\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$ is a validity, if both $\langle p := \psi\rangle\varphi$ and φ are clean formulas.*

We will prove it in the remainder of this part, following a similar method to the proof of Theorem 3.6 in Section 3. Let us first define the following order on clean MISL-formulas, which is an extension of the one given by Definition 3.8:

Definition 4.7 (Order on clean formulas of MISL)**.** *We define a binary relation $\ll$ on clean $\mathcal{L}_{\mathsf{MISL}}$-formulas:*

$$\varphi \ll \neg\varphi, \quad \varphi \ll \varphi \wedge \psi, \quad \varphi \ll \psi \wedge \varphi, \quad \varphi \ll \Diamond\varphi,$$
$$\varphi[\psi/p] \ll \langle p := \psi\rangle\varphi, \quad \langle p := \psi_0; (p := \psi_1)^n\rangle\varphi \ll \langle p := \psi_0; (p := \psi_1)^*\rangle\varphi,$$

where $n \in \mathbb{N}$, the main connective of the φ in the clause for $\varphi[\psi/p]$ is not an iterative substitution operator, and the replacement $\varphi[\psi/p]$ is defined in Definition 4.3. Again, we use $\gg$ for the converse of $\ll$.

With the condition used in the clause for $\langle p := \psi\rangle\varphi$, one can check that whenever the formula to the right-hand-side of $\ll$ is clean, the formula to the left-hand-side of $\ll$ is also clean. Now, we can prove the following result that is similar to Proposition 3.9:

Proposition 4.8. *Let $\langle p := \psi; (p := \psi)^*\rangle\varphi$ and $\langle p := \psi; (p := \psi)^*\rangle\chi$ be clean, and suppose φ is not atomic. If $\varphi \gg \chi$, then $\varphi[\psi/p] \gg \chi[\psi/p]$. On the other hand, if $\varphi[\psi/p] \gg \chi$, then there is a formula χ' such that $\chi'[\psi/p]$ is the same formula as χ, and $\varphi \gg \chi'$.*

Proof. Since $\langle p := \psi; (p := \psi)^*\rangle\varphi$ and $\langle p := \psi; (p := \psi)^*\rangle\chi$ are clean, based on (Renaming$_{\mathsf{MISL}}$) we can assume that all formulas φ, ψ, χ, $\varphi[\psi/p]$ and $\chi[\psi/p]$ are clean (Remark 4.5). We consider the different cases of the main connective of φ, and on the basis of the proof for Proposition 3.9, we only need to consider the case that φ is of the form $\langle q := \psi_0; (q := \psi_1)^*\rangle\theta$. We now consider two different situations.

(1) Proposition q is different from p. Assume that $\varphi \gg \psi$. Then, χ is $\langle q := \psi_0; (q := \psi_1)^n\rangle\theta$ for some natural number n. Now, formula $\varphi[\psi/p]$ is $\langle q := \psi_0[\psi/p]; (q := \psi_1[\psi/p])^*\rangle(\theta[\psi/p])$, and formula $\chi[\psi/p]$ is $\langle q := \psi_0[\psi/p]; (q := \psi_1[\psi/p])^n\rangle(\theta[\psi/p])$, which is the same formula as $(\langle q := \psi_0; (q := \psi_1)^n\rangle\theta)\,[\psi/p]$. Immediately, $\varphi[\psi/p] \gg \chi[\psi/p]$. On the other hand, if $\varphi[\psi/p] \gg \chi$, then χ is $(\langle q := \psi_0; (q := \psi_1)^n\rangle\theta)\,[\psi/p]$ for some natural number n. Setting $\chi' = \langle q := \psi_0; (q := \psi_1)^n\rangle\theta$, we have χ is the same as $\chi'[\psi/p]$ and $\varphi \gg \chi'$.

(2) Proposition q is p. Then, formula $\varphi[\psi/p]$ is $\langle q := \psi_0[\psi/p]; (q := \psi_1)^*\rangle(\theta[\psi/p])$, and formula $\chi[\psi/p]$ is $\langle q := \psi_0[\psi/p]; (q := \psi_1)^n\rangle(\theta[\psi/p])$, which is the same formula as $(\langle q := \psi_0; (q := \psi_1)^n\rangle\theta)\,[\psi/p]$. The rest follows via a similar argument as (1).

To sum up, $\varphi[\psi/p] \gg \chi[\psi/p]$ always holds. □

Lemma 4.9. *The order $\ll$ defined in Definition 4.7 is well-founded.*

Proof. We prove that any clean $\varphi \in \mathcal{L}_{\mathsf{MISL}}$ only spans finite descending $\ll$-chains, by induction, first on $\mathsf{DOI}(\varphi) = m$ and then on its length k.

(1) For the base case that $m = 0$, using the argument in the proof for Lemma 3.10, we can show that any φ can only span finite descending $\ll$-chains.

(2) For the induction step, let $1 \leqslant m \in \mathbb{N}$ and suppose the statement is true for (i) every formula ψ with $\mathsf{DOI}(\psi) < m$ and (ii) every formula χ such that $\mathsf{DOI}(\chi) = m$ and its length is smaller than k. We consider the case that $\mathsf{DOI}(\varphi) = m$ and the length is k.

(2.1) When φ is $\neg\psi$, $\psi_1 \wedge \psi_2$ or $\Box\psi$, it only descends along $\ll$ to strictly shorter formulas without making the depth of iterative substitution operators bigger. By induction hypothesis, any such formula only spans finite descending $\ll$-chains. Therefore φ only spans finite descending $\ll$-chains.

(2.2) When φ is $\langle p := \psi\rangle\chi$ and the main connective of χ is not an iterative substitution operator, the case can be proved with a similar way to that of Lemma 3.10. We skip the details.

(2.3) When φ is $\langle p := \psi; (p := \theta)^*\rangle\chi$, for every $n \in \mathbb{N}$, it descends to $\langle p := \psi; (p := \theta)^n\rangle\chi$. For each $\delta \in \{\psi, \theta, \chi\}$, either (i) $\mathsf{DOI}(\delta) < m$ or (ii) $\mathsf{DOI}(\delta) = m$ and the length of δ is smaller than k. By induction hypothesis, each of them spans finite descending $\ll$-chains, and we write the following for three finite chains spanned by them:

$$\begin{aligned}\varepsilon_\psi: &\quad \psi \gg \psi_1 \gg \psi_2 \gg \cdots \gg q_\psi\\ \varepsilon_\theta: &\quad \theta \gg \theta_1 \gg \theta_2 \gg \cdots \gg q_\theta\\ \varepsilon_\chi: &\quad \chi \gg \chi_1 \gg \chi_2 \gg \cdots \gg q_\chi\end{aligned}$$

By definition, $\langle p := \psi; (p := \theta)^n\rangle\chi$ descends to $(\langle(p := \theta)^n\rangle\chi)[\psi/p]$, i.e., $\langle p := \theta[\psi/p]; (p := \theta)^{n-1}\rangle\chi$. It follows from Proposition 4.8 that:

$$\theta[\psi/p] \gg \theta_1[\psi/p] \gg \theta_2[\psi/p] \gg \cdots \gg q_\theta[\psi/p]$$

As the case in Lemma 3.10, if $q_\theta \neq p$, then the above is a finite descending $\ll$-chain spanned by $\theta[\psi/p]$, otherwise by adding ε_ψ to the chain above we can obtain a finite chain spanned by $\theta[\psi/p]$. We write ε_1 for the resulting chain.

Next, $\langle p := \theta[\psi/p]; (p := \theta)^{n-1}\rangle\chi$ descends to $\langle p := \theta[(\theta[\psi/p])/p]; (p := \theta)^{n-2}\rangle\chi$. Again, by Proposition 4.8,

$$\theta[(\theta[\psi/p])/p] \gg \theta_1[(\theta[\psi/p])/p] \gg \theta_2[(\theta[\psi/p])/p] \gg \cdots \gg q_\theta[(\theta[\psi/p])/p]$$

As the case above, we can still get a finite $\ll$-chain spanned by $\theta[(\theta[\psi/p])/p]$, denoted ε_2.

In what follows, we call $\theta[(\theta[\psi/p])/p]$ '2-*layer* $\theta[\psi/p]$', $\theta[(\theta[(\theta[\psi/p])/p])/p]$ '3-*layer* $\theta[\psi/p]$', and so on. Repeating the reasoning above, we can get a finite $\ll$-chain ε_n spanned by the n-layer $\theta[\psi/p]$. Now we consider the following

$$\chi[\dagger/p] \gg \chi_1[\dagger/p] \gg \chi_2[\dagger/p] \gg \cdots \gg q_\chi[\dagger/p]$$

where $\dagger$ is the n-layer $\theta[\psi/p]$. Like before, if q_χ is different p, then the above is a finite chain spanned by $\chi[\dagger/p]$, and if q_χ is p, then by adding ε_n to the sequence above we obtain a finite chain spanned by $\chi[\dagger/p]$. Now one can see that for any $n \in \mathbb{N}$, $\langle p := \psi; (p := \theta)^n\rangle\chi$ only spans finite descending chains. Hence φ only spans finite descending $\ll$-chains. This completes the proof. □

We then define *the depth of a clean formula in* $\mathcal{L}_{\mathsf{MISL}}$ $\mathsf{dep}(\varphi)$ to be the (ordinal) length of the longest descending $\ll$-chains spanned by φ. The details are as follows:

Definition 4.10 (Depth of a clean $\mathcal{L}_{\mathsf{MISL}}$-formula)**.** *For a clean* $\varphi \in \mathcal{L}_{\mathsf{MISL}}$*, its* ordinal depth $\mathsf{dep}(\varphi)$ *is defined recursively by extending the clauses for* $\mathcal{L}_{\mathsf{MSL}}$ *in Definition 3.11 with the following:*

$$\mathsf{dep}(\langle p := \psi; (p := \chi)^*\rangle\varphi) := \sup_{n\in\mathbb{N}}\{\mathsf{dep}(\langle p := \psi; (p := \chi)^n\rangle\varphi)\} + 1$$

In the definition above, $\mathsf{dep}(\langle p := \psi; (p := \chi)^*\rangle\varphi)$ is always no less than $\omega + 1$. Now we are ready to prove Theorem 4.6 by induction along $\ll$, which is similar to the proof of Theorem 3.6.

Proof of Theorem 4.6. It goes by transfinite induction on $\mathsf{dep}(\varphi[\psi/p])$. The base case, as well as the cases where the main connective of φ is $\neg$, $\wedge$ or $\Diamond$ can be proved with the same argument as that in the proof of Theorem 3.6. Since φ is clean, the main connective of φ cannot be an iterative substitution operator. Therefore we are left with the situation where φ is $\langle q := \chi_1\rangle\chi_2$. There are two different situations.

(1) The main connective of χ_2 is not an iterative substitution operator, in which case χ_2 is a clean formula and so $\mathsf{dep}(\chi_2)$ is well defined. Then again we can prove the following equivalences through the same argument as that in the proof of Theorem 3.6, while changing n for depth of clean formulas to be an ordinal number:

$$\begin{aligned}\langle p := \psi\rangle\langle q := \chi_1\rangle\chi_2 &\leftrightarrow \langle p := \psi\rangle(\chi_2[\chi_1/q])\\ &\leftrightarrow (\chi_2[\chi_1/q])[\psi/p]\\ &\leftrightarrow (\langle q := \chi_1\rangle\chi_2)[\psi/p]\end{aligned}$$

This shows that $\langle p := \psi\rangle\varphi \leftrightarrow \varphi[\psi/p]$.

(2) The main connective of χ_2 is an iterative substitution operator. Since $\varphi = \langle q := \chi_1\rangle\chi_2$ is clean, χ_2 can be written as $\langle(q := \chi_3)^*\rangle\chi_4$. Now, for any $n \in \mathbb{N}$, $\mathsf{dep}(\varphi) > \mathsf{dep}(\langle q := \chi_1; (q := \chi_3)^n\rangle\chi_4)$, and by induction hypothesis, the following is valid:

$$\langle p := \psi\rangle\langle q := \chi_1; (q := \chi_3)^n\rangle\chi_4 \leftrightarrow (\langle q := \chi_1; (q := \chi_3)^n\rangle\chi_4)[\psi/p].$$

Let $(\boldsymbol{M}, w)$ be a pointed model. Assume that $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \langle p := \psi\rangle\langle q := \chi_1; (q := \chi_3)^*\rangle\chi_4$. Then, for some $m \in \mathbb{N}$, $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \langle p := \psi\rangle\langle q := \chi_1; (q := \chi_3)^m\rangle\chi_4$. So, $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} (\langle q := \chi_1; (q := \chi_3)^m\rangle\chi_4)[\psi/p]$, which then gives us $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} (\langle q := \chi_1; (q := \chi_3)^*\rangle\chi_4)[\psi/p]$. The converse can be proved in a similar way. Therefore, $\langle p := \psi\rangle\langle q := \chi_1; (q := \chi_3)^*\rangle\chi_4 \leftrightarrow (\langle q := \chi_1; (q := \chi_3)^*\rangle\chi_4)[\psi/p]$ is valid. The proof is completed. □

4.2. Example revisited: backward induction

Let us now get back to the applications of MISL, and establish a connection between backward induction as described in Section 1 and the corresponding iterative substitution operators. With respect to the board games defined in Scenario 2, we start by introducing several key notions, which are useful for both our proof and the illustration of the advantages of our approach:

Definition 4.11 (Ranks of winning positions)**.** *Let $\mathcal{B} = (W, R)$ be a finite game board. For any $i \in \{0, 1\}$ and any $k \in \mathbb{N}$, we define $\mathsf{Win}_i^k \subseteq W$ such that for any $s \in W$, $s \in \mathsf{Win}_i^k$ when player i has a strategy to win the game $\mathcal{G}_s = (\mathcal{B}, s)$ within k rounds. For all $0 < k \in \mathbb{N}$, the positions in $\mathsf{Win}_i^k \backslash \mathsf{Win}_i^{k-1}$ are called* rank-k winning positions of player i*, which means that player i has a strategy to win the game $\mathcal{G}_t$ in exactly k rounds. For the case when $k = 0$, the* rank-0 winning positions of player i *are exactly the members of Win_i^0. In addition, when a node u is not a finite rank winning position of any player, we call it* a rank-∞ winning position of player 0*, which means that player 0 has a strategy to force $\mathcal{G}_u$ to run infinitely.*

For player 1, we define $\mathsf{Win}_1 := \bigcup_{k \in \mathbb{N}} \mathsf{Win}_1^k$, consisting of the winning positions of player 1. Now, we are ready to show that the principle $\langle p := \Box\bot; (p := p \vee \Box\Diamond p)^*\rangle p$ involved in Section 1 defines the set:

Proposition 4.12. *Let $\mathcal{B} = (W, R)$ be a finite board and $\boldsymbol{M}$ be a model extending $\mathcal{B}$ with an arbitrary valuation function. Then, in the model, $\langle p := \Box\bot; (p := p \vee \Box\Diamond p)^k\rangle p$ is true exactly at the nodes in Win_1^k. More generally,*

$$\langle p := \Box\bot; (p := p \vee \Box\Diamond p)^*\rangle p$$

captures the set Win_1 of winning positions of player 1.

Proof. For convenience, let $R[s] = \{t : Rst\}$ denote the successors of $s \in W$. Define the function $[R] : \mathcal{P}(W) \to \mathcal{P}(W)$ as $[R] : X \mapsto \{s : R[s] \subseteq X\}$, and its dual function is $\langle R\rangle : X \mapsto \{s : R[s] \cap X \neq \varnothing\}$. Then, $\mathsf{Win}_1^0 = \{s : R[s] = \varnothing\}$. Using backward induction, one can see that

$$\mathsf{Win}_1^{k+1} = \bigcup_{i<k+1} \mathsf{Win}_1^i \cup [R](\langle R\rangle(\mathsf{Win}_1^k)),$$

which implies that if the truth set of p in the k-th stage of the iterative substitution is Win_1^k, then the truth set of p in the $(k+1)$-th stage is Win_1^{k+1}. Therefore, by induction, the truth set of p in the k-th stage of the iterative substitution is exactly Win_1^k. As a direct consequence, the truth set of $\langle p := \Box\bot; (p := p \vee \Box\Diamond p)^*\rangle p$ in $\boldsymbol{M}$ is $\bigcup_k \mathsf{Win}_1^k = \mathsf{Win}_1$. □

It is also instructive to point out that on a finite board, Win_1 can also be expressed by the formula

$$\mu p.(\Box\bot \vee \Box\Diamond p)$$

of *the modal μ-calculus* μML [23, 24]. We would like to argue that MISL has the capacity to analyze the nuances of winning positions' ranks in greater detail, though this discussion will be reserved for the following section.

Digression Finally, let us briefly note that such the interplay between MISL and game-theoretic reasoning is widespread. Beyond the example provided, another instance is $\exists n \in \mathbb{N}\ (\Box\Diamond)^n p$, which is proposed and discussed in the context of the hide and seek game logic [30]. However, the formula is not well-defined due to its infinite length in the cited work, but now it can be defined easily in MISL as $\langle(p := \Box\Diamond p)^*\rangle p$. It is crucial for characterizing the existence of winning strategies in the games analyzed in [30]. For an in-depth exploration of how the substitution operators capture the concept of $\exists n \in \mathbb{N}\ (\Box\Diamond)^n p$, we refer to [5], and for more interactions between games and logic, we refer to, e.g., [31, 32].

5. On relations between MISL and other relevant logics

In Section 4.2, we have shown that MISL can be used to reason about the winning positions for players, and in particular, given a finite game board, the formula $\langle p := \Box\bot; (p := p \vee \Box\Diamond p)^*\rangle p$ captures the winning positions Win_1 of player 1. However, as mentioned, this can also be defined with μML. So, what is the advantage of proposing another logic, viz. MISL? This question motivates us to explore the connections between MISL and other logical systems involving iterative computation, including μML, the *infinitary modal logic* ML^∞ [25] extending the standard modal logic with countable conjunctions and disjunctions, and *the propositional dynamic logic* PDL [29]. Moreover, the investigation into relation between MISL and *iterative modal relativization* IMR [15] is suggested in [5], and we will also provide results concerning their connection in the next section.

5.1. *Relation between* MISL *and* μML

We begin with the connections between MISL and μML. First of all, let us note the following:

Remark 5.1. *Since* MISL *only allows finite steps of iteration, for simplicity, in what follows we only consider the fragment of* μML *where the fixed point of every formula can be reached by a finite sequence of approximation, although in general* μML *admits fixed point reached by ordinal sequences of approximation as well.*

As mentioned, MISL is capable of reasoning about the ranks of winning positions (cf. Definition 4.11). For instance, consider the formula

$$\langle p := \Box\bot; (p := p \vee \Box\Diamond p)^*\rangle(\Diamond p \wedge \Diamond(\neg p \wedge \Box\Diamond p))$$

saying that the current state has different successors that are player 1's winning positions, and their ranks differ by one. In Figure 3, node d is of rank 0 and node c is of rank 1. This property cannot be defined by μML:

Theorem 5.2. *The formula*

$$\langle p := (\Box\bot \wedge b) \vee (\Diamond b \wedge a); (p := \Diamond p)^*\rangle (p \wedge (\langle q := b \wedge p; (q := \Diamond q)^*\rangle q))$$

is not definable by μML.

Proof. For simplicity, we write φ for the formula. We try to arrive at a contradiction by assuming that there exists a formula φ_μ of μML that is equivalent to φ. Since every formula of μML can be translated into a formula of *monadic second order logic* MSO (see e.g., [33]), our assumption indicates that there is an MSO-formula $\alpha(x)$ such that for any pointed model $(\boldsymbol{M}, s)$, $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \varphi$ iff $\boldsymbol{M}, [x \mapsto s] \vDash_{\mathsf{MSO}} \alpha(x)$, where the latter means that when we assign the value s to the variable x, $\alpha(x)$ is true in $\boldsymbol{M}$.

We will make use of the following structure. Set alphabet $\Sigma = \{a, b\}$. Any word $w \in \Sigma^*$ can be viewed as a model $\boldsymbol{M}_w = (\{1, 2, \dots, |w|\}, P_a, P_b, <)$ such that:

- $<$ the usual 'less than' relation on natural numbers,
- P_a and P_b are unary predicates such that a natural number i of the domain only satisfies P_{w_i} where w_i is the i-th symbol of w.

Also, for any $i, j \in \mathbb{N}$, we will use $i \leqslant j$ for $i < j \vee i = j$. *A formula* ψ *of* MSO *is said to define a language* $\mathbb{L}$ when it is the case that $w \in \mathbb{L}$ iff $\boldsymbol{M}_w \vDash_{\mathsf{MSO}} \psi$. From Büchi's theorem [34], we know that MSO-formulas define exactly the *regular languages*. Now, consider the following set of MSO-formulas:

$$\psi_1 := \exists x P_a x \wedge \exists x P_b x \wedge \forall x(P_a x \vee P_b x)$$
$$\psi_2 := \forall x \forall y\, ((P_a x \wedge P_b y) \to (x < y))$$
$$\psi_3 := \exists x\, (\alpha(x) \wedge \forall y (x \leqslant y))$$

Let $\psi := \psi_1 \wedge \psi_2 \wedge \psi_3$. We show that ψ defines the non-regular language $\{a^m b^n : 0 < m \leqslant n\}$. It is simple to see that if $\boldsymbol{M}_w \vDash_{\mathsf{MSO}} \psi_1 \wedge \psi_2$, then $w \in \{a^m b^n : m, n > 0\}$. Then, for any $w = a^m b^n$, $\boldsymbol{M}_w \vDash_{\mathsf{MSO}} \psi_3$ if and only if $\boldsymbol{M}_w, 1 \vDash_{\mathsf{MISL}} \varphi$, i.e., for some natural numbers i, j,

$$\boldsymbol{M}_w, 1 \vDash_{\mathsf{MISL}} \langle p := (\Box\bot \wedge b) \vee (\Diamond b \wedge a); (p := \Diamond p)^i\rangle \left(p \wedge \left(\langle q := b \wedge p; (q := \Diamond q)^j\rangle q\right)\right).$$

Initially, p is satisfied on state m (where $\Diamond b \wedge a$ holds) and state $(m+n)$ (where $\Box\bot \wedge b$ holds). After i iterations of the substitution $p := \Diamond p$, p is satisfied on state 1, and moreover, there exists a state labeled by $(j+1)$ where $b \wedge p$ holds. Such condition holds if and only if $i = m-1, j = n, n \geqslant m > 0$. When i, j range over all natural numbers, the satisfying words w are those belonging to the non-regular language $\{a^m b^n : 0 < m \leqslant n\}$. Figure 5 provides such a model $\boldsymbol{M}_w$.

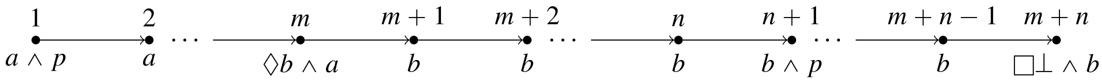

Figure 5. A model $\boldsymbol{M}_w$ such that $\boldsymbol{M}_w \vDash_{\mathsf{MSO}} \psi_1 \wedge \psi_2 \wedge \psi_3$, where $w = a^m b^n$ and $n \geqslant m > 0$. The propositional letter p is marked on its valuation after the substitution $\langle p := (\Box\bot \wedge b) \vee (\Diamond b \wedge a); (p := \Diamond p)^{m-1}\rangle$.

Therefore the original MISL-formula φ is not equivalent to any μML-formula. □

On the other hand, recall that to calculate the least fixed-point of a monotone function in μML, one starts with $\bot$ and computes the approximation sequence. As mentioned in [5], if the least fixed point of φ can be reached by a finite sequence of approximation, then the μML-formula $\mu p.\varphi$ is equivalent to

$$\langle p := \bot; (p := \varphi)^* \rangle p.$$

Now, combining this and Theorem 5.2, we have the following as a corollary:

Corollary 5.3. *When only finite sequences of approximation are allowed,* μML *is strictly contained in* MISL.

It is important to emphasize that the equivalence between $\mu p.\varphi$ and $\langle p := \bot; (p := \varphi)^* \rangle p$ is based on the convention that in μML only finite sequences of approximation are allowed (Remark 5.1). And, we do *not* have the equivalence when ordinal sequences are allowed in μML: for instance, $\mu p.\Box p$ is true $(\boldsymbol{M}, s)$ depicted in Figure 6 while $\langle p := \bot; (p := \Box p)^* \rangle p$ is false there. For the general case without the restriction imposed in the corollary above, it remains to be determined whether or not μML is still contained in MISL.

5.2. *Relation between* MISL *and* ML$^\infty$

Let us now proceed to discuss ML$^\infty$. As stated in [26], μML under the restriction imposed in Remark 5.1 can be embedded into ML$^\infty$. Now, we provide an analogous result stating that MISL is also a fragment of ML$^\infty$:

Theorem 5.4. *There is a translation $T : \mathcal{L}_{\mathsf{MISL}} \to \mathcal{L}_{\mathsf{ML}^\infty}$ such that for any formula $\varphi \in \mathcal{L}_{\mathsf{MISL}}$, it holds that:*

$$\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \varphi \quad \textit{iff} \quad \boldsymbol{M}, s \vDash_{\mathsf{ML}^\infty} T(\varphi).$$

Proof. We are going to show explicitly the construction of T. Let $T = T_0 \circ T_1$, where $T_0 : \mathcal{L}_{\mathsf{clean}} \to \mathcal{L}_{\mathsf{ML}^\infty}$ is a translation from clean MISL-formulas to ML$^\infty$-formulas and $T_1 : \mathcal{L}_{\mathsf{MISL}} \to \mathcal{L}_{\mathsf{clean}}$ is a translation from arbitrary MISL-formulas to the clean ones. The translation T_1 can be obtained by bound variable renaming (Renaming$_{\mathsf{MISL}}$), and T_0 keeps atoms p the same, permutes with Boolean connectives and $\Diamond$, and

$$T_0(\langle p := \psi \rangle \varphi) := T_0(\varphi)[T_0(\psi)/p]$$
$$T_0(\langle p := \psi_0; (p := \psi_1)^* \rangle \varphi) := \bigvee\nolimits_{n \in \mathbb{N}} T_0(\langle p := \psi_0; (p := \psi_1)^n \rangle \varphi)$$

where formula φ in the clause for $T_0(\langle p := \psi \rangle \varphi)$ does not have its main connective being an iterative substitution operator.[8] It can be verified that the translation T_0 and thus T translate formulas to their equivalent forms. Therefore MISL can be translated to ML$^\infty$. □

W.r.t. the relation between MISL and ML$^\infty$, it remains to determine the following:

Open problem Is MISL strictly contained in ML$^\infty$? Are there some classes of models over which they are equivalent?

[8]Also, note that the replacement $[T_0(\psi)/p]$ of the variable p is carried out in an ML$^\infty$-formula, which substitutes all occurrences of propositional letter p with $T_0(\psi)$.

5.3. *Relation between* MISL *and* PDL

Finally, let us turn to another logical framework, PDL, involving finite steps of iteration. Since the language $\mathcal{L}_{\mathsf{PDL}}$ may contain more than one primitive programs (e.g., $\Delta = \{a, b, \dots\}$), we also add the corresponding modalities $\langle a\rangle, \langle b\rangle, \dots$ to the language of MISL, and write $\mathcal{L}_{\mathsf{MISL}(\Delta)}$ and $\mathsf{MISL}(\Delta)$ for the resulting language and logic respectively. We now show that PDL without the test programs $\varphi?$ can be embedded into $\mathsf{MISL}(\Delta)$:

Theorem 5.5. *There exists a translation $T : \mathcal{L}_{\mathsf{PDL}} \to \mathcal{L}_{\mathsf{MISL}(\Delta)}$ such that for any* PDL*-formula φ, it holds that*

$$\boldsymbol{M}, s \vDash_{\mathsf{PDL}} \varphi \quad \textit{iff} \quad \boldsymbol{M}, s \vDash_{\mathsf{MISL}(\Delta)} T(\varphi).$$

where $\boldsymbol{M}$ is a model containing relations R_c with $c \in \Delta$.

Proof. A desired translation T keeps propositional variables the same, permutes with Boolean connectives and the modalities $\langle a\rangle$ for primitive programs, and

$$\begin{aligned} T(\langle \pi_1; \pi_2\rangle\varphi) &:= T(\langle \pi_1\rangle\langle \pi_2\rangle\varphi) \\ T(\langle \pi_1 \cup \pi_2\rangle\varphi) &:= T(\langle \pi_1\rangle\varphi) \vee T(\langle \pi_2\rangle\varphi) \\ T(\langle \pi^*\rangle\varphi) &:= \langle q := T(\varphi); (q := T(\langle \pi\rangle q))^*\rangle q \end{aligned}$$

where $\pi_1; \pi_2$, $\pi_1 \cup \pi_2$, and π^* are complex programs, and q is a fresh variable. We leave the proof for the correctness of the translation to the reader. □

Having seen a number of expressive languages that can be encoded within MISL,[9] we guess that the complexity of the framework should be high. This is confirmed by the findings presented in the next section, which studies various properties of MISL.

6. Expressiveness and undecidability of MISL

In this section, we explore further properties of MISL, including its expressive power and the decidability of its satisfiability problem. We will show that even though MISL is much more powerful than ML, truth of MISL-formulas is still invariant under the standard notion of bisimulation for ML (Section 6.1); meanwhile, the logic will be proved to be highly undecidable: its satisfiability problem is Σ^1_1-complete (Section 6.2). Moreover, when we restrict our attention to the very simple class of finite tree models, the logic is still shown to be undecidable (Section 6.3).

6.1. *Bisimulation for* MISL

Let us start by considering the expressive power of MISL. Specifically, we will first introduce the details for the notion of bisimulation for ML, and then show that although MISL is equipped with additional substitution operators, it is still invariant under the established notion of bisimulation.

Definition 6.1 (Bisimulation [27])**.** *Let $\boldsymbol{M} = (W, R, V)$ and $\boldsymbol{M}' = (W', R', V')$ be two models and let $s \in W$ and $s' \in W'$. A non-empty relation $Z \subseteq W \times W'$ is called a* bisimulation between $\boldsymbol{M}$ and $\boldsymbol{M}'$*, if the following are satisfied:*

Atom: *If sZs', then $(\boldsymbol{M}, s)$ and $(\boldsymbol{M}', s')$ satisfy the same propositional letters.*

Zig: *If sZs' and there exists $t \in W$ such that Rst, then there exists $t' \in W'$ such that $R's't'$ and tZt'.*

Zag: *If sZs' and there exists $t' \in W'$ such that $R's't'$, then there exists $t \in W$ such that Rst and tZt'.*

When there is a bisimulation Z linking states $s \in \boldsymbol{M}$ and $s' \in \boldsymbol{M}'$, we say that $(\boldsymbol{M}, s)$ and $(\boldsymbol{M}', s')$ are *bisimilar*, and write $(\boldsymbol{M}, s) \leftrightarroweq (\boldsymbol{M}', s')$, or just $s \leftrightarroweq s'$ when the models are clear from the context.

[9] Moreover, the undecidable framework iterative modal relativization IMR [15] can also be encoded within MISL (see Definition 6.7).

Theorem 6.2 (Invariance under bisimulation). *If* $(\boldsymbol{M}, s) \leftrightarroweq (\boldsymbol{M}', s')$, *then for any* $\varphi \in \mathcal{L}_{\mathsf{MISL}}$,

$$\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \varphi \text{ iff } \boldsymbol{M}', s' \vDash_{\mathsf{MISL}} \varphi.$$

Proof. We prove by induction on formulas φ. When φ is an ML-formula, it is easy to see that the equivalence holds. We now consider other cases. In what follows, we write Z for a bisimulation that links s and s'.

(1) Formula φ is $\langle p := \psi\rangle\chi$. Assume that $(t, t') \in Z$. We are going to show that $(\boldsymbol{M}|_{p:=\psi}, t) \leftrightarroweq (\boldsymbol{M}'|_{p:=\psi}, t')$. Models $\boldsymbol{M}|_{p:=\psi}$ and $\boldsymbol{M}$ may only differ on the truth set of p, and similarly for $\boldsymbol{M}'|_{p:=\psi}$ and $\boldsymbol{M}'$. So, we only need to prove that $\boldsymbol{M}|_{p:=\psi}, t$ and $\boldsymbol{M}'|_{p:=\psi}, t'$ agree on the propositional variable p. By induction hypothesis, $\boldsymbol{M}, t \vDash_{\mathsf{MISL}} \psi$ iff $\boldsymbol{M}', t' \vDash_{\mathsf{MISL}} \psi$. Hence, $\boldsymbol{M}|_{p:=\psi}, t \vDash_{\mathsf{MISL}} p$ iff $\boldsymbol{M}'|_{p:=\psi}, t' \vDash_{\mathsf{MISL}} p$. Therefore, $(\boldsymbol{M}|_{p:=\psi}, s) \leftrightarroweq (\boldsymbol{M}'|_{p:=\psi}, s')$. Again, by induction hypothesis, $\boldsymbol{M}|_{p:=\psi}, s \vDash_{\mathsf{MISL}} \chi$ iff $\boldsymbol{M}'|_{p:=\psi}, s' \vDash_{\mathsf{MISL}} \chi$. Thus, based on the semantics, $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle p := \psi\rangle\chi$ iff $\boldsymbol{M}', s' \vDash_{\mathsf{MISL}} \langle p := \psi\rangle\chi$.

(2) Formula φ is $\langle(p := \psi)^*\rangle\chi$. Assume that $(t, t') \in Z$. We show by induction that for all $n \in \mathbb{N}$, it holds that $(\boldsymbol{M}|_{(p:=\psi)^n}, t) \leftrightarroweq (\boldsymbol{M}'|_{(p:=\psi)^n}, t')$. The base case $n = 0$ is direct. Also, for the case that $n = 1$, we have known from the reasoning in case (1) that $(\boldsymbol{M}|_{p:=\psi}, t) \leftrightarroweq (\boldsymbol{M}'|_{p:=\psi}, t')$. For any $n \geqslant 2$, by induction hypothesis, we have $(\boldsymbol{M}|_{(p:=\psi)^n}, t) \leftrightarroweq (\boldsymbol{M}'|_{(p:=\psi)^n}, t')$. Now, using the reasoning similar to that in case (1), we can show that $(\boldsymbol{M}|_{(p:=\psi)^{n+1}}, t) \leftrightarroweq (\boldsymbol{M}'|_{(p:=\psi)^{n+1}}, t')$.

Now assume that $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle(p := \psi)^*\rangle\chi$. Then, $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle(p := \psi)^m\rangle\chi$ for some $m \in \mathbb{N}$, which indicates that $\boldsymbol{M}|_{(p:=\psi)^m}, s \vDash_{\mathsf{MISL}} \chi$. It follows from the reasoning above that $(\boldsymbol{M}|_{(p:=\psi)^m}, s) \leftrightarroweq (\boldsymbol{M}'|_{(p:=\psi)^m}, s')$. By induction hypothesis, $\boldsymbol{M}|_{(p:=\psi)^m}, s \vDash_{\mathsf{MISL}} \chi$ implies $\boldsymbol{M}'|_{(p:=\psi)^m}, s' \vDash_{\mathsf{MISL}} \chi$. So, $\boldsymbol{M}', s' \vDash_{\mathsf{MISL}} \langle(p := \psi)^m\rangle\chi$, which gives us $\boldsymbol{M}', s' \vDash_{\mathsf{MISL}} \langle(p := \psi)^*\rangle\chi$. The converse direction can be proved similarly. This completes the proof. □

6.2. *Satisfiability problem of* MISL*:* Σ^1_1*-complete*

In this part, we prove that the satisfiability problem for MISL is undecidable, by encoding the same for the *iterative modal relativization* IMR, whose complexity is known to be Σ^1_1-complete [15]. Moreover, as we shall see, the logic is undecidable even when we consider certain simple classes of models, for example, finite tree models. As a warm-up, we first show the following:

Proposition 6.3. MISL *lacks the finite model property.*

Proof. It suffices to construct an MISL-formula that can only have infinite models. Consider the following:

$$[p := \Box\bot; (p := \neg p \wedge \Box p)^*]\Diamond p.$$

First, the formula is satisfiable. One can check that it is true at the state s in model $\boldsymbol{M}$ depicted in Figure 6.

Next, given a model in which the formula is true, one can see that the truth set of p, obtained at the n-stage of the iteration, consists of the states of height n, and the formula says that there is no finite upper bound on the height of the current state. Therefore, any model satisfying the formula must be infinite, as needed. □

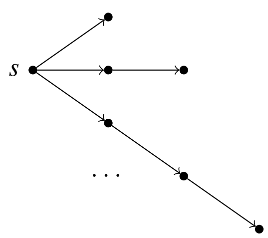

Figure 6. An infinite model $\boldsymbol{M}$ such that $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} [p := \Box\bot; (p := \neg p \wedge \Box p)^*]\Diamond p$

Next we figure out the exact complexity of the satisfiability problem for MISL:

Theorem 6.4 (Undecidability of MISL)**.** *The satisfiability problem for* MISL *is undecidable and is* Σ^1_1*-complete.*

To prove Theorem 6.4, we first show the satisfiability problem for MISL is undecidable and Σ^1_1-hard, by embedding iterative modal relativization IMR [15] into MISL. Here is a quick introduction to IMR:

Basics of IMR The language $\mathcal{L}_{\mathsf{IMR}}$ of IMR extends $\mathcal{L}_{\mathsf{ML}}$ with the public announcement operators $[\psi]$ and their finitely iterative generalizations $[\psi^*]$, and the common knowledge operator $\Box^*$ (i.e., the reflexive-transitive closure of $\Box$).

To show that IMR can be translated into MISL, let us first prove that the simplest relativization operator $[p]$ can be mimicked with MISL. To facilitate discussion, let us define a function $(\cdot)^p$ on MISL-formulas such that $(\varphi)^p$ intuitively captures the relativization $[p]\varphi$. Details are as follows:

Definition 6.5 (Intermediate relativization)**.** *Let $p \in \mathbf{P}$ be a propositional variable. For any $\varphi \in \mathcal{L}_{\mathsf{MISL}}$ such that $p \notin bv(\varphi)$, we define $(\varphi)^p$ recursively as follows:*

$$
\begin{array}{rlrl}
(q)^p := & p \to q & (\neg\varphi)^p := & p \to \neg(\varphi)^p \\
(\varphi_1 \wedge \varphi_2)^p := & (\varphi_1)^p \wedge (\varphi_2)^p & (\Box\varphi)^p := & p \to \Box(\varphi)^p \\
(\langle q := \varphi_1\rangle\varphi_2)^p := & \langle q := (\varphi_1)^p\rangle(\varphi_2)^p & (\langle(q := \varphi_1)^*\rangle\varphi_2)^p := & \langle(q := (\varphi_1)^p)^*\rangle(\varphi_2)^p
\end{array}
$$

In the definition above, the restriction $p \notin bv(\varphi)$ is used to exclude the situation that p is bounded by some substitution operators in φ. Let us now show the following, which intuitively states that $(\varphi)^p$ is equivalent to $[p]\varphi$:

Lemma 6.6. *Let $\boldsymbol{M} = (W, R, V)$ be a model, $s \in W$ and $p \in \mathbf{P}$. For any $\varphi \in \mathcal{L}_{\mathsf{MISL}}$ such that $p \notin bv(\varphi)$, it holds that*

$$\boldsymbol{M}, s \vDash_{\mathsf{MISL}} (\varphi)^p \quad \textit{iff} \quad s \notin W^p \textit{ or } \boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \varphi,$$

where $\boldsymbol{M}^p = (W^p, R^p, V^p)$ is the model obtained by relativizing p in $\boldsymbol{M}$, i.e.,

- $W^p := \{s \in W : \boldsymbol{M}, s \vDash_{\mathsf{MISL}} p\}$,
- $R^p := R \cap (W^p \times W^p)$, *and*
- *for any* $q \in \mathbf{P}$, $V^p(q) := V(q) \cap W^p$.

Proof. It is by induction on φ. The cases for the Boolean connectives and $\Box$ are straightforward. We now consider the cases for substitution operators.

(1) Formula φ is $\langle q := \psi\rangle\chi$. Then, by definition, $(\varphi)^p = \langle q := (\psi)^p\rangle(\chi)^p$. Since $p \notin bv(\varphi)$, p is different from q. We are going to show that $\boldsymbol{M}^p|_{q:=\psi}$ is the same model as $(\boldsymbol{M}|_{q:=(\psi)^p})^p$. Notice that if they can be different, then their difference can only be the truth sets of q. By induction hypothesis,

$$
\begin{aligned}
V^p|_{q:=\psi}(q) &= \{s \in W^p : \boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \psi\} \\
&= \{s \in W^p : \boldsymbol{M}, s \vDash_{\mathsf{MISL}} (\psi)^p\} \\
&= (V|_{q:=(\psi)^p})^p(q)
\end{aligned}
$$

Hence the two models are the same. Therefore,

$$
\begin{aligned}
\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle q := (\psi)^p\rangle\chi^p &\Leftrightarrow s \notin W^p \text{ or } (\boldsymbol{M}|_{q:=(\psi)^p})^p, s \vDash_{\mathsf{MISL}} \chi \\
&\Leftrightarrow s \notin W^p \text{ or } \boldsymbol{M}^p|_{q:=\psi}, s \vDash_{\mathsf{MISL}} \chi \\
&\Leftrightarrow s \notin W^p \text{ or } \boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \langle q := \psi\rangle\chi
\end{aligned}
$$

(2) Formula φ is $\langle(q := \psi)^*\rangle\chi$. Then, by definition, $(\varphi)^p = \langle(q := (\psi)^p)^*\rangle(\chi)^p$. Since $p \notin bv(\varphi)$, p is not q. Let us first prove that for any $n \in \mathbb{N}$, it holds that

$$\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle(q := (\psi)^p)^n\rangle\chi^p \Leftrightarrow s \notin W^p \text{ or } \boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \langle(q := \psi)^n\rangle\chi.$$

The case for $n = 0$ holds immediately by induction hypothesis. Let us consider the case that $n = k + 1$.

$$\begin{aligned} \boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle (q := (\psi)^p)^{k+1} \rangle (\chi)^p &\Leftrightarrow \boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle q := (\psi)^p \rangle \langle (q := (\psi)^p)^k \rangle (\chi)^p \\ &\Leftrightarrow \boldsymbol{M}|_{q:=(\psi)^p}, s \vDash_{\mathsf{MISL}} \langle (q := (\psi)^p)^k \rangle (\chi)^p \\ &\Leftrightarrow s \notin W^p \text{ or } (\boldsymbol{M}|_{q:=(\psi)^p})^p, s \vDash_{\mathsf{MISL}} \langle (q := \psi)^k \rangle \chi \\ &\Leftrightarrow s \notin W^p \text{ or } \boldsymbol{M}^p|_{q:=\psi}, s \vDash_{\mathsf{MISL}} \langle (q := \psi)^k \rangle \chi \\ &\Leftrightarrow s \notin W^p \text{ or } \boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \langle (q := \psi)^{k+1} \rangle \chi \end{aligned}$$

The third equivalence holds by induction hypothesis. The fourth can be obtained by the same reasoning as that in (1).

Now, assume that $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} (\langle (q := \psi)^* \rangle \chi)^p$. Then, $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle (q := (\psi)^p)^m \rangle (\chi)^p$ for some $m \in \mathbb{N}$. By the reasoning above, $s \notin W^p$ or $\boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \langle (q := \psi)^m \rangle \chi$. The latter implies that $\boldsymbol{M}^p, s \vDash_{\mathsf{MISL}} \langle (q := \psi)^* \rangle \chi$, as needed. The other direction can be proved in a similar way. This completes the proof. □

Now we are ready to show the details of the translation:

Definition 6.7 (Translation from IMR to MISL)**.** *The translation $T : \mathcal{L}_{\mathsf{IMR}} \to \mathcal{L}_{\mathsf{MISL}}$ is given by the following:*

$$T(p) := p \quad T(\neg\varphi) := \neg T(\varphi) \quad T(\varphi_1 \wedge \varphi_2) := T(\varphi_1) \wedge T(\varphi_2)$$
$$T(\Box\varphi) := \Box T(\varphi) \quad T(\Box^*\varphi) := [q := T(\varphi); (q := \Box q)^*]q$$
$$T([\psi]\varphi) := \langle q := T(\psi) \rangle (T(\varphi))^q \quad T(\langle \psi^* \rangle \varphi) := \langle q := \top; (q := q \wedge (T(\psi))^q)^* \rangle (T(\varphi))^q$$

where q is a fresh variable.

Theorem 6.8. *For any pointed model $(\boldsymbol{M}, s)$ and any formula $\varphi \in \mathcal{L}_{\mathsf{IMR}}$, we have the following:*

$$\boldsymbol{M}, s \vDash_{\mathsf{IMR}} \varphi \quad \textit{iff} \quad \boldsymbol{M}, s \vDash_{\mathsf{MISL}} T(\varphi)$$

where T is the translation given in Definition 6.7.

Proof. It is by induction on $\varphi \in \mathcal{L}_{\mathsf{IMR}}$. The cases for Boolean formulas and $\Box\varphi$ are easy to prove. We consider others.

(1) Formula φ is $\Box^*\psi$. By definition, $T(\varphi) = [q := T(\psi); (q := \Box q)^*]q$. Then, we have the following sequence:

$$\begin{aligned} \boldsymbol{M}, s \vDash_{\mathsf{IMR}} \varphi &\Leftrightarrow \text{for all } n \in \mathbb{N},\ \boldsymbol{M}, s \vDash_{\mathsf{IMR}} \Box^n \psi \\ &\Leftrightarrow \text{for all } n \in \mathbb{N},\ \boldsymbol{M}, s \vDash_{\mathsf{MISL}} \Box^n T(\psi) \\ &\Leftrightarrow \text{for all } n \in \mathbb{N},\ \boldsymbol{M}, s \vDash_{\mathsf{MISL}} [q := T(\psi)]\Box^n q \\ &\Leftrightarrow \text{for all } n \in \mathbb{N},\ \boldsymbol{M}, s \vDash_{\mathsf{MISL}} [q := T(\psi); (q := \Box q)^n]q \\ &\Leftrightarrow \boldsymbol{M}, s \vDash_{\mathsf{MISL}} [q := T(\psi); (q := \Box q)^*]q \\ &\Leftrightarrow \boldsymbol{M}, s \vDash_{\mathsf{MISL}} T(\varphi) \end{aligned}$$

The second equivalence holds by induction hypothesis.

(2) Formula φ is $[\psi]\chi$. Take a fresh variable q, and $T(\varphi) = \langle q := T(\psi) \rangle T(\chi)^q$. We write $\boldsymbol{M}^\psi = (W^\psi, R^\psi, V^\psi)$ for the model obtained by relativizing $\psi \in \mathcal{L}_{\mathsf{IMR}}$. By induction hypothesis, $\boldsymbol{M}, s \vDash_{\mathsf{IMR}} \psi$ iff $\boldsymbol{M}, s \vDash_{\mathsf{MISL}} T(\psi)$. So, W^ψ is the same as the domain of the model $(\boldsymbol{M}|_{q:=T(\psi)})^q$ that is obtained by relativizing q in $\boldsymbol{M}|_{q:=T(\psi)}$. Models $\boldsymbol{M}^\psi$ and $\left(\boldsymbol{M}|_{q:=T(\psi)}\right)^q$ may only differ on the valuation of q. Since q does not appear in χ, it holds that:

$$\boldsymbol{M}^\psi, s \vDash_{\mathsf{IMR}} \chi \Leftrightarrow \left(\boldsymbol{M}|_{q:=T(\psi)}\right)^q, s \vDash_{\mathsf{IMR}} \chi.$$

Then, the following sequence of equivalences holds:

$$\begin{aligned} \boldsymbol{M}, s \vDash_{\mathsf{IMR}} \varphi &\Leftrightarrow \boldsymbol{M}, s \nvDash_{\mathsf{IMR}} \psi \text{ or } \boldsymbol{M}^\psi, s \vDash_{\mathsf{IMR}} \chi \\ &\Leftrightarrow \boldsymbol{M}, s \nvDash_{\mathsf{IMR}} \psi \text{ or } \left(\boldsymbol{M}|_{q:=T(\psi)}\right)^q, s \vDash_{\mathsf{IMR}} \chi \\ &\Leftrightarrow \boldsymbol{M}|_{q:=T(\psi)}, s \nvDash_{\mathsf{MISL}} q \text{ or } \left(\boldsymbol{M}|_{q:=T(\psi)}\right)^q, s \vDash_{\mathsf{MISL}} T(\chi) \\ &\Leftrightarrow \boldsymbol{M}|_{q:=T(\psi)}, s \vDash_{\mathsf{MISL}} (T(\chi))^q \\ &\Leftrightarrow \boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle q := T(\psi) \rangle (T(\chi))^q \\ &\Leftrightarrow \boldsymbol{M}, s \vDash_{\mathsf{MISL}} T(\varphi) \end{aligned}$$

The third equivalence holds by induction hypothesis, and the fourth one is obtained by Lemma 6.6.

(3) Formula φ is $\langle\psi^*\rangle\chi$. Pick a fresh variable q, and $T(\varphi) = \langle q := \top; (q := q \wedge (T(\psi))^q)^*\rangle(T(\chi))^q$. For any $n \in \mathbb{N}$, we write $\boldsymbol{M}^{\psi^n}$ for the model obtained by repeatedly relativizing $\psi \in \mathcal{L}_{\mathsf{IMR}}$ for n times. By induction hypothesis, for any $(\boldsymbol{M}, s)$ and $\varphi_0 \in \{\psi, \chi\}$, we have the following:

$$\boldsymbol{M}, s \vDash_{\mathsf{IMR}} \varphi_0 \;\Leftrightarrow\; \boldsymbol{M}, s \vDash_{\mathsf{MISL}} T(\varphi_0). \qquad (\ddagger)$$

Now, let us first show by induction on $n \in \mathbb{N}$ that for any $n \in \mathbb{N}$, $\boldsymbol{M}^{\psi^n}$ and $\left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q$ may only disagree on the valuation of q.

For the base case that $n = 0$, $\boldsymbol{M}^{\psi^n} = \boldsymbol{M}$ and $\left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q = \boldsymbol{M}|_{q:=\top}$. Therefore the two models can only disagree on the valuation of q.

For induction step, suppose that all the cases for $n-1$ are proved, where $n \geqslant 1$. Then $\left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}\right)^q$ and $\boldsymbol{M}^{\psi^{n-1}}$ can only disagree on the valuation of q. Then, for the domain of $\boldsymbol{M}^{\psi^n} = \left(\boldsymbol{M}^{\psi^{n-1}}\right)^\psi$, we have the following:

$$\begin{aligned}
&\{s \in \boldsymbol{M}^{\psi^{n-1}} : \boldsymbol{M}^{\psi^{n-1}}, s \vDash_{\mathsf{IMR}} \psi\}\\
=&\{s \in \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}\right)^q : \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}\right)^q, s \vDash_{\mathsf{IMR}} \psi\}\\
=&\{s \in \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}\right)^q : \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}\right)^q, s \vDash_{\mathsf{MISL}} T(\psi)\}\\
=&\{s \in \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}} : \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}, s \vDash_{\mathsf{MISL}} q \textit{ and } \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}\right)^q, s \vDash_{\mathsf{MISL}} T(\psi)\}\\
=&\{s \in \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}} : \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}}, s \vDash_{\mathsf{MISL}} q \wedge (T(\psi))^q\}\\
=&\{s \in \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n-1}} : \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^{n}}, s \vDash_{\mathsf{MISL}} q\}
\end{aligned}$$

The first equation holds by induction hypothesis and the fact that q does not appear in ψ, while the second equation holds by ($\ddagger$). Also, notice that the last set in the sequence above is exactly the domain of $\left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q$. Now it is easy to see that $\boldsymbol{M}^{\psi^n}$ and $\left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q$ can only disagree on the valuation of q. Since q does not appear in χ, we have

$$\boldsymbol{M}^{\psi^n}, s \vDash_{\mathsf{IMR}} \chi \;\Leftrightarrow\; \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q, s \vDash_{\mathsf{IMR}} \chi.$$

Therefore, the following sequence holds:

$$\begin{aligned}
\boldsymbol{M}, s \vDash_{\mathsf{IMR}} [\psi^n]\chi \;&\Leftrightarrow\; s \notin \boldsymbol{M}^{\psi^n} \text{ or } \boldsymbol{M}^{\psi^n}, s \vDash_{\mathsf{IMR}} \chi\\
&\Leftrightarrow\; s \notin \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q \text{ or } \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q, s \vDash_{\mathsf{IMR}} \chi\\
&\Leftrightarrow\; s \notin \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q \text{ or } \left(\boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}\right)^q, s \vDash_{\mathsf{MISL}} T(\chi)\\
&\Leftrightarrow\; \boldsymbol{M}|_{q:=\top;(q:=q\wedge(T(\psi))^q)^n}, s \vDash_{\mathsf{MISL}} (T(\chi))^q\\
&\Leftrightarrow\; \boldsymbol{M}, s \vDash_{\mathsf{MISL}} \langle q := \top; (q := q \wedge (T(\psi))^q)^n\rangle(T(\chi))^q\\
&\Leftrightarrow\; \boldsymbol{M}, s \vDash_{\mathsf{MISL}} T(\varphi)
\end{aligned}$$

The third equivalence holds by ($\ddagger$), and the fourth one holds by Lemma 6.6. Now the proof is completed. □

So far, we have shown that the satisfiability problem for MISL is undecidable and Σ^1_1-hard. In what follows, we will show that the problem is in Σ^1_1, which concludes that MISL is Σ^1_1-complete. To do so, we define the *evaluation relation* that intuitively expresses whether or not a formula φ is true at a pointed model:

Definition 6.9. *Let $\boldsymbol{M} = (W, R, V)$ be a model and $\mathcal{L}_{\mathsf{clean}}$ be the set of clean* MISL*-formulas. We define a binary relation $E_{\boldsymbol{M}} \subseteq \mathcal{L}_{\mathsf{clean}} \times W$ such that:*

$$E_{\boldsymbol{M}}(\varphi, s) \text{ iff } \boldsymbol{M}, s \vDash_{\mathsf{MISL}} \varphi.$$

Now, w.r.t. a model $\boldsymbol{M}$ and its associated $E_{\boldsymbol{M}}$, we can show the following:

Lemma 6.10. *Given a model $\boldsymbol{M} = (W, R, V)$, for all $s \in W$, it holds that $V(p) = \{s : E_{\boldsymbol{M}}(p, s)\}$, and also, all the following properties hold for $E_{\boldsymbol{M}}$:*

$$\begin{aligned}
E_{\boldsymbol{M}}(\bot, s) &\leftrightarrow \bot \\
E_{\boldsymbol{M}}(\neg\varphi, s) &\leftrightarrow \neg E_{\boldsymbol{M}}(\varphi, s) \\
E_{\boldsymbol{M}}(\varphi_1 \wedge \varphi_2, s) &\leftrightarrow E_{\boldsymbol{M}}(\varphi_1, s) \wedge E_{\boldsymbol{M}}(\varphi_2, s) \\
E_{\boldsymbol{M}}(\Box\varphi, s) &\leftrightarrow \forall t \in W\,(Rst \to E_{\boldsymbol{M}}(\varphi, t)) \\
E_{\boldsymbol{M}}(\langle p := \psi\rangle\varphi, s) &\leftrightarrow E_{\boldsymbol{M}}(\varphi[\psi/p], s) \\
E_{\boldsymbol{M}}(\langle p := \psi_0; (p := \psi_1)^*\rangle\varphi, s) &\leftrightarrow \exists n \in \mathbb{N}\, E_{\boldsymbol{M}}(\langle p := \psi_0; (p := \psi_1)^n\rangle\varphi, s)
\end{aligned}$$

where the main connective of the formula φ in the clause for $E_{\boldsymbol{M}}(\langle p := \psi\rangle\varphi, s)$ is not an iterative substitution operator.

This lemma can be proved by induction on formulas in $\mathcal{L}_{\mathsf{clean}}$ along $\ll$ (with the help of Theorem 4.6). Notice that a formula occurring on the right-hand-side of a condition is 'less than' ($\ll$) the formula occurring on the left-hand-side of this condition. Now we are ready to prove that MISL is in Σ^1_1.

Theorem 6.11. *The satisfiability problem for* MISL *is in Σ^1_1.*

Proof. Since MISL can be treated as a fragment of ML^∞ (Theorem 5.4) and the latter has the downward Löwenheim-Skolem property, an $\mathcal{L}_{\mathsf{MISL}}$-formula is satisfiable iff it is satisfiable w.r.t. countable models $\boldsymbol{M} = (\omega, R, V)$.

In the sequel, we only need to consider the clean MISL-formulas of $\mathcal{L}_{\mathsf{clean}}$, since for an arbitrary MISL-formula, we can always recursively transform it into an equivalent clean formula via ($\text{Renaming}_{\mathsf{MISL}}$). Our method is motivated by [15]. We show that for any $\varphi \in \mathcal{L}_{\mathsf{clean}}$, φ is satisfiable if and only if there exist relations $R \subseteq \omega \times \omega$ and $E \subseteq \mathcal{L}_{\mathsf{clean}} \times \omega$ such that the equivalences in Lemma 6.10 hold and that $E(\varphi, m)$ for some $m \in \omega$. More specifically, we show that the satisfiability of φ is equivalent to the Σ^1_1-formula φ-SAT that is defined in the following:

$$\begin{aligned}
&\exists R \exists E\, \forall s \in \omega \\
&\qquad \neg E(\bot, s) \\
&\qquad \wedge\ \forall \chi \in \mathcal{L}_{\mathsf{clean}}\,(E(\neg\chi, s) \leftrightarrow \neg E(\chi, s)) \\
&\qquad \wedge\ \forall \chi_1, \chi_2 \in \mathcal{L}_{\mathsf{clean}}\,(E(\chi_1 \wedge \chi_2, s) \leftrightarrow E(\chi_1, s) \wedge E(\chi_2, s)) \\
&\qquad \wedge\ \forall \chi \in \mathcal{L}_{\mathsf{clean}}\,(E(\Box\chi, s) \leftrightarrow \forall t \in \omega\,(Rst \to E(\chi, t))) \\
&\qquad \wedge\ \forall \langle p := \psi\rangle\chi \in \mathcal{L}_{\mathsf{clean}}\,(E(\langle p := \psi\rangle\chi, s) \leftrightarrow E(\chi[\psi/p], s)), \\
&\qquad\quad \text{if the main connective of } \chi \text{ is not an iterative substitution operator} \\
&\qquad \wedge\ \forall \langle p := \psi_0; (p := \psi_1)^*\rangle\chi \in \mathcal{L}_{\mathsf{clean}} \\
&\qquad\qquad E(\langle p := \psi_0; (p := \psi_1)^*\rangle\chi, s) \leftrightarrow \exists n \in \mathbb{N}\, E(\langle p := \psi_0; (p := \psi_1)^n\rangle\chi, s) \\
&\qquad \wedge\ \exists m \in \omega\, E(\varphi, m)
\end{aligned} \tag{φ-SAT}$$

Note that the condition in the fifth conjunction can be checked by splitting this conjunction to several sub-cases basing on the main connective of χ. We assume the existence of an encoding of $\mathbf{P}$ with ω (e.g., p_i is encoded by i), and then the formulas $\neg\chi$, $\chi_1 \wedge \chi_2$, $\Box\chi$, $\langle p := \psi\rangle\chi$, $\chi[\psi/p]$ and $\langle p := \psi_0; (p := \psi_1)^*\rangle\chi$ involved in Σ^1_1-formula φ-SAT are actually abbreviations for the recursive functions respectively:

$$f_1 : \chi \mapsto \neg\chi \qquad f_2 : \chi_1, \chi_2 \mapsto \chi_1 \wedge \chi_2 \qquad f_3 : \chi \mapsto \Box\chi \qquad f_4 : p, \psi, \chi \mapsto \langle p := \psi\rangle\chi$$

$$f_5 : p, \psi, \chi \mapsto \chi[\psi/p] \qquad f_6 : p, \psi_0, \psi_1, \chi \mapsto \langle p := \psi_0; (p := \psi_1)^*\rangle\chi$$

Now we proceed to prove that the satisfiability of φ is equivalent to the Σ^1_1-formula φ-SAT.

First, assume that φ is satisfiable. Then, there is a countable model $\boldsymbol{M} = (\omega, R, V)$ such that $\boldsymbol{M}, m \vDash_{\mathsf{MISL}} \varphi$ for some $m \in \omega$. Let E be a relation such that $E(p, s)$ iff $s \in V(p)$, and for any $\chi \in \mathcal{L}_{\mathsf{clean}}$, $E(\chi, s)$ is defined recursively according to the equivalences in Lemma 6.10. More specifically, define E to be a relation satisfying the following:

$$\begin{aligned} E(\bot, s) &:= \bot \\ E(p, s) &\text{ iff } s \in V(p) \\ E(\neg\chi, s) &:= \neg E(\chi, s) \\ E(\chi_1 \wedge \chi_2, s) &:= E(\chi_1, s) \wedge E(\chi_2, s) \\ E(\Box\chi, s) &:= \forall t \in W\,(Rst \rightarrow E(\chi, t)) \\ E(\langle p := \psi\rangle\chi, s) &:= E(\chi[\psi/p], s) \\ E(\langle p := \psi_0; (p := \psi_1)^*\rangle\chi, s) &:= \exists n \in \mathbb{N}\, E(\langle p := \psi_0; (p := \psi_1)^n\rangle\chi, s) \end{aligned}$$

where the main connective of χ used in the clause for $E(\langle p := \psi\rangle\chi, s)$ is not an iterative substitution operator with p as its pivot. Note that E is well-defined since the recursion follows a descending chain of the well-founded order $\ll$. It can be proved by induction on formulas along relation $\ll$ that E agrees with $E_{\boldsymbol{M}}$ (the evaluation relation defined in Lemma 6.10), and hence $E(\varphi, m)$ holds. Therefore we can find R, E and m that make the Σ^1_1-formula φ-SAT true.

Moreover, for the other direction, when the Σ^1_1-formula φ-SAT is true for some R, E and m, we just need to set $\boldsymbol{M} = (\omega, R, V)$ where $V(p) = \{s : E(p, s) \textit{ holds}\}$. By induction on formulas along the relation $\ll$, we can show that $\boldsymbol{M}, m \vDash_{\mathsf{MISL}} \varphi$ is the case.

To sum up, the satisfiability problem of φ is in Σ^1_1. Therefore MISL is in Σ^1_1. □

With Theorems 6.8 and 6.11, we have Theorem 6.4 immediately. Let us end this part with the following:

The relation between MISL *and* IMR. We have already shown that the iterative modal relativization IMR can be translated to MISL (Theorem 6.8). Also, since both MISL and IMR are Σ^1_1-complete, there is a converse reduction in principle, which we leave for further study:

Open problem Find a sound and faithful translation from MISL to IMR.

6.3. *Undecidability of* MISL *on finite tree models*

Having seen that the satisfiability problem for MISL is highly undecidable, in this part we confine ourselves to the class of finite tree models and show that even on such a simple class of models, the logic is undecidable. To achieve this, we will make use of the *post correspondence problem* (PCP), whose details are as follows:

Definition 6.12 (Post correspondence problem)**.** *Let Σ be an alphabet. An* instance *of the post correspondence problem over alphabet Σ is a pair of sequences (U, V) such that $U = (u_1, u_2, \ldots, u_m) \in (\Sigma^*)^m$ for some $m \in \mathbb{N}$, $V = (v_1, v_2, \ldots, v_m) \in (\Sigma^*)^m$ and the pairs $(u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m)$ are distinct. The* post correspondence problem *is to find whether there is a finite sequence of pairs $(u_{i_1}, v_{i_1}), (u_{i_2}, v_{i_2}), \ldots, (u_{i_l}, v_{i_l})$ such that for any $1 \leqslant j \leqslant l$, $i_j \in \{1, \ldots, m\}$ and $u_{i_1}u_{i_2}\ldots u_{i_l} = v_{i_1}v_{i_2}\ldots v_{i_l}$, which is called a* solution *to the problem.*

We will work with the PCP over the alphabet $\Sigma = \{a, b\}$ with two elements, which is undecidable [35]. We prove the undecidability of MISL over finite tree models by showing that for any instance of PCP over Σ, there exists an MISL-formula φ such that φ is satisfied in a finite tree model if and only if there is a solution to the PCP.

Theorem 6.13. *For any instance (U, V) of the post correspondence problem over alphabet $\Sigma = \{a, b\}$, there is an* MISL*-formula $\varphi(U, V)$ such that*

$$\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \varphi(U, V) \textit{ for some finite tree model } \boldsymbol{M} \quad \textit{iff} \quad (U, V) \textit{ has a solution.}$$

Proof. Before going into the proof details, let us note that in the proof we will assume that models $\boldsymbol{M} = (W, R_c, R_d, V)$ contain two relations R_c and R_d, and corresponding to this, we will use two different modalities $\Diamond_c$ and $\Diamond$ in our language: $\Diamond_c$ moves only along R_c and $\Diamond$ moves only along R_d. With the help of propositional letters, the two modalities and the two relations can be encoded by our standard case with only one modality and one relation (cf. e.g., [30, 36]), but the usage of the two relations would make the proof easier to be understood. Let us now begin.

Suppose that $U = \{u_1, \ldots, u_m\}$ and $V = \{v_1, \ldots, v_m\}$. In the sequel we first construct the formula $\varphi(U, V)$ such that if $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \varphi(U, V)$ then a solution to the PCP instance (U, V) can be 'read' immediately from $(\boldsymbol{M}, w)$. Before showing the details of the construction, let us first explain the key ideas:

- The construction will involve three kinds of states, including the witness state, the configuration states, and the states on the branches associated to the configuration states.

- The state w denotes the '*witness state*' in $\boldsymbol{M}$, and its R_d-successors are '*configuration states*', whose intuition will be made clear in the item below. Also, we will use R_d^* for the *transitive closure of* R_d, and use $\Box^*$ for the corresponding modality that is incorporated from IMR: since we have proved that IMR can be translated into MISL (Theorem 6.8), we can use $\Box^*$ directly whenever needed.

- Along the relation R_c, configuration states can only reach configuration states, while via R_d, each configuration state is associated with two R_d-branches, a 'left' branch and a 'right' branch, which intuitively stand for a pair of strings in $(\Sigma^*)^* \times (\Sigma^*)^*$. Given a configuration state, the associated pair of branches is called a '*configuration*' of the configuration state .[10]

- The formula $\varphi(U, V)$ constructed will apply iterative substitutions to search for configuration states whose configurations are pairs of the form $(u_{i_1}u_{i_2}\dots u_{i_l}, v_{i_1}v_{i_2}\dots v_{i_l})$, until the configuration of a configuration state is found to be the solution. In what follows, the configuration states whose configurations are of this form will also be called '*candidate states*', and corresponding to this, those configurations will also be called '*candidates*' (of the solution of PCP).[11]

We first define the configuration states, and each of them is associated with a left R_d-branch and a right R_d-branch. Each branch corresponds to strings $s_1s_2\dots s_l$ with $l \in \mathbb{N}$ and $s_i \in \Sigma^*$. Every node on the left branch is tagged by the proposition left, one of the propositional letters a and b, and one of the propositional letters odd and even for parities of natural numbers. Similarly for the nodes on the right branch, except that each node on the right branch is tagged by the proposition right. In this way, any path residing in a branch can be viewed as a string by reading the propositions a, b along the relation R_d. The parity propositions odd and even denote the separation of strings, which helps to read the strings encoded by a branch. For instance, when the first half s_1 of a branch consists of some odd-states and the second half s_2 consists of some even-states, we read the branch as s_1s_2.

Let $\oplus$ denote XOR (i.e., *exclusive or*). Let left, right, odd, even, a and b be propositions that will function as mentioned. Define the formula Configuration characterizing the configuration states as follows:

$$\begin{aligned}
\mathsf{Configuration} :=& (\neg\mathsf{odd} \wedge \neg\mathsf{even} \wedge \neg\mathsf{left} \wedge \neg\mathsf{right}) \wedge && (1)\\
&\big(\langle p := \Box\bot \wedge \mathsf{left} \wedge (\mathsf{odd} \oplus \mathsf{even}) \wedge (a \oplus b); (p := \mathsf{left} \wedge (\mathsf{odd} \oplus \mathsf{even}) \wedge (a \oplus b) \wedge \Diamond p)^*; && (2)\\
&\quad q := \Box\bot \wedge \mathsf{right} \wedge (\mathsf{odd} \oplus \mathsf{even}) \wedge (a \oplus b); (q := \mathsf{right} \wedge (\mathsf{odd} \oplus \mathsf{even}) \wedge (a \oplus b) \wedge \Diamond q)^*\rangle\\
&\quad (\Diamond\top \to (\Diamond p \wedge \Diamond q \wedge \Box(p \vee q)))\big) \wedge\\
&\bigwedge_{X\in\{\mathsf{left},\mathsf{right}\},\, Y\in\{\mathsf{odd},\mathsf{even}\},\, s\in\{a,b\}} [q := X \wedge Y \wedge s; (q := \Diamond q)^*](\Diamond q \to \Box(X \to q)) && (3)
\end{aligned}$$

The conjunct (2) says that each branch associated to a configuration state is either a left-branch or a right-branch, on which each node is tagged with exactly one of $\{\mathsf{odd}, \mathsf{even}\}$ and with exactly one of $\{a, b\}$. The last conjunct (3) says that all left-branches of the configuration state represent the same sequence of strings, and the same for right branches. Note that for any $n \in \mathbb{N}$, after the substitution $q := X \wedge Y \wedge s; (q := \Diamond q)^n$, the truth set of q consists of states that can reach a state tagged by X, Y and s via an R_d-path of length n. Moreover, by construction, it might be the case that a configuration state has the empty configuration, and we use start for this situation, i.e.,

$$\mathsf{start} := \neg\mathsf{odd} \wedge \neg\mathsf{even} \wedge \neg\mathsf{left} \wedge \neg\mathsf{right} \wedge \Box\bot,$$

which implies Configuration.

[10]It is important to keep in mind that a configuration state is different from a configuration: a configuration state is a state that leads to a pair of branches, and such a pair is a configuration.

[11]In a candidate $(u_{i_1}u_{i_2}\dots u_{i_l}, v_{i_1}v_{i_2}\dots v_{i_l})$, both the '$u$'-part and the '$v$'-part contain l strings, but a configuration does not have to be so.

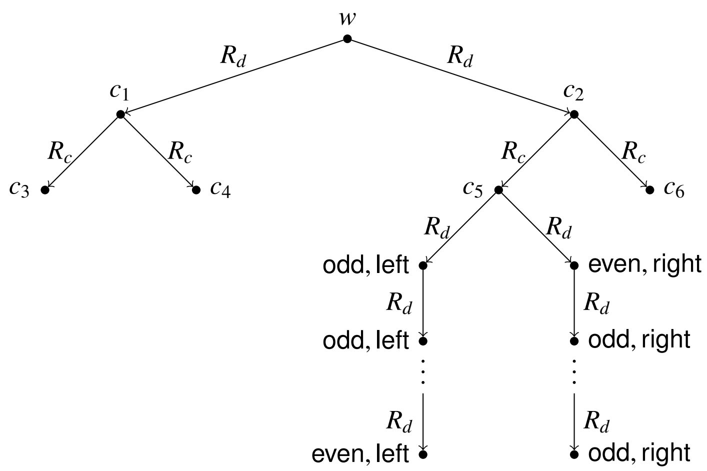

Figure 7. The state w is a witness state, and along R_d, it can only reach configuration states c_1 and c_2; also, all states c_3-c_6 that can be reached from c_1 and c_2 along R_c are configuration states. For each $1 \leqslant i \leqslant 6$, the configuration state c_i is associated with a pair of R_d-branches that give the configuration of c_i, and in the picture, only the configuration of c_5 is depicted, and those for other configuration states are omitted. As we can see, each state on the left branch is labelled with left, and each one on the right is labelled with right. However, the information for a and b is omitted.

The witness state only reaches configuration states and the states on the configurations associated to those configuration states in finitely many steps along R_d, and those configuration states only reach configuration states via finitely many steps along R_c. The formula Witness that characterizes the witness state is defined as follows:

$$\mathsf{Witness} := \Diamond\top \wedge \Box(\mathsf{Configuration} \wedge \Box_c \mathsf{Configuration})$$

A form of the witness state is depicted in Figure 7.

Now we will construct formulas that are useful in defining candidate states, a special kind of configuration states. For $X \in \{\mathsf{left}, \mathsf{right}\}$, $Y \in \{\mathsf{odd}, \mathsf{even}\}$, and a non-empty $s \in \Sigma^*$, we define $\mathsf{String}(X, Y, s)$ recursively as follows:

For $s \in \{a, b\}$, $\mathsf{String}(X, Y, s) := X \wedge Y \wedge s \wedge \Box\bot$.

For $k \geqslant 2$, $s^i \in \{a, b\}$, $\mathsf{String}(X, Y, s^1 s^2 \dots s^k) := X \wedge Y \wedge s^1 \wedge \Diamond\top \wedge \Box\mathsf{String}(X, Y, s^2 \dots s^k)$.

So, $\mathsf{String}(X, Y, s)$ is true at states u such that all states on an 's-path' (which is an R_d-path) starting from u and ending with a dead end are tagged with X and Y.

Next, let us proceed to define candidate states. As mentioned, we call a pair of strings of form

$$(u_{i_1} u_{i_2} \dots u_{i_l}, v_{i_1} v_{i_2} \dots v_{i_l})$$

a candidate (of the solution). For any $l \in \mathbb{N}$, we say that a candidate is of order l if it is a pair of strings of form $(u_{i_1} u_{i_2} \dots u_{i_l}, v_{i_1} v_{i_2} \dots v_{i_l})$. Meanwhile we say that a candidate state is of order l if its candidate is of order l.

Now, we are going to recursively construct a class $\{\mathsf{cand}^l : l \in \mathbb{N}\}$ of formulas: intuitively, for each $l \in \mathbb{N}$, cand^l represents a set of candidate states of order l. It is useful to keep in mind that our construction below will ensure that the configuration states in cand^l for each l are all candidate states that have the same candidate $(u_{i_1} u_{i_2} \dots u_{i_l}, v_{i_1} v_{i_2} \dots v_{i_l})$ for some sequence of $i_1 i_2 \dots i_l$. For the basic case $l = 0$, we define that:

$$\mathsf{cand}^0 := \mathsf{start}$$

referring to the candidate states with the empty string as their candidates. Now, for any $l \in \mathbb{N}$, we present a method to construct cand^{l+1} from cand^l.

Suppose for now that the candidates of all the candidate states in cand^l are the same, e.g., the pair of sequences of strings (U', V').[12] Intuitively, we are going to make the elements in cand^{l+1} be candidate states with the candidate

[12] This property is not yet true but will be stated in the formula $\mathsf{Append}(i, \mathsf{cand}^l)$, i.e., if $\mathsf{Append}(i, \mathsf{cand}^l)$ is satisfied on any configuration state α, it must be the case that the candidates of all candidate states in cand^l are the same, and this candidate pair of strings is a prefix of the pair of string encoded by α.

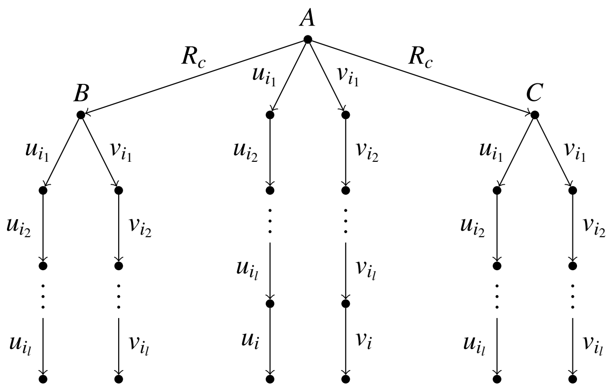

Figure 8. In the picture, links labeled with R_c are edges in R_c, links without any relation-label are edges of R_d, and the labels u_{i_j} and v_{i_k} of the R_d-edges mean that the states on the edges give us strings u_{i_j} and v_{i_k}. The state A is a candidate state satisfying $\mathsf{Append}(i, \mathsf{cand}^l)$, and it has an R_c-successor B that is a candidate state in cand^l, in which all candidate states have the same candidate. Also, A may have other R_c-successors C that are configuration states not in cand^l, but those Cs have the candidate of B as their candidate. Moreover, for each pair of branches depicted, states on the left branch satisfy the property left, and those on the right branch satisfy the property right.

extending (U', V') with (u_i, v_i) for a fixed $i \in \{1, \dots, m\}$.[13] This is characterized by the formula $\mathsf{Append}(i, \mathsf{cand}^l)$:

$$\begin{aligned}
\mathsf{Append}(i, \mathsf{cand}^l) := {}& \Diamond_c \mathsf{cand}^l \wedge \\
& (\langle q := \mathsf{even} \wedge \Box\bot; (q := \Diamond q)^* \rangle \Diamond_c q \to (\Diamond^* \mathsf{String}(\mathsf{left}, \mathsf{odd}, u_i) \wedge \Diamond^* \mathsf{String}(\mathsf{right}, \mathsf{odd}, v_i))) \wedge \\
& (\langle q := \mathsf{odd} \wedge \Box\bot; (q := \Diamond q)^* \rangle \Diamond_c q \to (\Diamond^* \mathsf{String}(\mathsf{left}, \mathsf{even}, u_i) \wedge \Diamond^* \mathsf{String}(\mathsf{right}, \mathsf{even}, v_i))) \wedge \\
& \bigwedge_{\substack{X \in \{\mathsf{left}, \mathsf{right}\} \\ Y \in \{\mathsf{odd}, \mathsf{even}\} \\ s \in \{a, b\}}} [q := X \wedge Y \wedge s; (q := \Diamond q)^*] (\Diamond_c q \to q) \wedge \\
& \bigwedge_{\substack{X \in \{\mathsf{left}, \mathsf{right}\} \\ Y, Y' \in \{\mathsf{odd}, \mathsf{even}\} \\ s \in \{a, b\}}} [q := X \wedge Y' \wedge s \wedge \Diamond \mathsf{String}(X, Y, w_i^X); (q := \Diamond q)^*] (q \to \Box_c q)
\end{aligned}$$

where $w_i^X := \begin{cases} u_i, & \text{if } X \text{ is left}, \\ v_i, & \text{if } X \text{ is right}. \end{cases}$

The first conjunct in $\mathsf{Append}(i, \mathsf{cand}^l)$ expresses that the current state has at least an R_c-successor that is a candidate state in cand^l. By the second and the third conjuncts, the configuration of the current state has the form $(U_1 u_i, V_1 v_i)$ ending with (u_i, v_i) and tagged with the proper parity (the parity of $l + 1$). Since the current state can reach a candidate state in cand^l via R_c and we have assumed that the candidate states in cand^l have the same candidate (U', V'), the fourth conjunct ensures that $(U_1 u_i, V_1 v_i)$ has (U', V') as its prefix. But by the second and the third conjuncts, the parity of (u_i, v_i) is different from the final segment of (U', V'), and so it must be the case that (U_1, V_1) has (U', V') as its prefix. By the fifth conjunct, (U_1, V_1) is a prefix of (U', V'), and combining the fourth conjunct, we have $(U_1, V_1) = (U', V')$. Putting the five conjuncts together, we know that the configuration of the current state is obtained by appending (u_i, v_i) to (U', V'). A form of a candidate state satisfying $\mathsf{Append}(i, \mathsf{cand}^l)$ is described in Figure 8.

Now we can define a set of candidate states of order $(l + 1)$ to be the truth set of cand^{l+1}:

$$\mathsf{cand}^{l+1} := \bigvee_{1 \leqslant i \leqslant m} \mathsf{Append}(i, \mathsf{cand}^l)$$

[13]Notice that we have assumed that the instance (U, V) of the post correspondence problem we are working on is given by $U = \{u_1, \dots, u_m\}$ and $V = \{v_1, \dots, v_m\}$, which give us the natural number m.

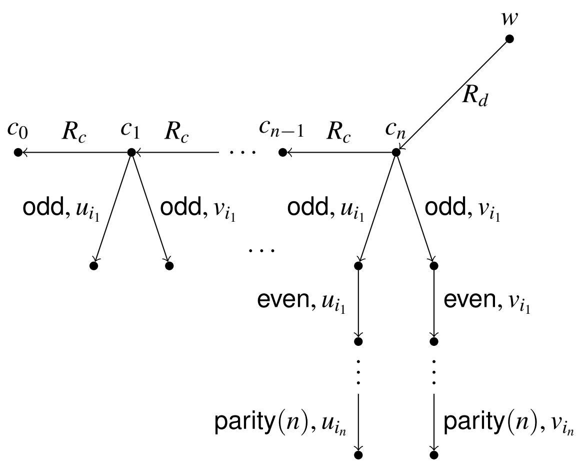

Figure 9. A model satisfying $\varphi(U, V)$, given that $(u_{i_1}u_{i_2}\dots u_{i_l}, v_{i_1}v_{i_2}\dots v_{i_l})$ is a solution. In the model, links with the label R_d are edges of R_d; links with the label R_c are edges of R_c; and links without any relation-label are edges in the transitive closure R_d^* of R_d. The labels even and odd for links mean that the states on the edges of R_d^* are even and odd respectively, and parity(n) gives us the parity of the number n. Also, labels u_{i_j} and u_{v_k} for links mean that the states on the edges of R_d^* give us the strings u_{i_j} and u_{v_k} respectively. In the picture, all $c_0, \dots, c_n$ are configurations states, but we only draw the configurations for states c_0, c_1 and c_n (notice that c_0 has the empty string as its configuration), and the configurations of other configuration states are omitted.

Note that cand^{l+1} only accounts for the candidate states of order $(l + 1)$ that have those candidate states in cand^l as their R_c-successors, but not for arbitrary candidate states of order $(l + 1)$.

In the end we need to define when a configuration state can be treated as a solution to the PCP instance. We justify the acceptance of a configuration state by testing whether its 'left-string' and 'right-string' are identical, using the Accept formula as follows:

$$\mathsf{Accept} := \Diamond\top \wedge ([q := a; (q := \Diamond q)^*](\Diamond q \to \Box q)) \wedge ([q := b; (q := \Diamond q)^*](\Diamond q \to \Box q))$$

The second and the third conjuncts are analogous, and let us explain the second one. It intuitively states that for a given candidate state, if it can reach a in some fixed n steps for an arbitrary number n, then it must reach a in n steps on both the left-branch and the right-branch. So, the formula ensures that the left-branch and the right-branch of a candidate state are the same. Finally, define the formula $\varphi(U, V)$ as follows:

$$\varphi(U, V) := \mathsf{Witness} \wedge \langle \mathsf{cand} := \mathsf{cand}^0; (\mathsf{cand} := \bigvee_{1 \leqslant i \leqslant m} \mathsf{Append}(i, \mathsf{cand}))^* \rangle \Diamond(\mathsf{cand} \wedge \mathsf{Accept})$$

If $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \varphi(U, V)$, then w is the witness state that can reach some candidate states via R_d. Also, recall that cand^0 means the set of candidate states having the empty string as their candidate, and cand^l is the truth set of the variable cand obtained by the l-th iterative substitution. For all $l \in \mathbb{N}$, as explained for the construction of $\mathsf{Append}(i, \mathsf{cand}^l)$, the truth set of cand^l is a subset of the set of all candidate states of order l. Therefore, when $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \varphi(U, V)$, w has an R_d-successor that is a candidate state of suitable order corresponding to the solution of the PCP instance (U, V).

On the other hand, assume that (U, V) has a solution $(u_{i_1}u_{i_2}\dots u_{i_l}, v_{i_1}v_{i_2}\dots v_{i_l})$. Consider the model $\boldsymbol{M}$ depicted in Figure 9. In the model, the witness state is w, which has an R_d-successor c_n that is a configuration state, and there is an R_c-sequence $(c_n, \dots, c_0)$ starting from c_n (and so all these states $c_0, \dots, c_n$ are configuration states). According to the construction of $\varphi(U, V)$, it can be verified that the truth set of cand^i is $\{c_i\}$ for $0 \leqslant i \leqslant l$, and that $\boldsymbol{M}, c_l \vDash_{\mathsf{MISL}} \mathsf{Accept}$. Therefore $\boldsymbol{M}, w \vDash_{\mathsf{MISL}} \varphi(U, V)$.

To sum up, given a PCP instance (U, V), $\varphi(U, V)$ is satisfiable in a finite tree model if and only if (U, V) has a solution. Therefore, we conclude that the satisfiability problem for MISL w.r.t. finite tree models is undecidable. □

7. Conclusion

Summary Motivated by the ubiquitous applications of substitutions, we develop logical frameworks containing single-step substitutions and iterative substitutions as modal operators that update valuation of propositional variables.

Our starting point is a simple setting MSL with only single-step substitution operators. For the logic, we provide a complete proof system and show its decidability. Also, we clarify the differences between those operators with ordinary syntactic replacements.

Then, for the more intricate proposal MISL containing iterative substitution operators, we develop various validities to show how those operators work and analyze its applications to crucial notions in games. Besides, we compare MISL with many other important logics that share a similar iterative feature, including the modal μ-calculus, the infinitary modal logic ML^{∞}, the propositional dynamic logic PDL and the iterative modal relativization IMR, which also illustrates advantages of our logic. Moreover, we explore suitable criteria to measure the expressiveness of the logics and study the computational behavior of both MISL and its restriction to the class of finite tree models.

Further directions Let us end by a few directions that are worth pursuing in future. Several open problems have been identified along the way, including relations between MISL and other relevant logics. In addition to the logic frameworks mentioned here, it is crucial to recognize that there are many other important logics that have the iterative reasoning concepts, including modal logics with inflationary fixed-point MIC [37] and the modal logic of oscillations [3]. Some of them can also be translated into MISL or its further extensions,[14] but the precise connections remain to be determined. Also, the explorations in the article have a model-theoretic approach, and it is important to study the logical proposals from a proof-theoretic perspective as well, e.g., sequent calculi. Moreover, it is meaningful to allow ordinal sequences of iteration, and based on the simultaneous one-step substitutions [20, 21, 22], one can also study the iterative generalization like $\langle (p := \psi; q = \chi)^* \rangle\varphi$, to make the logics applicable in broader scenarios. Finally, as suggested by the article, the logics for substitutions have promising applications in various fields, including games [38, 31], social networks [2, 1], and knowledge update [12, 11, 14]. In the next step, we will investigate how the abstract substitution mechanism can be instantiated in these concrete settings, with particular attention to its combination with the logics for the hide and seek game [30, 36, 39, 40] and broader logics for graph games [41, 42, 43, 32] which serve as an important motivation for this article.

Acknowledgments We acknowledge Johan van Benthem for insightful discussions and valuable comments, which were indispensable to the completion of the paper. We thank Alexandru Baltag and Yifeng Ding for helpful comments, and Yanjing Wang for drawing our attention to work on the dynamic logic of propositional assignment. Also, we wish to thank the two anonymous reviewers for their very useful comments that helped improve the paper. Fenrong Liu is supported by Tsinghua University's Initiative for Advancing First-Class and World-Leading Disciplines in the Humanities and Social Sciences. Sujata Ghosh acknowledges Department of Science and Technology, Government of India for financial support vide Reference No DST/CSRI/2018/202 under Cognitive Science Research Initiative (CSRI) to carry out this work.

References

[1] F. Liu, J. Seligman, P. Girard, Logical dynamics of belief change in the community, Synthese 191 (11) (2014) 2403–2431.
[2] A. Baltag, Z. Christoff, R. K. Rendsvig, S. Smets, Dynamic epistemic logics of diffusion and prediction in social networks, Studia Logica 107 (2019) 489–531.
[3] J. van Benthem, Oscillations, logic, and dynamical systems, in: S. Ghosh, J. Szymanik (Eds.), The Facts Matter, College Publications, 2015, p. 9–22.
[4] D. Gale, F. M. Stewart, Infinite games with perfect information, in: H. W. Kuhn, A. W. Tucker (Eds.), Contributions to the Theory of Games, Volume II, Vol. 28 of Annals of Mathematics Studies, Princeton University Press, 1953, pp. 245–266.
[5] J. van Benthem, An abstract look at the fixed-point theorem for provability logic, in: N. Bezhanishvili, R. Iemhoff, F. Yang (Eds.), Dick de Jongh on Intuitionistic and Provability Logics, Vol. 28 of Outstanding Contributions to Logic, Springer, 2024, pp. 75–88.

[14] For instance, the framework 1MIC (i.e., MIC without simultaneous inductions) can be embedded into MISL (when only finite sequences of approximation are allowed) and the oscillation operators can be defined by the extension of MISL with the universal modality [26], but we leave the details to other occasions.

[6] W. van der Hoek, M. Wooldridge, On the logic of cooperation and propositional control, Artificial Intelligence 164 (2005) 81–119.
[7] D. Thompson, Local fact change logic, in: F. Liu, H. Ono, J. Yu (Eds.), Knowledge, Proof and Dynamics, Logic in Asia: Studia Logica Library, Springer, 2020, pp. 73–96.
[8] P. Balbiani, A. Herzig, N. Troquard, Dynamic logic of propositional assignments: A well-behaved variant of pdl, in: O. Kupferman (Ed.), Logic in Computer Sciences (LICS), IEEE, 2013, pp. 143–152.
[9] W. Tang, Substitution as Modality, Bachelor's thesis (in Chinese), Peking University (2022).
[10] W. Tang, Y. Wang, Substitution as modality, in: Advances in Modal Logic. Booklet of Short Papers, 2024, pp. 57–63.
[11] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic Epistemic Logic, Springer, 2007.
[12] J. van Benthem, Logical Dynamics of Information and Interaction, Cambridge University Press, 2011.
[13] J. A. Plaza, Logics of public communications, in: M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, Z. W. Ras (Eds.), Proceedings of the 4th international symposium on methodologies for intelligent systems, Oak Ridge National Laboratory, 1989, pp. 201–216.
[14] A. Baltag, L. S. Moss, S. Solecki, The logic of public announcements and common knowledge and private suspicions, in: I. Gilboa (Ed.), Proceedings of TARK '98, 1998, pp. 43–56.
[15] J. S. Miller, L. S. Moss, The undecidability of iterated modal relativization, Studia Logica 79 (2005) 373–407.
[16] B. Kooi, Dynamic term-modal logic, in: J. van Benthem, S. Ju, F. Veltman (Eds.), Proceedings of LORI 2007, College Publications, 2007, pp. 173–185.
[17] J. van Eijck, Guarded actions, Technical report, CWI, Amsterdam (2004).
[18] J. van Benthem, J. van Eijck, B. Kooi, Logics of communication and change, Information and Computation 204 (2006) 1620–1662.
[19] H. van Ditmarsch, B. Kooi, Semantic results for ontic and epistemic change, in: G. Bonanno, W. van der Hoek, M. Wooldridge (Eds.), Logic and the foundations of game and decision theory (LOFT 7), Amsterdam University Press, 2008, p. 87–117.
[20] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic epistemic logic with assignment, in: F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, M. Wooldridge (Eds.), Proceedings of AAMAS 2005, ACM, 2005, p. 141–148.
[21] H. van Ditmarsch, A. Herzig, T. de Lima, Public announcements, public assignments and the complexity of their logic, Journal of Applied Non-Classical Logics 22 (2012) 249–273.
[22] B. Kooi, Expressivity and completeness for public update logics via reduction axioms, Journal of Applied Non-Classical Logics 17 (2007) 231–253.
[23] D. Kozen, R. Parikh, A decision procedure for the propositional μ-calculus, in: E. Clarke, D. Kozen (Eds.), Proceedings of Workshop on Logics of Programs 1983, Vol. 164 of Lecture Notes in Computer Science, Springer, 1984, pp. 313–325.
[24] Y. Venema, Lectures on the Modal μ-calculus, Lecture Notes, ILLC, University of Amsterdam (2020).
[25] S. Radev, Infinitary propositional normal modal logic, Studia Logica 46 (1987) 291–309.
[26] J. van Benthem, Modal Logic for Open Minds, CSLI Publications, 2010.
[27] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, 2001.
[28] Y. Wang, Q. Cao, On axiomatizations of public announcement logic, Synthese 190 (2013) 103–134.
[29] D. Harel, D. Kozen, J. Tiuryn, Dynamic Logic, The MIT Press, 2000.
[30] D. Li, S. Ghosh, F. Liu, Y. Tu, A simple logic of the hide and seek game, Studia Logica 111 (2023) 821–853.
[31] J. van Benthem, Logic in Games, The MIT Press, 2014.
[32] J. van Benthem, F. Liu (Eds.), Graph Games and Logic Design: Recent Developments and Future Directions, Springer, 2026.
[33] D. Janin, I. Walukiewicz, On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic, in: U. Montanari, V. Sassone (Eds.), CONCUR '96: Concurrency Theory, Vol. 1119 of Lecture Notes in Computer Science, Springer, 1996, p. 263–277.
[34] J. R. Büchi, Weak second-order arithmetic and finite automata, Mathematical Logic Quarterly 6 (1960) 66–92.
[35] J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson, 2006.
[36] Q. Chen, D. Li, Logic of the hide and seek game: characterization, axiomatization, decidability, in: N. Gierasimczuk, F. R. Velázquez-Quesada (Eds.), Proceedings of DaLÍ 2023, Vol. 14401 of LNCS, Springer, 2024, pp. 20–34.
[37] A. Dawar, E. Grädel, S. Kreutzer, Inflationary fixed points in modal logic, ACM Transactions on Computational Logic 5 (2004) 282–315.
[38] M. Osborne, A. Rubinstein, A Course in Game Theory, The MIT Press, Cambridge, 1994.
[39] D. Li, S. Ghosh, F. Liu, Reasoning under uncertainty in the game of cops and robbers, Synthese 206 (2025) 1–35.
[40] K. Sano, F. Liu, D. Li, Hybrid logic of the hide and seek game, Studia Logica doi:10.1007/s11225-024-10149-7.
[41] D. Li, Formal threads in the social fabric: Studies in the logical dynamics of multi-agent interaction, Ph.D. thesis, Department of Philosophy, Tsinghua University and ILLC, University of Amsterdam (2021).
[42] G. Aucher, J. van Benthem, D. Grossi, Modal logics of sabotage revisited, Journal of Logic and Computation 28 (2) (2018) 269–303.
[43] J. van Benthem, F. Liu, Graph games and logic design, in: F. Liu, H. Ono, J. Yu (Eds.), Knowledge, Proof and Dynamics, Springer, Singapore, 2020, p. 125–146.