

Modelling strategies in hide and seek game: A substitution approach

Yaxin Tu¹, Sujata Ghosh², Fenrong Liu^{3,4}, and Dazhu Li^{5,6}

¹ Department of Computer Science, Princeton University, Princeton, USA

² Indian Statistical Institute, Chennai, India

³ The Tsinghua-UvA JRC for Logic, Department of Philosophy, Tsinghua University, Beijing, China

⁴ Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands

⁵ Institute of Philosophy, Chinese Academy of Sciences, Beijing, China

⁶ Department of Philosophy, University of Chinese Academy of Sciences, Beijing, China

Abstract. In this chapter we discuss a modal logic of substitutions which provide us a way to describe winning strategies in hide and seek game played over any game graph. To this end, a modal logic of iterated substitutions is proposed, and a number of properties of this logic are explored including their expressive strength and satisfiability problems. The proposed framework is also compared with the existing ones involving iterative reasoning, which in turn showcases certain advantages of the proposal.

Keywords: Hide and seek, Modal substitution logic, Iterative substitution, Axiomatization, Fixed-point logic, Expressiveness, Undecidability

1 Introduction

In [16], we have proposed a modal logic for the hide and seek game that describes various aspects of reasoning in such games. We have studied various properties of the logic that can characterize winning conditions as well as certain winning strategies in the game. In particular, one-step winning conditions can be expressed by the said logic whereas general winning positions cannot be expressed in this logic (cf. see [16]). Towards getting such expressive power, we explore a logical framework that deals with ‘substitutions’ [22], in particular, ‘iterated substitutions’, first introduced in [5]. To set the groundwork for this discussion, let us first clarify our approach to substitutions.

There can be different kinds of substitutions. For an 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, especially when we need to encode dynamic information with fixed formulas.⁷ In addition, we incorporate a mechanism for iterative substitutions, which helps describe iterative updates of dynamic information in various scenarios. To illustrate this with regard to the hide and seek game, let us now present a concrete example.

⁷ In what follows, to distinguish between these two approaches, we will often call the syntactic substitutions ‘replacements’ and the semantic ones ‘substitutions’.

Reasoning about the hide and seek game Consider a 2-player hide and seek game where the Hider moves to keep away from the Seeker while the Seeker moves to reach the Hider’s position. Let us work in the scenario where the game is of perfect information and the players’ positions and admissible moves are defined by a finite directed graph $\mathcal{B} = (W, R)$, such that when a player is at position $s \in W$, the player can move to a position in the neighbourhood of s in the next round. A concrete example for such a graph for hide and seek game is depicted in Figure 1.

The logic of hide and seek game (LHS, [16]) introduced in the chapter *A Modal Logic for the Hide and Seek Game* can describe the actions and reasoning of both players when they are at a specified pair of positions. LHS uses modalities $[H]$ and $[S]$ for Hider and Seeker’s admissible moves respectively, and uses an atom \mathcal{I} to denote that Hider and Seeker are at the same node. For example, in Figure 1 when Hider is at node c and Seeker is at node b , the formula $[H]\langle S \rangle \mathcal{I}$ holds, which indicates that whichever position Hider chooses to move to, Seeker can catch the Hider in the next move.

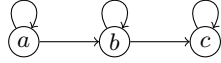


Fig. 1. A toy example for the hide and seek game \mathcal{B}_1

More generally, on a finite graph for hide and seek game $\mathcal{B} = (W, R)$, we can use backward induction reasoning to determine *winning positions* for Hider and Seeker: a pair of nodes (s, t) for Hider and Seeker, respectively, is a winning position for Seeker whenever Seeker can catch Hider within finite rounds, whatever strategy Hider uses, starting from the position (s, t) . In other words, a pair of nodes (s, t) is a winning position for Hider if Hider has a strategy to prevent getting caught by the Seeker in any finite rounds.⁸ We follow the convention that Hider moves first from node s and then the Seeker moves from node t , and the game continues with the players moving alternately. Since the winning positions of Hider and Seeker are complementary, in the sequel we focus on the winning position of Seeker, which can be captured by the following expression, which evidently does not have an equivalent formula in LHS:

$$(s, t) \text{ is a winning position for Seeker} \Leftrightarrow \exists \text{ natural number } n, ([H]\langle S \rangle)^n \mathcal{I} \text{ holds at } (s, t)^9$$

The process of finding the truth set of $\exists n([H]\langle S \rangle)^n \mathcal{I}$ (i.e. the set of winning positions for Seeker) is equivalent to conducting backward induction, which bears a natural *iterative* substitution core as follows:

$$\text{Wins}_S := \mathcal{I}; (\text{Wins}_S := [H]\langle S \rangle \text{Wins}_S)^*$$

Here, Wins_S denotes the set of winning positions of the Seeker. One can read the formula as follows: initially, Wins_S is the immediate winning positions of Seeker where Hider and Seeker start at the same node, and then the Wins_S -states (of the next stage) are iteratively re-defined as the states with the property $[H]\langle S \rangle \text{Wins}_S$ for finitely many times. For illustration, in Figure 1 for hide and seek game,

⁸ By the *Gale-Stewart Theorem* [9], a pair of nodes (s, t) is either a winning position for Hider or Seeker. So, the complement of the set of winning positions of Seeker is the set of winning positions of Hider.

⁹ In fact, n can be bounded above by $2^{|W|^2}$ ($|W|$ denotes the *cardinality* of the set of nodes W), since after $2^{|W|^2}$ iterations the truth set of $([H]\langle S \rangle)^n \mathcal{I}$ will repeat.

Win₅ evolves in the iterative substitutions as follows:

$$\begin{aligned} & \{(a, a), (b, b), (c, c)\} \\ & \{(a, a), (b, b), (c, c), (b, c)\} \\ & \{(a, a), (b, b), (c, c), (b, c), (a, b), (a, c)\} \end{aligned}$$

where the set stabilizes after the third iteration. The sequence of sets found in this process is exactly the sequence of sets found by backward induction. We defer the precise proof of this connection to Section 2.3.¹⁰

To understand the properties of the iterative substitution cores, we are going to explore logical frameworks with iterative substitutions, which will be integrated into standard modal logic as additional modalities. Beyond the widespread application of substitutions, as illustrated in the example, they also play a crucial role in numerous other logical systems. The remainder of this section will discuss related work.

Related work We start by mentioning some lines of research that are closely relevant to the notion of one-step substitutions, which is the base for iterative substitutions. *Local fact change logic* [11,21] extends the standard modal logic with an update operator that works somewhat differently in comparison to the substitution operator that we propose here.¹¹ The modal operators describing *syntactic substitutions* are studied in [20], in contrast to our semantic framework. Another important tradition is *dynamic-epistemic logic* DEL [8,3] that deals with changing models with various mechanisms [18,1,13,14].

As mentioned earlier, the study on modal logic with iterative substitutions was initiated in [5], where the logic is referred to as *Modal Substitution Logic*.¹² In addition, there are quite a few milestone contributions on iterative reasoning that have close connections to our proposed framework, including the *modal μ -calculus* [15,23], *the infinitary modal logic* [19], the *propositional dynamic logic* [2] and *iterative modal relativization* [17]. As mentioned above, 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.

Outline Our main goal is to explore the framework of iterative substitutions, formalized by *Modal Iterative Substitution Logic* MISL. The organization of our study is as follows: Section 2 lays out the basics of MISL, whose language extends the standard modal language with single-step substitution operators $\langle p := \psi \rangle$ and iterative substitution operators $\langle (p := \psi)^* \rangle$. Section 2.1 discuss various validities. Section 2.3 analyzes the applications of the resulting logic to our motivating example and other relevant game-theoretic notions in the literature. Next, in Section 3 we systematically investigate various properties of MISL, involving its expressive power and computational behavior. Section 3.2 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 3.3 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

¹⁰ One may observe that the backward induction described can also be formalized by the modal μ -calculus (see Section 2.3), and we will show that our proposal with the substitution operators are capable to reason about more intrinsic details with respect to the game (Section 4).

¹¹ The updates in this logic are implicit rather than explicitly described as in our case. 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.

¹² We will refer to this logic as *Modal Iterative Substitution Logic* here, to differentiate between the frameworks with and without iterative substitutions. See [22] for more details.

models). 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 4, which also demonstrates the merits of our proposal. Finally, Section 5 concludes the article with several directions that deserve to be explored in the future.

2 A modal logic for iterative substitutions (MISL)

Let us now follow [22] and present a formal proposal to explore the nature of substitutions. In this part, we define Modal Iterative Substitution Logic (MISL) with both simple substitution operators and iterative substitution operators. This section will present some validities to show how the iterative operators work and formally show that the backward induction reasoning introduced in Section 1 can be characterized by MISL. Let us first define the language of MISL, which extends the language \mathcal{L}_{ML} of the standard modal logic ML in the following manner:

Definition 1 (Language $\mathcal{L}_{\text{MISL}}$). *Let \mathbf{P} be a countable set of propositional letters. The language $\mathcal{L}_{\text{MISL}}$ of MISL is given by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \diamond\varphi \mid \langle p := \varphi \rangle\varphi \mid \langle (p := \varphi)^* \rangle\varphi$$

where $p \in \mathbf{P}$. Abbreviations \top , \perp , \vee and \rightarrow are as usual, so are the box operators \square , $[p := \varphi]$, and $[(p := \varphi)^*]$ for the De Morgan duals of \diamond , $\langle p := \varphi \rangle$, and $\langle (p := \varphi)^* \rangle$ respectively.

The readings of basic modal formulas are as usual. 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 formula $\langle (p := \psi)^* \rangle\varphi$ states 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*. 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. To distinguish different substitution operators, those $\langle p := \psi \rangle$ are termed as *simple substitution operators*. However, both $\langle p := \psi \rangle$ and $\langle (p := \psi)^* \rangle$ are called *substitution operators*. 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. For a formula $\langle (p := \psi)^* \rangle\varphi$, p is the *pivot* of the iterative substitution operator, while the *scope* of operator $\langle (p := \psi)^* \rangle$ is the whole formula $\langle (p := \psi)^* \rangle\varphi$. Also, we sometimes write $\langle \text{substitution}_1; \text{substitution}_2 \rangle\varphi$ for $\langle \text{substitution}_1 \rangle \langle \text{substitution}_2 \rangle\varphi$, where $\text{substitution}_i (i = 1, 2)$ represents a simple or an iterative substitution of form $p := \psi$ or $(p := \psi)^*$. Similarly, we write $[\text{substitution}_1; \text{substitution}_2]\varphi$ for $[\text{substitution}_1][\text{substitution}_2]\varphi$. The notion of *subformulas* for MISL, written as $\text{Sub}(\varphi)$, extends that for the standard modal logic with the following:

$$\begin{aligned} \text{Sub}(\langle p := \psi \rangle\varphi) &:= \text{Sub}(\psi) \cup \text{Sub}(\varphi) \cup \{\langle p := \psi \rangle\varphi\} \\ \text{Sub}(\langle (p := \psi)^* \rangle\varphi) &:= \text{Sub}(\psi) \cup \text{Sub}(\varphi) \cup \{\langle (p := \psi)^* \rangle\varphi\} \end{aligned}$$

As in the case for ML, the formulas of $\mathcal{L}_{\text{MISL}}$ are evaluated in *Kripke models* $\mathbf{M} = (W, R, V)$, where W is a non-empty set of states, $R \subseteq W \times W$ is a binary relation, and $V : \mathbf{P} \rightarrow 2^W$ is a valuation function. As usual, for any $w \in W$, (\mathbf{M}, w) is a *pointed model*. For simplicity, we often write \mathbf{M}, w . Moreover, we also write $s \in \mathbf{M}$ when $s \in W$.

Definition 2 (Semantics). *Let $\mathbf{M} = (W, R, V)$ be a model and $w \in W$. Truth of MISL-formulas φ at (\mathbf{M}, w) , written as $\mathbf{M}, w \models_{\text{MISL}} \varphi$, is given recursively as follows:*

$$\begin{aligned}
\mathbf{M}, w \models_{\text{MISL}} p &\Leftrightarrow w \in V(p) \\
\mathbf{M}, w \models_{\text{MISL}} \neg\varphi &\Leftrightarrow \mathbf{M}, w \not\models_{\text{MISL}} \varphi \\
\mathbf{M}, w \models_{\text{MISL}} \varphi_1 \wedge \varphi_2 &\Leftrightarrow \mathbf{M}, w \models_{\text{MISL}} \varphi_1 \text{ and } \mathbf{M}, w \models_{\text{MISL}} \varphi_2 \\
\mathbf{M}, w \models_{\text{MISL}} \diamond\varphi &\Leftrightarrow \mathbf{M}, v \models_{\text{MISL}} \varphi \text{ for some } v \in W \text{ s.t. } Rww \\
\mathbf{M}, w \models_{\text{MISL}} \langle p := \psi \rangle \varphi &\Leftrightarrow \mathbf{M}|_{p:=\psi}, w \models_{\text{MISL}} \varphi \\
\mathbf{M}, w \models_{\text{MISL}} \langle\langle p := \psi \rangle^*\rangle \varphi &\Leftrightarrow \mathbf{M}|_{(p:=\psi)^n}, w \models_{\text{MISL}} \varphi \text{ for some } n \in \mathbb{N}
\end{aligned}$$

where $\mathbf{M}|_{p:=\psi} = (W, R, V|_{p:=\psi})$ is obtained through model transformation from \mathbf{M} and disagrees with \mathbf{M} only on the valuation of p , where $V|_{p:=\psi}(p) = \{s \in W : \mathbf{M}, s \models_{\text{MISL}} \psi\}$. For any natural number n , $\mathbf{M}|_{(p:=\psi)^n} = (W, R, V|_{(p:=\psi)^n})$ is obtained by applying model transformations on \mathbf{M} , where $V|_{(p:=\psi)^n}(p)$ is defined in the following inductive manner: where $\mathbf{M}|_{p:=\psi} = (W, R, V|_{p:=\psi})$ is obtained through model transformation from \mathbf{M} and disagrees with \mathbf{M} only on the valuation of p , where $V|_{p:=\psi}(p) = \{s \in W : \mathbf{M}, s \models_{\text{MISL}} \psi\}$. For any natural number n , $\mathbf{M}|_{(p:=\psi)^n} = (W, R, V|_{(p:=\psi)^n})$ is obtained by applying model transformations, n times on \mathbf{M} , where $V|_{(p:=\psi)^n}(p)$ is defined in the following inductive manner:

$$\begin{aligned}
V|_{(p:=\psi)^0}(p) &:= V(p) \\
V|_{(p:=\psi)^{n+1}}(p) &:= \{s \in W : \mathbf{M}|_{(p:=\psi)^n}, s \models_{\text{MISL}} \psi\}
\end{aligned}$$

Note that the simple substitution operator $\langle p := \psi \rangle$ is self-dual, but the iterative substitution operators are not self-dual, i.e., $[\langle p := \psi \rangle^*] \varphi \leftrightarrow \langle\langle p := \psi \rangle^*\rangle \varphi$ fails in general.

Precisely, for any model \mathbf{M} and formula φ , the *truth set of φ in \mathbf{M}* is given by the set $\{s \in \mathbf{M} : \mathbf{M}, s \models_{\text{MISL}} \varphi\}$. Also, notions such as *satisfiability*, *validity* and *logical consequence* are defined as usual [6]. 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, $\langle p := \Box q \rangle \Box q \rightarrow p$ is not a validity, and for a counterexample, see Figure 2, taken from [22].

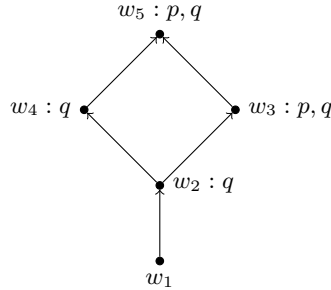


Fig. 2. A model \mathbf{M} in which $\langle p := \Box q \rangle \Box q \rightarrow p$ is false at w_1 . In the model, formula $\Box q$ is true everywhere. So, after the substitution, p is true everywhere, and in particular, $\Box q$ is still true at w_1 in the new model. However, as we can see, w_1 does not have the property p in \mathbf{M} .

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

Remark 1. The fragment of MISL without iterative substitutions has a complete and sound proof system, and is equivalently expressive as ML (despite being an extension of ML). Consequently, the satisfiability problem for this fragment of MISL is decidable. We refer the readers to [22] for details.

2.1 Validities of MISL

The following schemata for simple and iterative substitution operators respectively are valid:

Proposition 1. *The following recursion axioms involving simple substitution operators are valid:*

$$\begin{aligned} \langle p := \psi \rangle q &\leftrightarrow q \quad \text{given that } p \text{ is not } q. \\ \langle p := \psi \rangle p &\leftrightarrow \psi \\ \langle p := \psi \rangle \neg \varphi &\leftrightarrow \neg \langle p := \psi \rangle \varphi \\ \langle p := \psi \rangle (\varphi_1 \wedge \varphi_2) &\leftrightarrow \langle p := \psi \rangle \varphi_1 \wedge \langle p := \psi \rangle \varphi_2 \\ \langle p := \psi \rangle \diamond \varphi &\leftrightarrow \diamond \langle p := \psi \rangle \varphi \end{aligned}$$

Proposition 2. *The following formulas involving iterative substitution operators are valid:*

$$\begin{aligned} \langle \langle p := \psi \rangle^* \rangle q &\leftrightarrow q, \quad \text{if } q \text{ is not } p. \\ \langle \langle p := \psi \rangle^* \rangle p &\leftrightarrow p \vee \langle p := \psi; (p := \psi)^* \rangle p \\ [\langle p := \psi \rangle^*](\varphi \rightarrow [\langle p := \psi \rangle]\varphi) &\leftrightarrow (\varphi \rightarrow [\langle p := \psi \rangle^*]\varphi) \\ \langle \langle p := \psi \rangle^* \rangle \neg \varphi &\leftrightarrow \neg [\langle p := \psi \rangle^*]\varphi \\ \langle \langle p := \psi \rangle^* \rangle (\varphi_1 \wedge \varphi_2) &\rightarrow \langle \langle p := \psi \rangle^* \rangle \varphi_1 \wedge \langle \langle p := \psi \rangle^* \rangle \varphi_2 \\ \langle \langle p := \psi \rangle^* \rangle \diamond \varphi &\leftrightarrow \diamond \langle \langle p := \psi \rangle^* \rangle \varphi \end{aligned}$$

Note that the fifth formula is only one-directional. Some patterns above are similar to that of the recursion axioms in the proof system **MSL** [22], whereas the others have typical forms for iterative operations like those in propositional dynamic logic PDL [10].

2.2 Some useful concepts

Let us now take a closer look at the iterative substitution operators. As stated, an iterative substitution operator $\langle \langle p := \psi \rangle^* \rangle$ 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 $\langle \langle p := \psi \rangle^* \rangle$ 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 can be considered to be free, while those in the later stages can be considered to be bound. To make this clear, let us first introduce the following:

Definition 3 (Normal $\mathcal{L}_{\text{MISL}}$ -formulas). *An MISL-formula is normal if every iterative substitution $\langle p := \psi \rangle^*$ follows immediately after a single-step initialization $p := \psi_0$.*

For instance, $\langle p := \perp; (p := \Box p)^* \rangle p$ is a normal formula, while $\langle p := \perp; q := \top; (p := \Box p)^* \rangle p$ is not. Given a normal formula, its subformulas might not be normal.¹³ In a normal formula, each iterative substitution is a part of $p := \psi_0; (p := \psi)^*$. Now, we introduce the notions of *free occurrences*, *free variables*, *bound occurrences* and *bound variables* in *normal formulas*, needed for some technical considerations. To give an example, in formula $\langle p := p \vee q; (p := \Box p)^* \rangle p$, only the second occurrence of p is considered to be free, while its other occurrences are bound. To make this clear, we define the notions of *free variables* and *bound variables* in a formula:

¹³ For instance, formula $\langle p := \neg p; (p := \Box p)^* \rangle p$ is normal, but $\langle \langle p := \Box p \rangle^* \rangle p$ is not.

Definition 4 (Free variables and bound variables in $\mathcal{L}_{\text{MISL}}$). For any $\varphi \in \mathcal{L}_{\text{MISL}}$, an occurrence of a propositional variable p is a bound occurrence if it is a pivot or appears in the scope of an iterative 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 as we can see in the example before. Usually, it would be easier to consider the formulas in which no propositional letter is both free and bound:

Definition 5 (Clean $\mathcal{L}_{\text{MISL}}$ -formulas). A formula $\varphi \in \mathcal{L}_{\text{MISL}}$ is clean if $fv(\varphi) \cap bv(\varphi) = \emptyset$.

We note that it is enough to consider the clean formulas of MISL, since every MISL-formula can be transformed to an equivalent clean formula:

Fact 1 Every MISL-formula is equivalent to a clean formula.

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

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

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:

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

where the main connective of φ in $\langle p := \psi \rangle \varphi$ of the first equivalence is not an iterative substitution operator with p as its pivot, the propositional variable q is a fresh variable in both the rules, 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 6 (Replacement of variables in a normal formula). For any normal formulas $\varphi, \psi \in \mathcal{L}_{\text{MISL}}$, $\varphi[\psi/p]$ is defined as follows:

$$\begin{aligned} q[\psi/p] &:= \begin{cases} q & q \text{ is distinct from } p \\ \psi & q \text{ is } p \end{cases} \\ (\neg\varphi)[\psi/p] &:= \neg(\varphi[\psi/p]) \\ (\varphi_1 \wedge \varphi_2)[\psi/p] &:= \varphi_1[\psi/p] \wedge \varphi_2[\psi/p] \\ (\diamond\varphi)[\psi/p] &:= \diamond(\varphi[\psi/p]) \\ \langle (q := \chi)\varphi \rangle[\psi/p] &:= \begin{cases} \langle q := \chi[\psi/p] \rangle (\varphi[\psi/p]) & q \text{ is different from } p \\ \langle q := \chi[\psi/p] \rangle \varphi & q \text{ is } p \end{cases} \\ \langle (q := \chi_0; (q := \chi_1)^*)\varphi \rangle[\psi/p] &:= \begin{cases} \langle q := \chi_0[\psi/p]; (q := \chi_1[\psi/p])^* \rangle (\varphi[\psi/p]) & q \text{ is different from } p \\ \langle q := \chi_0[\psi/p]; (q := \chi_1)^* \rangle \varphi & q \text{ is } p \end{cases} \end{aligned}$$

where the main connective of φ in the first clause is not an iterative substitution operator with q as its pivot.

Remark 2. When saying that an MISL-formula φ is clean, we also assume that:

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

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

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

Theorem 1. $\langle p := \psi \rangle \varphi \leftrightarrow \varphi[\psi/p]$ is a validity, if both $\langle p := \psi \rangle \varphi$ and φ are clean formulas.

Proof sketch. Prove by induction¹⁴ on the structure of φ , and split into cases according to the main connective of φ . If the main connective of φ is \neg, \vee or \diamond , the statement follows directly with the help of the valid schemata in Proposition 1. If the main connective of φ is a simple substitution operator of form $\langle q := \chi \rangle$, then the statement follows by induction hypothesis and carefully applying the replacement definition in Definition 6. Lastly, by the constraint of $\langle p := \psi \rangle \varphi$ and φ both being clean, the main connective of φ cannot be an iterative substitution operator. This concludes the proof.

2.3 Example revisited: reasoning hide and seek games

Although we have defined the language of MISL as a direct extension of ML, we note that the dynamics of iterative substitution can apply to many more variants of modal logics. In this section, let us **informally** combine MISL with LHS to describe the winning positions of the hide and seek game, and establish a connection between backward induction as described in Section 1 and the corresponding iterative substitution operators. We will abuse certain notations in MISL in this section to denote the extension of LHS with both simple and iterative substitution operators. One can make all arguments rigorous by writing down the syntax and semantics of this extension as in the beginning of Section 2, which we will leave for our readers.

Recall that in the hide and seek game, the winning positions of Hider and Seeker are complementary. Therefore we focus on discussing the winning positions of Seeker. We start by introducing several key notions, which are useful for both our proof and the illustration of the advantages of our approach:¹⁵

Definition 7 (Ranks of winning positions). Let $\mathcal{B} = (W, R)$ be a finite graph for hide and seek game. For any $i \in \{0, 1\}$ and any $k \in \mathbb{N}$, we define $\text{Win}_{\mathcal{S}}^k \subseteq W \times W$ such that for any $s, t \in W$, $(s, t) \in \text{Win}_{\mathcal{S}}^k$ when Seeker has a strategy to win the game $\mathcal{G}_{s,t} = (\mathcal{B}, (s, t))$ within k rounds, where (s, t) are the initial positions for Hider and Seeker respectively in game $\mathcal{G}_{s,t}$. For all $0 < k \in \mathbb{N}$, the positions in $\text{Win}_{\mathcal{S}}^k \setminus \text{Win}_{\mathcal{S}}^{k-1}$ are called rank- k winning positions of Seeker, which means that Seeker 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 Seeker are exactly the members of $\text{Win}_{\mathcal{S}}^0$.

¹⁴ The induction here is not a usual induction on the length of $\langle p := \psi \rangle \varphi$, but following a designated well-founded order \ll on the clean MISL formulas as shown in Definition 11 (see [22] for proof details).

¹⁵ In fact, the notions and inference can be easily generalized to many combinatorial games.

We define $\text{Win}_S := \bigcup_{k \in \mathbb{N}} \text{Win}_S^k$, consisting of all finite-rank winning positions of Seeker. When a node is not a finite-rank winning position of Seeker, it is a winning position of Hider. Now, we are ready to show that the principle $\langle p := \mathcal{I}; (p := [\mathbf{H}]\langle S \rangle p)^* \rangle p$ ¹⁶ involved in Section 1 defines the set:

Proposition 3. *Let $\mathcal{B} = (W, R)$ be a finite graph for hide and seek game and \mathbf{M} be a model extending \mathcal{B} with an arbitrary valuation function. Then, in the model, $\langle p := \mathcal{I}; (p := [\mathbf{H}]\langle S \rangle p)^k \rangle p$ is true exactly at the nodes in Win_S^k . More generally, the formula*

$$\langle p := \mathcal{I}; (p := [\mathbf{H}]\langle S \rangle p)^* \rangle p$$

captures the set Win_S of winning positions of Seeker.

Proof. For convenience, let $R[s] = \{t : Rst\}$ denote the successors of $s \in W$. Define the functions $[R]_H, [R]_S : \mathcal{P}(W) \times \mathcal{P}(W) \rightarrow \mathcal{P}(W) \times \mathcal{P}(W)$ with their dual functions $\langle R \rangle_H$ and $\langle R \rangle_S$ as

$$\begin{aligned} [R]_H : X &\mapsto \{(s, t) : R[s] \times \{t\} \subseteq X\} \\ \langle R \rangle_H : X &\mapsto \{(s, t) : R[s] \times \{t\} \cap X \neq \emptyset\} \\ [R]_S : X &\mapsto \{(s, t) : \{s\} \times R[t] \subseteq X\} \\ \langle R \rangle_S : X &\mapsto \{(s, t) : \{s\} \times R[t] \cap X \neq \emptyset\} \end{aligned}$$

By the rules of hide and seek game, $\text{Win}_S^0 = \{(s, s) : s \in W\}$. We note that $[R]_H$ and $\langle R \rangle_S$ are both monotone functions, and that $\text{Win}_S^0 \subseteq [R]_H \circ \langle R \rangle_S(\text{Win}_S^0)$. Therefore $([R]_H \circ \langle R \rangle_S)^i(\text{Win}_S^0) \subseteq ([R]_H \circ \langle R \rangle_S)^j(\text{Win}_S^0)$ for any natural numbers $i \leq j$. For any $k \in \mathbb{N}$, the truth set of p in the k -th stage of the iterative substitution is exactly $([R]_H \circ \langle R \rangle_S)^k(\text{Win}_S^0)$, and when k grows the sequence of truth sets of p is monotone non-decreasing. Now we show by induction that $([R]_H \circ \langle R \rangle_S)^k(\text{Win}_S^0) = \text{Win}_S^k$ for all $k \in \mathbb{N}$. For $k = 0$ it holds directly. Suppose $([R]_H \circ \langle R \rangle_S)^k(\text{Win}_S^0) = \text{Win}_S^k$ for all $k \in \mathbb{N}$ holds for all $k \leq n$, then by backward induction, one can see that

$$\begin{aligned} \text{Win}_S^{n+1} &= ([R]_H \circ \langle R \rangle_S)(\text{Win}_S^n) \cup \bigcup_{i \leq n} \text{Win}_S^i \\ &= ([R]_H \circ \langle R \rangle_S)(\text{Win}_S^n) \cup \bigcup_{i \leq n} ([R]_H \circ \langle R \rangle_S)^i(\text{Win}_S^0) \\ &= ([R]_H \circ \langle R \rangle_S)(\text{Win}_S^n) \\ &= ([R]_H \circ \langle R \rangle_S)^{n+1}(\text{Win}_S^0) \end{aligned}$$

Therefore, by induction, the truth set of p in the k -th stage of the iterative substitution is exactly Win_S^k . As a direct consequence, the truth set of $\langle p := \mathcal{I}; (p := [\mathbf{H}]\langle S \rangle p)^* \rangle p$ in \mathbf{M} is $\bigcup_k \text{Win}_S^k = \text{Win}_S$. \square

It is also instructive to point out that in a finite graph for hide and seek game, Win_S can be expressed by

$$\mu p.(\mathcal{I} \vee ([\mathbf{H}]\langle S \rangle p))$$

of the *the modal μ -calculus* (μML) [15,23] extension of LHS. 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.

¹⁶ This formula belongs to a modal language extending LHS with iterative substitution operators, in the same way as MISL extends ML. Even though we do not define this extension here, we use this formula as a place-holder to showcase the connection between substitutions and the hide-and-seek game. We leave the formal definition for our readers.

3 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 3.1); meanwhile, the logic will be proved to be highly undecidable: its satisfiability problem being Σ_1^1 -complete (Section 3.2). Moreover, when we restrict our attention to the very simple class of finite tree models, the logic is still shown to be undecidable (Section 3.3).

3.1 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 8 (Bisimulation[6]). *Let $\mathbf{M} = (W, R, V)$ and $\mathbf{M}' = (W', R', V')$ 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 \mathbf{M} and \mathbf{M}' , if the following are satisfied:*

Atom: *If sZs' , then (\mathbf{M}, s) and (\mathbf{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 $s \in \mathbf{M}$ and $s' \in \mathbf{M}'$, we say that (\mathbf{M}, s) and (\mathbf{M}', s') are *bisimilar*, and write $(\mathbf{M}, s) \Leftrightarrow (\mathbf{M}', s')$, or just $s \Leftrightarrow s'$ when the models are clear from the context.

Theorem 2 (Invariance under bisimulation). *If $(\mathbf{M}, s) \Leftrightarrow (\mathbf{M}', s')$, then for any $\varphi \in \mathcal{L}_{\text{MISL}}$,*

$$\mathbf{M}, s \models_{\text{MISL}} \varphi \text{ iff } \mathbf{M}', s' \models_{\text{MISL}} \varphi.$$

Proof. We follow the proof provided in [22], and 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 the bisimulation that links (\mathbf{M}, s) and (\mathbf{M}', s') .

(1). Formula φ is $\langle p := \psi \rangle \chi$. Assume that $(t, t') \in Z$. We are going to show that $(\mathbf{M}|_{p:=\psi}, t) \Leftrightarrow (\mathbf{M}'|_{p:=\psi}, t')$. The difference between $\mathbf{M}|_{p:=\psi}$ and \mathbf{M} is only about the truth set of p , and similarly for $\mathbf{M}'|_{p:=\psi}$ and \mathbf{M}' . So, we only need to prove that $\mathbf{M}|_{p:=\psi}, t$ and $\mathbf{M}'|_{p:=\psi}, t'$ agree on the propositional variable p . By induction hypothesis, $\mathbf{M}, t \models_{\text{MISL}} \psi$ iff $\mathbf{M}', t' \models_{\text{MISL}} \psi$. Hence, $\mathbf{M}|_{p:=\psi}, t \models_{\text{MISL}} p$ iff $\mathbf{M}'|_{p:=\psi}, t' \models_{\text{MISL}} p$. Therefore, it holds that $(\mathbf{M}|_{p:=\psi}, t) \Leftrightarrow (\mathbf{M}'|_{p:=\psi}, t')$. Again, by induction hypothesis, $\mathbf{M}|_{p:=\psi}, s \models_{\text{MISL}} \chi$ iff $\mathbf{M}'|_{p:=\psi}, s' \models_{\text{MISL}} \chi$. Thus, based on the semantics, $\mathbf{M}, s \models_{\text{MISL}} \langle p := \psi \rangle \chi$ iff $\mathbf{M}', s' \models_{\text{MISL}} \langle p := \psi \rangle \chi$.

(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 $(\mathbf{M}|_{(p:=\psi)^n}, t) \Leftrightarrow (\mathbf{M}'|_{(p:=\psi)^n}, t')$. The base case $n = 0$ is direct. Also, for the case that $n = 1$, we have known from the reasoning in case (1) that $(\mathbf{M}|_{p:=\psi}, t) \Leftrightarrow (\mathbf{M}'|_{p:=\psi}, t')$. For any $n \geq 2$, by induction hypothesis, we have $(\mathbf{M}|_{(p:=\psi)^n}, t) \Leftrightarrow (\mathbf{M}'|_{(p:=\psi)^n}, t')$. Now, using the reasoning similar to that in case (1), we can show that $(\mathbf{M}|_{(p:=\psi)^{n+1}}, t) \Leftrightarrow (\mathbf{M}'|_{(p:=\psi)^{n+1}}, t')$. Therefore, we have the following sequence:

$$\begin{aligned} \mathbf{M}, s \models_{\text{MISL}} \langle (p := \psi)^* \rangle \chi &\Leftrightarrow \text{for some } n \in \mathbb{N}, \mathbf{M}|_{(p:=\psi)^n}, s \models_{\text{MISL}} \chi \\ &\Leftrightarrow \text{for some } n \in \mathbb{N}, \mathbf{M}'|_{(p:=\psi)^n}, s' \models_{\text{MISL}} \chi \\ &\Leftrightarrow \mathbf{M}', s' \models_{\text{MISL}} \langle (p := \psi)^* \rangle \chi \end{aligned}$$

This completes the proof. \square

3.2 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 [17].¹⁷ 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 4. MISL lacks the finite model property.

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

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

First, the formula is satisfiable. One can check that it is true at the state s in model \mathbf{M} depicted in Figure 3.

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. \square

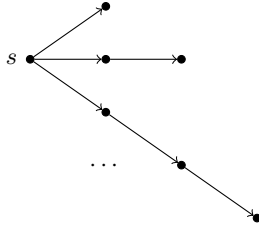


Fig. 3. An infinite model \mathbf{M} such that $\mathbf{M}, s \models_{\text{MISL}} [p := \Box \perp; (p := \neg p \wedge \Box p)^*] \Diamond p$

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

Theorem 3 (Undecidability of MISL). *The satisfiability problem for MISL is undecidable and is Σ_1^1 -complete.*

To prove Theorem 3, we show the satisfiability problem for MISL is on the one hand undecidable and Σ_1^1 -hard, by embedding iterative modal relativization IMR [17] into MISL, and on the other hand inside Σ_1^1 , by expressing the satisfiability problem for MISL using a Σ_1^1 -formula. In the sequel we show the ideas of the proof for both directions, for the detailed proof, see [22].

The satisfiability problem for MISL is Σ_1^1 -hard. We'll construct a translation from IMR to MISL, which implies that the satisfiability problem for MISL is Σ_1^1 -hard. Here is a quick introduction to IMR:

Basics of IMR The language \mathcal{L}_{IMR} of IMR extends \mathcal{L}_{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).

¹⁷ The investigation into relation between MISL and IMR has been suggested in [5].

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

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

In the definition above, the restriction $p \notin \text{bv}(\varphi)$ is used to exclude the situations that p is bounded by some substitution operators in φ . The condition is harmless, since with the rule ($\text{Renaming}_{\text{MISL}}$) one can always apply bound variable renaming on φ . The following lemma intuitively states that $(\varphi)^p$ is equivalent to $[p]\varphi$:

Lemma 1. *Let $\mathbf{M} = (W, R, V)$ be a model, $s \in W$ and $p \in \mathbf{P}$. For any $\varphi \in \mathcal{L}_{\text{MISL}}$ such that $p \notin \text{bv}(\varphi)$, it holds that*

$$\mathbf{M}, s \models_{\text{MISL}} (\varphi)^p \quad \text{iff} \quad s \notin W^p \text{ or } \mathbf{M}^p, s \models_{\text{MISL}} \varphi,$$

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

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

The proof of Lemma 1 is done by induction on the structure of φ , which is neglected here. Equipped with the intermediate translation, we're ready to show the full translation from IMR to MISL:

Definition 10 (Translation from IMR to MISL). *The translation $T : \mathcal{L}_{\text{IMR}} \rightarrow \mathcal{L}_{\text{MISL}}$ is given by the following:*

$$\begin{aligned} T(p) &:= p \\ T(\neg\varphi) &:= \neg T(\varphi) \\ T(\varphi_1 \wedge \varphi_2) &:= T(\varphi_1) \wedge T(\varphi_2) \\ T(\Box\varphi) &:= \Box T(\varphi) \\ T(\Box^*\varphi) &:= [q := T(\varphi); (q := \Box q)^*]q \\ T([\psi]\varphi) &:= \langle q := T(\psi) \rangle (T(\varphi))^q \\ T(\langle\psi^*\rangle\varphi) &:= \langle q := \top; (q := q \wedge (T(\psi))^q)^* \rangle (T(\varphi))^q \end{aligned}$$

where q is a fresh variable.

The following theorem states that the translation defined above preserves formula equivalence:

Theorem 4. *For any pointed model (\mathbf{M}, s) and any formula $\varphi \in \mathcal{L}_{\text{IMR}}$, we have the following:*

$$\mathbf{M}, s \models_{\text{IMR}} \varphi \quad \text{iff} \quad \mathbf{M}, s \models_{\text{MISL}} T(\varphi)$$

where T is the translation given in Definition 10.

The proof for Theorem 4 is done by induction on the structure of $\varphi \in \mathcal{L}_{\text{IMR}}$, which is neglected here. For a detailed proof see [22]. The correct translation from IMR to MISL shows that the satisfiability problem for MISL is Σ_1^1 -hard.

The satisfiability problem for MISL is Σ_1^1 -complete. It remains to show that the problem is in Σ_1^1 , which concludes that MISL is Σ_1^1 -complete. To do so, let us first argue that, informally, one can enumerate the subformulas of any MISL-formula in finite many steps in parallel such that this set of formulas admits induction on their structure. In another words, we show there is a well-founded order on MISL-formulas:

Definition 11 (Order on clean formulas of MISL). We define a binary relation \ll on clean $\mathcal{L}_{\text{MISL}}$ -formulas:

$$\begin{aligned} \varphi \ll \neg\varphi, \quad \varphi \ll \varphi \wedge \varphi, \quad \varphi \ll \Box\varphi, \quad \varphi[\psi/p] \ll \langle p := \psi \rangle \varphi, \\ \langle p := \psi_0; (p := \psi_1)^n \rangle \varphi \ll \langle p := \psi_0; (p := \psi_1)^* \rangle \varphi, \end{aligned}$$

where $n \in \mathbb{N}$, the fourth clause only works for φ whose main connective is not an iterative substitution operator with p as its pivot (otherwise $\langle p := \psi \rangle \varphi$ should follow the fifth clause), and the replacement $\varphi[\psi/p]$ is defined in Definition 6. We use \gg for the converse of \ll .

Note that the relation is defined over the clean formulas of $\mathcal{L}_{\text{MISL}}$, rather than the full language. 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. The restrictive definition over clean formulas does not cause too much loss, since any MISL-formula is equivalent to a clean formula.

We state the lemma for the well-founded-ness of the relation \ll , and neglect the proof by induction.

Lemma 2. *The order \ll defined in Definition 11 is well-founded.*

Now we proceed to define the *evaluation relation* that intuitively expresses whether or not a formula φ is true at a pointed model:

Definition 12. Let $\mathbf{M} = (W, R, V)$ be a model and $\mathcal{L}_{\text{clean}}$ be the set of clean MISL-formulas. We define a binary relation $E_{\mathbf{M}} \subseteq \mathcal{L}_{\text{clean}} \times W$ such that:

$$E_{\mathbf{M}}(\varphi, s) \quad \text{iff} \quad \mathbf{M}, s \models_{\text{MISL}} \varphi.$$

With regard to a model \mathbf{M} and its associated $E_{\mathbf{M}}$, we can show the following:

Lemma 3. *Given a model $\mathbf{M} = (W, R, V)$, for all $s \in W$, it holds that $V(p) = \{s : E_{\mathbf{M}}(p, s)\}$, and also, all the following properties hold for $E_{\mathbf{M}}$ and R :*

$$\begin{aligned} E_{\mathbf{M}}(\perp, s) &\leftrightarrow \perp \\ E_{\mathbf{M}}(\neg\varphi, s) &\leftrightarrow \neg E_{\mathbf{M}}(\varphi, s) \\ E_{\mathbf{M}}(\varphi_1 \wedge \varphi_2, s) &\leftrightarrow E_{\mathbf{M}}(\varphi_1, s) \wedge E_{\mathbf{M}}(\varphi_2, s) \\ E_{\mathbf{M}}(\Box\varphi, s) &\leftrightarrow \forall t \in W (Rst \rightarrow E_{\mathbf{M}}(\varphi, t)) \end{aligned}$$

$$E_{\mathbf{M}}(\langle p := \psi \rangle \varphi, s) \leftrightarrow E_{\mathbf{M}}(\varphi[\psi/p], s)$$

$$E_{\mathbf{M}}(\langle p := \psi_0; (p := \psi_1)^* \rangle \varphi, s) \leftrightarrow \exists n \in \mathbb{N} E_{\mathbf{M}}(\langle p := \psi_0; (p := \psi_1)^n \rangle \varphi, s)$$

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

This lemma can be proved by induction on formulas in $\mathcal{L}_{\text{clean}}$ along \ll (with the help of Theorem 1). 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 5. *The satisfiability problem for MISL is in Σ_1^1 .*

Proof sketch. We use the fact that MISL can be treated as a fragment of ML^∞ , since we can define the translation $T_0 : \mathcal{L}_{\text{MISL}} \rightarrow \mathcal{L}_{\text{ML}^\infty}$ recursively, who translates the iterative substitution $\langle (p := \psi_1)^* \rangle \varphi$

$$T_0(\langle (p := \psi_0; (p := \psi_1)^* \rangle \varphi) := \bigvee_{n \in \mathbb{N}} T_0(\langle (p := \psi_0; (p := \psi_1)^n \rangle \varphi)$$

It is direct from the definition that the translation T_0 translates each MISL formula to an equivalent ML^∞ formula. Since the latter has the downward Löwenheim-Skolem property, an $\mathcal{L}_{\text{MISL}}$ -formula is satisfiable iff it is satisfiable w.r.t. countable models $\mathbf{M} = (\omega, R, V)$.

In the sequel, we only need to consider the clean MISL-formulas of $\mathcal{L}_{\text{clean}}$, since for an arbitrary MISL-formula, we can always recursively transform it into an equivalent clean formula via (Renaming_{MISL}). Our method is motivated by [17]. We show that for any $\varphi \in \mathcal{L}_{\text{clean}}$, φ is satisfiable if and only if there exist relations $R \subseteq \omega \times \omega$ and $E \subseteq \mathcal{L}_{\text{clean}} \times \omega$ such that the equivalences in Lemma 3 hold and that $E(\varphi, m)$ for some $m \in \omega$. More specifically, we show that the satisfiability of φ is equivalent to the following Σ_1^1 -formula (*) written informally:

$$\exists R \exists E \forall s \in \omega, E \text{ and } R \text{ satisfy all clauses listed in Lemma 3,}$$

w.r.t. any clean formulas appearing in the clauses following the corresponding restriction.

To formally write the formula (*), it should contain the conjunction of the satisfaction of each clause in Lemma 3, e.g., the satisfaction of the second clause is described by $\forall \chi \in \mathcal{L}_{\text{clean}} (E(\neg \chi, s) \leftrightarrow \neg E(\chi, s))$ where $\neg \chi$ an abbreviation for the recursive function $f_- : \mathcal{L}_{\text{MISL}} \rightarrow \mathcal{L}_{\text{MISL}}$ such that $f_- : \chi \mapsto \neg \chi$. Then we proceed to argue that the satisfiability of φ is equivalent to formula (*).

On the one hand, assume that φ is satisfiable. Then, there is a countable model $\mathbf{M} = (\omega, R, V)$ such that $\mathbf{M}, m \models_{\text{MISL}} \varphi$ for some $m \in \omega$. Let E be a relation such that $E(p, s)$ iff $s \in V(p)$, and for any $\chi \in \mathcal{L}_{\text{clean}}$, $E(\chi, s)$ is defined recursively according to the equivalences in Lemma 3. It can be proved by induction on formulas along relation \ll that E agrees with $E_{\mathbf{M}}$ (the evaluation relation defined in Lemma 3), and hence $E(\varphi, m)$ holds. Therefore we can find R, E and m that make the Σ_1^1 -formula (*) true.

On the other hand, when the (*) is true for some R, E and m , we just need to set $\mathbf{M} = (\omega, R, V)$ where $V(p) = \{s : E(p, s) \text{ holds}\}$. By induction on formulas along the relation \ll , we can show that $\mathbf{M}, m \models_{\text{MISL}} \varphi$ is the case.

To sum up, the satisfiability problem of φ is in Σ_1^1 . Therefore MISL is in Σ_1^1 . \square

With Theorems 4 and 5, we have Theorem 3 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 4). Also, since both MISL and IMR are Σ_1^1 -complete, there is a converse reduction in principle, which we leave for further study.

3.3 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 definition is as follows:

Definition 13 (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 $U = (u_1, u_2, \dots, u_m) \in (\Sigma^*)^m$ for some $m \in \mathbb{N}$, $V = (v_1, v_2, \dots, v_m) \in (\Sigma^*)^m$ and the pairs $(u_1, v_1), (u_2, v_2), \dots, (u_m, v_m)$ are distinct. The post correspondence problem is to find whether there is a finite sequence of pairs $(u_{i_1}, v_{i_1}), (u_{i_2}, v_{i_2}), \dots, (u_{i_l}, v_{i_l})$ such that for any $1 \leq j \leq l$, $i_j \in \{1, \dots, m\}$ and $u_{i_1}u_{i_2}\dots u_{i_l} = v_{i_1}v_{i_2}\dots v_{i_l}$, 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 a well-known undecidable problem [12]. We prove the undecidability of MISL over finite tree models by showing the following statement: 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. *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*

$$\mathbf{M}, w \models_{\text{MISL}} \varphi(U, V) \text{ for some finite tree model } \mathbf{M} \text{ iff } (U, V) \text{ has a solution.}$$

Proof sketch. We briefly explain how to construct the MISL-formula $\varphi(U, V)$, and refer readers to the full proof in [22]. Let's call a sequence of pairs $(u_{i_1}, v_{i_1}), (u_{i_2}, v_{i_2}), \dots, (u_{i_l}, v_{i_l})$ a length- l configuration for the post correspondence problem (U, V) . Ideally, we'd like the formula $\varphi(U, V)$ to express the following: starting from the empty configuration (length-0), there is a way to iteratively append a new pair to the current configuration for finitely many times, such that the resulting configuration is a solution to the post correspondence problem (U, V) . An informal way to write down the desired formula $\varphi(U, V)$ is as follows:

$$\begin{aligned} \varphi(U, V) := & \langle \text{configuration} := \text{empty}; \\ & (\text{configuration} := \bigvee_{i=1}^m \text{new configuration obtained by appending } (u_i, v_i))^* \rangle \\ & \text{configuration indicates a solution} \end{aligned}$$

Note that from the intuitive construction of $\varphi(U, V)$ above, the iteratively substituted set of **configuration** must contain multiple configurations to the PCP, since in each iteration **configuration** is extended by all the possible pairs. We resolve this issue by constraining the truth set of **configuration** to only contain a single possible configuration, by designing more restrictive formulas. Then one can argue that on the one hand, $\varphi(U, V)$ is satisfiable in some finite tree model \mathbf{M} then \mathbf{M} can be transformed to a PCP solution, and on the other hand, if the PCP has a solution that is a finite sequence of pairs, then it can be transformed to a finite tree model that satisfies $\varphi(U, V)$. \square

Combined with the fact that PCP over a binary alphabet is undecidable, we conclude that the satisfiability problem for MISL on finite tree models is undecidable. We leave the exact complexity of the satisfiability problem for MISL over finite trees as an open problem.

Remark 3. We would also like to identify a decidable fragment of MISL or some of its variants, given that the logic is highly undecidable. We have already ruled out certain approaches, e.g., restricting the logic to finite tree models. We leave this for the future.

Model-checking problem of MISL Before ending the discussion on various properties of the logic MISL, we note that the model-checking problem for MISL has not been explored here. It can be shown that the problem is decidable, but we do not go into the details. The complexity of the said problem is left for further study.

4 On relations between MISL and other relevant logics

In Section 2.3, we have shown that MISL can be used to reason about the winning positions for players, and in particular, given a hide and seek game, the formula $\langle p := \mathcal{I}; (p := p \vee [\mathbf{H}]\langle \mathbf{S} \rangle p)^* \rangle p$ captures the winning positions Wins_S of Seeker. However, as mentioned, this can also be defined with an extension of 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 propositional dynamic logic* PDL [10], *iterative modal relativization* IMR and logic for the inflationary fixed-point MIC [7]. We skip the discussion concerning the connection between our framework and IMR and refer the readers to the previous section.

4.1 Relation between MISL and μML

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

Remark 4. 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 7). For instance, consider the formula for the hide and seek game

$$\langle p := \mathcal{I}; (p := [\mathbf{H}]\langle \mathbf{S} \rangle p)^* \rangle (\langle \mathbf{S} \rangle p \wedge \langle \mathbf{S} \rangle (\neg p \wedge [\mathbf{H}]\langle \mathbf{S} \rangle p))$$

saying that at the current position (s, t) , Seeker can move to two positions t_1 and t_2 such that (s, t_1) and (s, t_2) are winning positions for Seeker but of different ranks. In Figure 1, the pairs of positions (b, c) is a rank-1 winning position for Seeker, (a, b) , (a, c) are of rank 2, and (b, b) satisfies the above formula.

However, the property described by the above formula cannot be defined by μML . To actually show the separation that $\text{MISL} \not\subseteq \mu\text{ML}$ (see [22]), we show that the formula $\langle p := \Box \perp; (p := p \vee \Box \Diamond p)^* \rangle (\Diamond p \wedge \Diamond (\neg p \wedge \Box \Diamond p))$ cannot be defined by μML , which describes the property saying that the current state has two neighbours that are winning positions for the \Diamond -player but of different ranks, on a finite board game¹⁸. This is the same property as we described above in the hide and seek game. One can easily convert the separation between $\text{MISL} \not\subseteq \mu\text{ML}$ to the statement saying LHS with iterative substitution is not a fragment of LHS extended with modal μ -calculus.

The proof is done by showing that $\langle p := \Box \perp; (p := p \vee \Box \Diamond p)^* \rangle (\Diamond p \wedge \Diamond (\neg p \wedge \Box \Diamond p))$ can define irregular languages, while μML cannot since μML can be embedded in *monadic second order logic* (MSO) which only defines the set of regular languages.

On the other hand, recall that to calculate the least fixed-point of a monotone function in μML , one starts with \perp and computes the approximation sequence. As mentioned in [5], if the least fixed

¹⁸ The finite board game is defined as a game on a board with finite states and edges between states, where two players move a single token from state to state in turn, and either player wins if the other player cannot make a move. See [22] for more details.

point of φ can be reached by a finite sequence of approximation, then the μML -formula $\mu p.\varphi$ is equivalent to

$$\langle p := \perp; (p := \varphi)^* \rangle p.^{19}$$

Therefore we have the following as a corollary:

Corollary 1. *The fragment of μML where the fixed point of every formula can be reached by a finite sequence of approximation is strictly contained in MISL.*²⁰

4.2 Relation between MISL and PDL

Let us turn to another logical framework, PDL, involving finite steps of iteration. It is well known that PDL can be viewed as a continuous fragment of μML , and thus, PDL can be embedded into MISL by simply applying the embedding of μML in MISL mentioned in the previous section. Nevertheless, we present a direct embedding from PDL into MISL in this section, for a better understanding of the similarity between the two logics. Since the language \mathcal{L}_{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 $\mathcal{L}_{\text{MISL}(\Delta)}$ and $\text{MISL}(\Delta)$ for the resulting language and logic respectively. We now show that PDL can be embedded into $\text{MISL}(\Delta)$:

Theorem 7. *There exists a translation $T : \mathcal{L}_{\text{PDL}} \rightarrow \mathcal{L}_{\text{MISL}(\Delta)}$ such that for any PDL-formula φ , it holds that*

$$\mathbf{M}, s \models_{\text{PDL}} \varphi \quad \text{iff} \quad \mathbf{M}, s \models_{\text{MISL}(\Delta)} T(\varphi).$$

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

Proof sketch. 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. \square

4.3 Relation between MISL and MIC

We have discussed about the relation between MISL and μML in that MISL can represent the least fixed-point of monotone functions computed by finite approximation sequences. In general, when only finite approximation sequences are allowed, MISL is capable of describing many types of fixed-points.²¹ Another example is the inflationary fixed-point, whose extension over ML is the *modal*

¹⁹ It is important to emphasize that we have this equivalence with the convention that in μML only finite sequences of approximation are allowed (Remark 4). And, we do *not* have the equivalence when ordinal sequences are allowed in μML : for instance, $\mu p.\Box p$ is true (\mathbf{M}, s) depicted in Figure 3 while $\langle p := \perp; (p := \Box p)^* \rangle p$ is false there.

²⁰ Without the restriction imposed in the corollary, μML is not contained in MISL (see [22] for details).

²¹ A similar point has been made in [5], where MISL is introduced to encode a modal fixed-point formula.

iterated calculus (MIC) [7]. Syntactically, MIC extends ML with the operator (**ifp** $X_i : S$) where S is a system of rules w.r.t. variables X_1, X_2, \dots, X_k and MIC-formulas $\varphi_1, \varphi_2, \dots, \varphi_k$:

$$S := \begin{cases} X_1 \leftarrow \varphi_1 \\ X_2 \leftarrow \varphi_2 \\ \dots \\ X_k \leftarrow \varphi_k \end{cases}$$

The system S defines the (ordinal-length) approximation sequence via inflationary induction. The truth set of (**ifp** $X_i : S$) is the i -th coordinate in the inflationary fixed-point reached by the inflationary induction defined by S starting from the original mode. If S only contains one line of form $X \leftarrow \varphi$, then we write (**ifp** $X : X \leftarrow \varphi$) for simplicity. The fragment of MIC without simultaneous inductions (i.e., with only one-line systems) is called 1MIC.

We claim that when only finite sequences of inflationary inductions are allowed, 1MIC is strictly contained in MISL. This is because on the one hand, when only finite sequences of inflationary inductions are allowed, the inflationary fixed-point (**ifp** $X : X \leftarrow \varphi$) can be computed by $\langle X := \perp; (X := X \vee \varphi)^* \rangle X$, and therefore $\text{MIC} \subseteq \text{MISL}$. On the other hand, [7] showed that the MIC-formula

$$\begin{aligned} \text{ifp } Y : X \leftarrow \Box \Box X \\ Y \leftarrow \neg X \wedge \Box X \end{aligned} \quad (*)$$

is not equivalent to any 1MIC formula, even when restricted to finite models. However, on finite models, formula (*) is equivalent to the MISL formula $\langle X := \perp; (X := \Box \Box X)^* \rangle (\neg X \wedge \Box X)$, which consequently is not equivalent to any 1MIC formula. Therefore when only finite sequences of inflationary inductions are allowed, 1MIC is strictly a smaller segment of MISL.

We leave it an open problem to discover where is MISL in the hierarchy of MIC consisting of all k MIC's, where k MIC is the fragment of MIC that only allows systems of no more than k lines.²²

We have seen a number of expressive languages that can be encoded within MISL²³, which agrees with the high complexity of the framework presented in the previous section.

5 Conclusion

Summary In this chapter we develop a logical framework containing iterative substitutions as modal operators that update valuation of propositional variables. That provides a way to deal with winning strategies in hide and seek games. We develop various validities to show how those operators work and analyze its applications to crucial notions in games. Besides, 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. Moreover, we compare MISL with many other important logics that share a similar iterative feature, including the modal μ -calculus, the propositional dynamic logic PDL, the iterative modal relativization IMR and the inflationary fixed-point logic MIC, which also illustrates advantages of our logic.

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

²² In fact, the separation among this hierarchy is left open in [7].

²³ Moreover, the undecidable framework iterative modal relativization IMR [17] can also be encoded within MISL (see Definition 10).

are many important logics that have the iterative reasoning concepts, including the fixed-point logics we mentioned and the modal logic of oscillations [4]. Some of them can also be translated into MISL or its further extensions, 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. Finally, going beyond the current framework, it is also meaningful to allow ordinal sequences of iteration and study the iteration of the combined simple substitutions like $\langle(p := \psi; q = \chi)^*\rangle\varphi$, to make the logics applicable in broader scenarios.

Acknowledgements. We are grateful to Johan van Benthem for his insightful discussions and questions that guided this project from its early stages to its current form. We also thank Alexandru Baltag for his inspiring comments. Yaxin Tu is supported by Simons Investigator Award and by the National Science Foundation of USA [CCF-2007462]. Sujata Ghosh acknowledges the 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. Fenrong Liu is supported by the Tsinghua University Initiative Scientific Research Program. Dazhu Li is supported by the National Social Science Foundation of China [22CZX063].

References

1. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements and common knowledge and private suspicions. In: Gilboa, I. (ed.) *Proceedings of TARK '98*. pp. 43–56 (1998)
2. van Benthem, J.: *Modal Logic for Open Minds*. CSLI Publications (2010)
3. van Benthem, J.: *Logical Dynamics of Information and Interaction*. Cambridge University Press (2011)
4. van Benthem, J.: Oscillations, logic, and dynamical systems. In: Ghosh, S., Szymanik, J. (eds.) *The Facts Matter*. p. 9–22. College Publications (2015)
5. van Benthem, J.: An abstract look at the fixed-point theorem for provability logic. In: Bezhanishvili, N., Iemhoff, R., Yang, F. (eds.) *Dick de Jongh on Intuitionistic and Provability Logics, Outstanding Contributions to Logic*, vol. 28, pp. 75–88. Springer (2024)
6. Blackburn, P., de Rijke, M., Venema, Y.: *Modal Logic*. Cambridge University Press (2001)
7. Dawar, A., Grädel, E., Kreutzer, S.: Inflationary fixed points in modal logic. *ACM Transactions on Computational Logic* **5**, 282–315 (2004)
8. van Ditmarsch, H., van der Hoek, W., Kooi, B.: *Dynamic Epistemic Logic*. Springer (2007)
9. Gale, D., Stewart, F.M.: Infinite games with perfect information. In: Kuhn, H.W., Tucker, A.W. (eds.) *Contributions to the Theory of Games, Volume II, Annals of Mathematics Studies*, vol. 28, pp. 245–266. Princeton University Press (1953)
10. Harel, D., Kozen, D., Tiuryn, J.: *Dynamic Logic*. The MIT Press (2000)
11. van der Hoek, W., Wooldridge, M.: On the logic of cooperation and propositional control. *Artificial Intelligence* **164**, 81–119 (2005)
12. Hopcroft, J.E., Motwani, R., Ullman, J.D.: *Introduction to Automata Theory, Languages, and Computation*. Pearson, 3 edn. (2006)
13. Kooi, B.: Dynamic term-modal logic. In: van Benthem, J., Ju, S., Veltman, F. (eds.) *Proceedings of LORI 2007*. pp. 173–185. College Publications (2007)
14. Kooi, B.: Expressivity and completeness for public update logics via reduction axioms. *Journal of Applied Non-Classical Logics* **17**, 231–253 (2007)
15. Kozen, D., Parikh, R.: A decision procedure for the propositional μ -calculus. In: Clarke, E., Kozen, D. (eds.) *Proceedings of Workshop on Logics of Programs 1983*. *Lecture Notes in Computer Science*, vol. 164, pp. 313–325. Springer (1984)
16. Li, D., Ghosh, S., Liu, F., Tu, Y.: A simple logic of the hide and seek game. *Studia Logica* **111**, 821–853 (2023)

17. Miller, J.S., Moss, L.S.: The undecidability of iterated modal relativization. *Studia Logica* **79**, 373–407 (2005)
18. Plaza, J.A.: Logics of public communications. In: Emrich, M.L., Pfeifer, M.S., Hadzikadic, M., Ras, Z.W. (eds.) *Proceedings of the 4th international symposium on methodologies for intelligent systems*, pp. 201–216. Oak Ridge National Laboratory (1989)
19. Radev, S.: Infinitary propositional normal modal logic. *Studia Logica* **46**, 291–309 (1987)
20. Tang, W.: Substitution as Modality. Bachelor’s thesis (in Chinese), Peking University (2022)
21. Thompson, D.: Local fact change logic. In: Liu, F., Ono, H., Yu, J. (eds.) *Knowledge, Proof and Dynamics*. pp. 73–96. *Logic in Asia: Studia Logica Library*, Springer (2020)
22. Tu, Y., Ghosh, S., Liu, F., Li, D.: A modal approach towards substitution (2025), (submitted)
23. Venema, Y.: Lectures on the Modal μ -calculus. <https://staff.science.uva.nl/y.venema/teaching/ml/notes/20201212-mu.pdf> (2020)