# Completeness results for memory logics

Carlos Areces [a,*], Santiago Figueira [b,c], Sergio Mera [b]

[a] INRIA Nancy Grand Est, France
[b] Departamento de Computación, FCEyN, Universidad de Buenos Aires, Argentina
[c] CONICET, Argentina

A R T I C L E  I N F O

A B S T R A C T

Memory logics are a family of modal logics in which standard relational structures are augmented with data structures and additional operations to modify and query these structures. In this paper we present sound and complete axiomatizations for some members of this family. We analyze the use of nominals to achieve completeness, and present one example in which they can be avoided.

© 2011 Elsevier B.V. All rights reserved.

## 1. Modal logics with memory

Modal logics [9,7] can be considered nowadays as languages specially designed to describe properties of relational structures. They try to find a balance between expressive power, easy of use, and computational complexity. Many attempts have been made in recent years to increase modal logic expressivity by adding some notion of *state* to standard relational structures. This is a natural need, since modal logics are used in many different scenarios as tools for modeling behavior.

One example of such logics are epistemic logic with dynamic operators. These languages are used to express the evolution of knowledge by means of knowledge-changing actions. Such logics are often called Dynamic Epistemic Logics (DEL) [17], and a large number of DELs have been proposed [10,14–16]. These logics differ considerably in expressive power among themselves, but the common idea is to represent knowledge evolution by accessing and changing the model structure through logic operators. For example, representing the fact that an agent obtains the information that $\varphi$ is true in state $w$ amounts to eliminating all possible successor states where $\varphi$ does not hold.

Other examples of logics which have the ability to model behavior are some of the languages used by the software verification community. The logic XCTL of Harel et al. [11], for example, is a temporal logic with explicit global clocks which are accessed and controlled through logic operators. Also from the software verification community, we could mention the extension of temporal logic with a concrete domain (e.g., the natural numbers with some operations like addition, comparison, etc.) which is accessed via the so-called freeze operator [1,12]. In the extended language, we can model qualitative properties using the temporal operators, and concrete properties – such as weight, temperature, etc. – using the new machinery. To cite yet another example, concrete domains have also been added to description logics, with much the same aims [13].

---

* Corresponding author.
*E-mail addresses:* carlos.areces@loria.fr (C. Areces), santiago@dc.uba.ar (S. Figueira), smera@dc.uba.ar (S. Mera).

We would like to take a step back, and analyze some of the basic intuitions that most of the formal languages mentioned above have in common. We want to try to investigate the idea of adding an *explicit state* to a model, and being able to access (and modify) it via logical operators. And we would like to take this idea in its simplest form, in order to be able to understand it in detail.

We can take a standard relational structure and complement it with a *data structure*, that will keep the state information we want to model. We will also add to the logical language a collection of operations to modify and access the data structure. Formally, given a relational structure $\langle D, (R_r)_{r \in \mathsf{Rel}}, L \rangle$ where $D$ is a non empty domain, $(R_r)_{r \in \mathsf{Rel}}$ is a set of relations over $D$, and $L : \mathsf{Atom} \to 2^D$ is a labeling function that assigns atomic properties to elements of $D$, we extend the structure with a set $S \subseteq D$. We can think of $S$ as a set of states that are 'known' to us, and it will represent our current 'memory'. Even in this simple setting we can define the following operators:

$$\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle, w \models \text{ⓡ}\varphi \quad \text{iff} \quad \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \cup \{w\} \rangle, w \models \varphi$$
$$\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle, w \models \text{ⓚ} \quad \text{iff} \quad w \in S.$$

As it is clear from the definition above, the 'remember' operator ⓡ (a unary modality) just marks the current state as being 'known' or 'already visited', by storing it in our 'memory' $S$. On the other hand, the zero-ary operator ⓚ (for 'known') queries $S$ to check if the current state has already been visited. Notice that the extension of $S$ is dynamic and it can vary during the evaluation of a formula; while the 'concrete' operation we can apply to $S$ is simple membership.

Other operators can naturally be added, for example:

$$\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle, w \models \text{ⓕ}\varphi \quad \text{iff} \quad \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \backslash \{w\} \rangle, w \models \varphi$$
$$\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle, w \models \text{ⓔ}\varphi \quad \text{iff} \quad \langle D, (R_r)_{r \in \mathsf{Rel}}, L, \emptyset \rangle, w \models \varphi.$$

I.e., we can use the forget operator ⓕ to eliminate the current point of evaluation from the memory $S$, while the erase operator ⓔ completely wipes out the memory $S$. We have introduced this family of logics, that we called *memory logics*, and investigated its expressive power in [2–4].

The language we have just described is very flexible, and it can be used to easily characterize model properties. For example if all states in the domain of a model $\mathcal{M}$ satisfy the formula ⓔⓡ$\langle r \rangle$ⓚ then the relation $R_r$ is reflexive (we wipe out the memory, memorize the current point of evaluation and verify that it is accessible). Similarly, if they satisfy ⓔⓡ$[r]\langle r \rangle$ⓚ then $R_r$ is symmetric. Actually, using ⓕ, ⓔ, ⓡ and ⓚ we can express properties similar to how it is done using *binders* in different hybrid languages [5,6].

The two families of memory and hybrid logics are intimately related, but there are differences among them. In [3,4], for example, we have shown that $\mathcal{ML}(\text{ⓡ}, \text{ⓚ})$,[1] the modal language extended with ⓡ and ⓚ, is strictly more expressive than the basic modal logic but strictly less expressive than the hybrid logic $\mathcal{HL}(\downarrow)$. If we add the ⓔ operator to $\mathcal{ML}(\text{ⓡ}, \text{ⓚ})$, the resulting language is still strictly less expressive than the hybrid logic $\mathcal{HL}(\downarrow)$. Furthermore, we also know that if we add ⓕ to $\mathcal{ML}(\text{ⓡ}, \text{ⓚ})$ we have at most the expressive power of $\mathcal{HL}(\downarrow)$. It is an open problem whether this inequality is strict although we believe it is. These two last results are not yet published. The former follows using an argument similar to the proof of Theorem 6 of [3] and the latter follows using a translation as the one used in the proof of Theorem 5 of [3].

In this article we are interested in providing complete axiomatizations for memory logics. With this aim in mind, we will extend the language of memory logics with further ingredients from the language of hybrid logics. In particular, we will include nominals (atomic symbols which are true at a unique point in the relational structure) and the @ operator (which allows us to control the point of evaluation). As discussed in [8], the hybrid machinery can be used to prove general completeness results, and to axiomatize logics which are otherwise difficult to characterize.

The rest of the paper is organized as follows. In the next section we formally introduce the different logics we will investigate. In Section 3 we present a sound and complete axiomatization for $\mathcal{HL}(@, \text{ⓡ}, \text{ⓚ})$, the basic modal logic extended with nominals, @, and the ⓡ and ⓚ memory operators. In Section 4 we discuss completeness for languages including the ⓕ and ⓔ operators.

As it will be clear from the details that we present in the corresponding sections, nominals and @ play a crucial role in these axiomatic characterizations. Moreover, the axiomatizations we present are non-finite. On the other hand, the results are fairly general, as they characterize not only the base case where the languages are interpreted over the class of all possible models, but also when we restrict ourselves to different subclasses (with the proviso that these classes can be defined using 'pure' axioms).

In [3] we prove that the satisfiability problem of $\mathcal{ML}$ extended with ⓡ and ⓚ is undecidable, and we introduce a decidable logic including ⓡ and ⓚ (strictly more expressive than $\mathcal{ML}$) defining additional constrains on how the modal and the memory operators interact. We will show a sound and complete axiomatization for this logic in Section 5. Moreover, this axiomatization does not require the hybrid machinery.

We conclude in Section 6 with some final remarks.

---

[1] Our convention for naming logics is as follows. We call $\mathcal{ML}$ the basic modal logic, with the standard operators; we use $\mathcal{HL}$ for the modal language extended with only nominals; and we then list the additional operators included in the language. For example $\mathcal{HL}(@, \text{ⓡ}, \text{ⓔ}, \text{ⓚ})$ is the modal language extended with nominals, and the @, ⓡ, ⓚ and ⓔ operators.

## 2. Syntax and semantics of memory logics

In this section we formally introduce the languages mentioned above, together with some basic notation and notions related to completeness.

**Definition 1** (*Syntax*)**.** Let Prop $= \{p_1, p_2, \ldots\}$ (the *propositional symbols*), Nom $= \{n_1, n_2, \ldots\}$ (the *nominal symbols*) and Rel $= \{r_1, r_2, \ldots\}$ (the *relational symbols*) be pairwise disjoint, countable infinite sets. Let Atom $=$ Prop $\cup$ Nom. The set Forms of formulas in the signature $\langle$Prop, Nom, Rel$\rangle$ is defined as:

$$\text{Forms} ::= \top \mid p \mid i \mid ⓚ \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle r \rangle \varphi \mid @_i \varphi \mid ⓡ\varphi \mid ⓕ\varphi \mid ⓔ\varphi,$$

where $p \in$ Prop, $i \in$ Nom, $r \in$ Rel and $\varphi, \varphi_1, \varphi_2 \in$ Forms. We take $[r]\varphi$ as a shorthand for $\neg\langle r \rangle\neg\varphi$, and use the standard definitions for Boolean operators like $\vee, \rightarrow$, etc.

**Definition 2** (*Semantics*)**.** Given a signature $\mathcal{S} = \langle$Prop, Nom, Rel$\rangle$, a model for $\mathcal{S}$ is a tuple $\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle$, satisfying the following conditions: (i) $D \neq \emptyset$; (ii) each $R_r$ is a binary relation on $D$; (iii) $L :$ Atom $\rightarrow 2^D$ is a labeling function such that $L(n)$ is a singleton whenever $n \in$ Nom; and (iv) $S \subseteq D$.

Given the model $\mathcal{M} = \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle$ and $w \in D$, the semantics for the different operators is defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models \top & \text{always} & \\
\mathcal{M}, w \models p & \text{iff} & w \in L(p) \quad p \in \text{Atom} \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \langle r \rangle\varphi & \text{iff} & \text{there is } w' \text{ such that } R_r(w, w') \text{ and } \mathcal{M}, w' \models \varphi \\
\mathcal{M}, w \models @_i\varphi & \text{iff} & \mathcal{M}, v \models \varphi \text{ where } L(i) = \{v\} \\
\mathcal{M}, w \models ⓡ\varphi & \text{iff} & \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \cup \{w\} \rangle, w \models \varphi \\
\mathcal{M}, w \models ⓚ & \text{iff} & w \in S \\
\mathcal{M}, w \models ⓕ\varphi & \text{iff} & \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \backslash \{w\} \rangle, w \models \varphi \\
\mathcal{M}, w \models ⓔ\varphi & \text{iff} & \langle D, (R_r)_{r \in \mathsf{Rel}}, L, \emptyset \rangle, w \models \varphi.
\end{array}
$$

Given a model $\mathcal{M}$, we say that $\varphi$ is *valid on* $\mathcal{M}$ and write $\mathcal{M} \models \varphi$ if for all states $w$ in the domain of $\mathcal{M}$ we have that $\mathcal{M}, w \models \varphi$.

In the rest of the paper the following notation will be useful. Let $\mathcal{M} = \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle$ be a model and $w \in D$, then we define

$$
\begin{aligned}
\mathcal{M}[+w] &= \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \cup \{w\} \rangle \\
\mathcal{M}[-w] &= \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \backslash \{w\} \rangle.
\end{aligned}
$$

For $[o_1 w_1, \ldots, o_n w_n]$ a nonempty ordered list with $w_i \in D$ and $o_i \in \{+, -\}$, let $\mathcal{M}[o_1 w_1, \ldots, o_n w_n] = (\mathcal{M}[o_1 w_1])$ $[o_2 w_2, \ldots, o_n w_n]$, where $\mathcal{M}[\ ] = \mathcal{M}$. We will usually write $[w_1, \ldots, w_n]$ instead of $[+w_1, \ldots, +w_n]$.

**Definition 3** (*Satisfiability, Validity, Completeness*)**.** Let $\mathcal{C}$ be a class of models. We say that $\varphi$ is *satisfiable in* $\mathcal{C}$ if there is a model $\mathcal{M} \in \mathcal{C}$ and a state $w$ in the domain of $\mathcal{M}$ such that $\mathcal{M}, w \models \varphi$. We say that $\varphi$ is *valid in* $\mathcal{C}$ if $\neg\varphi$ is not satisfiable in $\mathcal{C}$. The notions of satisfiability and validity can be extended to sets of formulas in the usual way. For example, we say that a set of formulas $\Gamma$ is satisfiable in a class of models $\mathcal{C}$ if there is a model $\mathcal{M} \in \mathcal{C}$ and a state $w$ in the domain of $\mathcal{M}$ such that for all formulas $\varphi \in \Gamma$ we have $\mathcal{M}, w \models \varphi$. We will note $T(\mathcal{C})$ the set of all valid formulas in $\mathcal{C}$.

Given an axiomatization $\mathcal{A}$, a formula $\varphi$ is a *theorem of* $\mathcal{A}$ if it is an axiom in $\mathcal{A}$, or it can be obtained by a finite number of applications of inference rules in $\mathcal{A}$ from axioms of $\mathcal{A}$. We write $T(\mathcal{A})$ for the set of all theorems in $\mathcal{A}$.

We say that a formula $\varphi$ is *consistent* with respect to an axiomatization $\mathcal{A}$ (or $\mathcal{A}$-consistent) if $\neg\varphi$ is not a theorem of $\mathcal{A}$. The notion of consistency can be extended to a set of formulas $\Gamma$ by requiring that for no finite subset $\Gamma^f$, the formula $\bigwedge \Gamma^f \rightarrow \neg\top$ be a theorem of $\mathcal{A}$.

Given an axiomatization $\mathcal{A}$ and a class of models $\mathcal{C}$ we say that $\mathcal{A}$ is *sound for* $\mathcal{C}$ if $T(\mathcal{A}) \subseteq T(\mathcal{C})$, and that it is *complete for* $\mathcal{C}$ if $T(\mathcal{C}) \subseteq T(\mathcal{A})$. Completeness can be equivalently defined in terms of consistency and satisfiability: $\mathcal{A}$ is complete for $\mathcal{C}$ if every formula consistent in $\mathcal{A}$ is satisfiable in $\mathcal{C}$.

Finally, we say that an axiomatization $\mathcal{A}$ is *strongly complete* with respect to $\mathcal{C}$, if every $\mathcal{A}$-consistent set of formulas is satisfiable in $\mathcal{C}$.

In this article we will present a number of axiomatizations and prove them (strongly) complete with respect to different classes of models. The different logical languages involved will be defined in terms of the operators introduced in Definitions 1 and 2; and we will be interested mainly in the class of all models, and the class $\{\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle \mid S = \emptyset\}$ of models with no previously 'remembered' states. This last class is a natural choice: in the absence of the ⓔ operator, evaluating formulas on such models provides additional expressivity, and the intuitive meaning of the *remember* and *known* operators are naturally captured. For example the formula $ⓡ\langle r \rangleⓚ$ characterizes reflexivity of $R_r$ over this class (that is, let $\mathcal{M} = \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle$ be an arbitrary model, except that $S = \emptyset$, then $\mathcal{M} \models ⓡ\langle r \rangleⓚ$ if and only if $R_r$ is reflexive). This no longer holds when $S$ is arbitrary. See [3] for further details.

**Axioms**:

| | | | |
|---|---|---|---|
| *CT* | All classical tautologies | *Intro* | $\vdash (i \wedge p) \rightarrow @_i p$ |
| $K_@$ | $\vdash @_i(p \rightarrow q) \rightarrow @_i p \rightarrow @_i q$ | *Self-dual$_@$* | $\vdash @_i p \leftrightarrow \neg @_i \neg p$ |
| $K_{[r]}$ | $\vdash [r](p \rightarrow q) \rightarrow ([r]p \rightarrow [r]q)$ | *Ref* | $\vdash @_i i$ |
| *Sym* | $\vdash @_i j \leftrightarrow @_j i$ | *Nom* | $\vdash (@_i j \wedge @_j p) \rightarrow @_i p$ |
| *Agree* | $\vdash @_j @_i p \leftrightarrow @_i p$ | *Back* | $\vdash \langle r \rangle @_i p \rightarrow @_i p$ |
| *Rem* | $\vdash @_i(ⓡ\varphi \leftrightarrow \varphi[ⓚ/(ⓚ \vee i)])$ | | |

**Rules**:

| | | | |
|---|---|---|---|
| *MP* | If $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ then $\vdash \psi$ | *Gen$_{[r]}$* | If $\vdash \varphi$ then $\vdash [r]\varphi$ |
| *Name* | $\vdash j \rightarrow \varphi$ then $\vdash \varphi$ ($j$ not in $\varphi$) | *Gen$_@$* | If $\vdash \varphi$ then $\vdash @_i \varphi$ |
| *Paste* | If $\vdash (@_i \langle r \rangle j \wedge @_j \varphi) \rightarrow \psi$ then $\vdash (@_i \langle r \rangle \varphi) \rightarrow \psi$ | | |
| | ($j \neq i$ and $j$ is not in $\varphi$ or $\psi$) | | |
| *SortedSub$_1$* | If $\vdash \varphi$ then $\vdash \varphi[p/\psi]$ for any $p \in$ Prop | | |
| *SortedSub$_2$* | If $\vdash \varphi$ then $\vdash \varphi[i/j]$ for any $i, j \in$ Nom | | |

The expression $\varphi[a/b]$ is the result of uniformly replacing all occurrences of $a$ in $\varphi$ by $b$.

**Fig. 1.** Axiomatization for $\mathcal{HL}(@, ⓡ, ⓚ)$.

As we mentioned in the introduction, we will also be interested in a logic in which the behavior of the *remember* operator is highly coupled with the modal transitions to ensure decidability. In this logic, every time we make a modal step, we are constrained to remember the current state. We change the semantic definition of $\langle r \rangle$ to be:

$$\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \rangle, w \models \langle r \rangle \varphi \quad \text{iff} \quad \begin{array}{l} \exists w' \in D, R_r(w, w') \text{ and} \\ \langle D, (R_r)_{r \in \mathsf{Rel}}, L, S \cup \{w\} \rangle, w' \models \varphi. \end{array}$$

We call this logic $\mathcal{ML}^-$ ($\mathcal{HL}^-$ for the hybrid case). As we proved in [4], $\mathcal{ML}^-(ⓡ, ⓚ)$ is decidable and strictly more expressive than $\mathcal{ML}$.

## 3. Completeness for $\mathcal{HL}(@, ⓡ, ⓚ)$

This section is devoted to prove a completeness result for $\mathcal{HL}(@, ⓡ, ⓚ)$. Our axiomatization is shown in Fig. 1. It is an extension of the axiomatization for $\mathcal{HL}(@)$ presented in [7].

The axiom characterizing the behavior of the memory operator is *Rem*. To show soundness of the axiomatization, we only have to look at this new axiom. Intuitively, the axiom says that, when standing in a state named by $i$, the act of remembering the current state is equivalent to increase the extension of ⓚ with $i$ throughout the formula. Formally:

**Lemma 1.** *Let $\mathcal{M}$ be a model and $w \in \mathcal{M}$ such that $\mathcal{M}, w \models i$. Then, for all $v \in \mathcal{M}$, $\mathcal{M}[w], v \models \varphi$ iff $\mathcal{M}, v \models \varphi[ⓚ/(ⓚ \vee i)]$.*

**Proof.** By induction on $\varphi$. For the base case, if $\varphi$ is a proposition symbol or a nominal, then since $\varphi = \varphi[ⓚ/(ⓚ \vee i)]$ we have $\mathcal{M}[w], v \models \varphi$ iff $\mathcal{M}, v \models \varphi$. For the ⓚ case we have to prove $\mathcal{M}[w], v \models ⓚ$ iff $\mathcal{M}, v \models ⓚ \vee i$.

($\Rightarrow$) Assume that $\mathcal{M}[w], v \models ⓚ$. If $v = w$, then $\mathcal{M}, v \models i$, and therefore $\mathcal{M}, v \models ⓚ \vee i$. If $v \neq w$, then $\mathcal{M}, v \models ⓚ$, and hence $\mathcal{M}, v \models ⓚ \vee i$.

($\Leftarrow$) Let us assume that $\mathcal{M}, v \models ⓚ \vee i$. If $v = w$, then $\mathcal{M}[w], v \models ⓚ$. On the other hand, if $v \neq w$, then we know that $\mathcal{M}[w], v \models \neg i$, and therefore $\mathcal{M}, v \models ⓚ$. We conclude $\mathcal{M}[w], v \models ⓚ$.

The conjunction, negation, diamond, @ and remember cases are straightforward, using the inductive hypothesis and the fact that the replacement operation $[ⓚ/(ⓚ \vee i)]$ distributes over $\wedge, \neg, \langle r \rangle, @$ and ⓡ. $\square$

**Corollary 2.** *Rem is sound over the class of all models.*

**Proof.** Take an arbitrary model $\mathcal{M}$ and let $w \in \mathcal{M}$ be such that $\mathcal{M}, w \models i$. By definition $\mathcal{M}, v \models @_i ⓡ \varphi$ iff $\mathcal{M}[w], w \models \varphi$. Applying the previous lemma, this happens iff $\mathcal{M}, w \models \varphi[ⓚ/(ⓚ \vee i)]$ iff (by definition) $\mathcal{M}, v \models @_i \varphi[ⓚ/(ⓚ \vee i)]$. $\square$

It is worth noting that having nominals in the language is a key feature to describe the ⓡ/ⓚ interaction with modal operators, and the *Rem* axiom strongly uses this feature. The possibility to identify with a nominal the state in which a remember operation is taking place allows us to fully describe the behavior of this interaction.

We now turn to completeness. We will build a Henkin model using named maximal consistent sets (MCSs) for an arbitrary consistent set (see [7] for further details).

**Definition 4.** An MCS is *named* if and only if it contains a nominal. We call any nominal belonging to an MCS a *name* for that MCS. Also, if $\Gamma$ is an MCS and $i$ is a nominal, then we call $\{\varphi \mid @_i \varphi \in \Gamma\}$ a *named set yielded by $\Gamma$*. Furthermore we say that a model is *named* if every state in the model is the denotation of some nominal (for all $w \in D$ there is some nominal $i$ such that $L(i) = \{w\}$).

The idea behind the construction presented in [7] is that we can extract all the information we need to build a named canonical model from a single MCS. We start by noting that hidden inside any MCS there is a collection of named MCSs with a number of relevant properties.

**Lemma 3.** *Let $\Gamma$ be an MCS. For every nominal i, let $\Delta_i$ be $\{\varphi \mid @_i\varphi \in \Gamma\}$. Then,* (i) *for every nominal i, $\Delta_i$ is an MCS that contains i;* (ii) *for all nominals i and j, if $i \in \Delta_j$, then $\Delta_i = \Delta_j$;* (iii) *for all nominals i and j, $@_i\varphi \in \Delta_j$ iff $@_i\varphi \in \Gamma$; and* (iv) *if i is a name for $\Gamma$ then $\Gamma = \Delta_i$.*

**Proof.** We only sketch the proof, the full details can be found in [7]. Claim (i) can be proved using *Ref* (to guarantee that $i \in \Delta_i$), *Gen$_@$* and *Self-dual$_@$* (to prove that $\Delta_i$ is an MCS). Claim (ii) is proved using *Sym* and *Nom*, Claim (iii) follows by *Agree*. And Claim (iv) is obtained by *Intro* and *Self-dual$_@$*. □

Given a consistent set of formulas $\Sigma$, we can always expand it to an MCS $\Sigma^+$ using the standard Lindenbaum's Lemma. The problem is that nothing guarantees that this MCS will be named. In addition, as we want to extract named MCSs from named sets yielded by $\Sigma^+$, we have to ensure that there are enough named MCSs to use as existential witnesses during the construction of the Henkin model. Here is where the *Name* and *Paste* rules are useful. Expanding the language with new nominals, the *Name* rule is going to solve our first problem, and the *Paste* rule solves the second. We call an MCS $\Gamma$ *pasted* iff $@_i\langle r\rangle\varphi \in \Gamma$ implies that for some nominal $j$, $@_i\langle r\rangle j \wedge @_j\varphi \in \Gamma$. *Name* and *Paste* guarantee that any consistent set of formulas can be extended to a named and pasted MCS.

**Lemma 4** (*Extended Lindenbaum Lemma*). *Let $\mathscr{S} = \langle Prop, Nom, Rel\rangle$ be a signature, let Nom$'$ be a countably infinite collection of nominals disjoint from Nom, and let $\mathscr{S}'$ be the signature obtained by extending $\mathscr{S}$ with Nom$'$. Then every $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$-consistent set of formulas in S can be extended to a named and pasted MCS in $\mathscr{S}'$.*

**Proof.** Full details can be found in [7]. The proof follows the standard Lindenbaum's construction with the following modifications. Take a consistent set of formulas $\Sigma$, and name it by adding a new nominal $k$ (use *Name* to prove consistency). Using an enumeration of all the formulas, we expand $\Sigma$ step-by-step with a formula that is consistent with the expanded set at each point. Because we want the final MCS to be pasted, at the $(m + 1)$-th step, when we are considering $\Sigma^m$ and the formula $\varphi_{m+1}$, if $\Sigma^m \cup \{\varphi_{m+1}\}$ is inconsistent, we set $\Sigma^{m+1} = \Sigma^m$. Else, if $\varphi_{m+1}$ has the form $@_i\langle r\rangle\varphi$, we set $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\} \cup \{@_i\langle r\rangle j \wedge @_j\langle r\rangle\varphi\}$, where $j$ is new (relying on the *Paste* rule for consistency). If $\varphi_{m+1}$ does not have the form $@_i\langle r\rangle\varphi$, we set $\Sigma^{m+1} = \Sigma^m \cup \{\varphi_{m+1}\}$ as usual. Finally, we take the infinite union of all the $\Sigma^i$. □

Now we can define the model we need, using the named sets yielded by a named and pasted MCS.

**Definition 5.** Let $\Gamma$ be a named and pasted MCS. The *named model yielded by* $\Gamma$ is $\mathcal{M}^\Gamma = (D^\Gamma, (R_r^\Gamma)_{r\in\mathsf{Rel}}, L^\Gamma, S^\Gamma)$. Here $D^\Gamma$ is the set of all named sets yielded by $\Gamma$, $R_r^\Gamma(u, v)$ holds iff for all formulas $\varphi$, $\varphi \in v$ implies $\langle r\rangle\varphi \in u$, $L^\Gamma(a) = \{w \in W^\Gamma \mid a \in w\}$ for any atom $a$, and $S^\Gamma = \{w \mid \textcircled{k} \in w\}$.

Note that $\mathcal{M}^\Gamma$ is a well defined model, since by items (i) and (ii) of Lemma 3, $L^\Gamma$ assigns to every nominal a singleton subset of $D^\Gamma$. Using the fact that $\Gamma$ is named and pasted, we can prove the following Existence Lemma.

**Lemma 5** (*Existence Lemma [7]*). *Let $\Gamma$ be a named and pasted MCS, and let $\mathcal{M} = \langle D, (R_r)_{r\in\mathsf{Rel}}, L, S\rangle$ be the named model yielded by $\Gamma$. Suppose $u \in \mathcal{M}$ and $\langle r\rangle\varphi \in u$. Then there is a $v \in \mathcal{M}$ such that $R_r(u, v)$ and $\varphi \in v$.*

Now we are ready to prove the Truth Lemma that will lead us to the desired completeness result. Before that, to treat the $\textcircled{r}$ case properly, we have to redefine the complexity of the formulas, to be able to handle the substitutions made by the *Rem* axiom.

**Definition 6.** We define the complexity of a formula as $comp(\varphi) = 2(k + 1)(r + 1)(d + 1) + v$, where $k$, $r$ and $d$ are the number of occurrences of $\textcircled{k}$, $\textcircled{r}$ and $\langle r\rangle$ respectively, and $v$ is the number of occurrences of all the other possible operators.

Note that with this definition, $comp(\textcircled{r}\varphi) > comp(\varphi[\textcircled{k}/(\textcircled{k} \vee i)])$.

**Lemma 6** (*Truth Lemma*). *Let $\mathcal{M} = \langle D, (R_r)_{r\in\mathsf{Rel}}, L, S\rangle$ be the named model yielded by a named and pasted MCS, and let $u \in D$. Then, for all formulas $\varphi$, $\varphi \in u$ iff $\mathcal{M}, u \models \varphi$.*

**Proof.** By Induction on the complexity *comp* of $\varphi$. The atomic, Boolean and modal cases are obvious (the Existence Lemma is used for the modal case, and the $\textcircled{k}$ case follows directly from the definition of $S^\Gamma$). We analyze the satisfaction operators. Suppose $\mathcal{M}, u \models @_i\psi$. This happens iff $\mathcal{M}, \Delta_i \models \psi$ (by items (i) and (ii) of Lemma 3, $\Delta_i$ is the only MCS containing $i$, and hence, by the atomic case of the present lemma, the only state in $\mathcal{M}$ where $i$ is true) iff $\psi \in \Delta_i$ (by inductive hypothesis) iff $@_i\psi \in \Delta_i$ (using the fact that $i \in \Delta_i$ together with *Intro* for the left-to-right direction and *Intro* and *Self-dual$_@$* for the right-to-left direction) iff $@_i\psi \in u$ (by *Agree*).

To finish let us analyze the case for $\textcircled{r}$. Given $u \in \mathcal{M}$, we know that for some nominal $i$ we have $u = \Delta_i$, so by definition, $\mathcal{M}, u \models i$ and $i \in u$. Suppose $\mathcal{M}, u \models \textcircled{r}\psi$. This happens iff $\mathcal{M}, u \models @_i\textcircled{r}\psi$ (because $\mathcal{M}, u \models i$) iff $\mathcal{M}, u \models @_i\psi[\textcircled{k}/(\textcircled{k} \vee i)]$ (by Corollary 2) iff $\mathcal{M}, u \models \psi[\textcircled{k}/(\textcircled{k} \vee i)]$ (again because $\mathcal{M}, u \models i$) iff $\psi[\textcircled{k}/(\textcircled{k} \vee i)] \in u$ (by inductive hypothesis) iff $@_i\psi[\textcircled{k}/(\textcircled{k} \vee i)] \in u$ (because $i \in u$, using *Intro* for the left-to-right direction, and *Self-dual$_@$* and *Intro* for the right-to-left direction) iff $@_i\textcircled{r}\psi \in u$ (by the *Rem* axiom) iff $\textcircled{r}\psi \in u$ (because $i \in u$, applying again *Intro* and *Self-dual$_@$*). □

**Theorem 7** (*Completeness for $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$*)**.** *Every MCS in $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$ is satisfiable in a countable named model.*

**Proof.** Let $\Sigma$ be a consistent set of formulas from $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$. We use the Extended Lindenbaum Lemma to expand it to a named and pasted set $\Sigma^+$ in an extended countable language. Let $\mathcal{M}$ be the named model yielded by $\Sigma^+$. By item (iv) of Lemma 3, because $\Sigma^+$ is named, $\Sigma^+$ is an element in the domain of $\mathcal{M}$. By the Truth Lemma, $\mathcal{M}, \Sigma^+ \models \Sigma$. The model is countable because each state is named by some nominal in the extended language, and there are only countably many of these. □

This establishes strong completeness as desired. But in fact, we have done more. Because our Henkin model is *named*, we can prove a more general result.

**Definition 7.** If a formula $\varphi$ contains no propositional symbols (that is, its atoms are nominals or $\textcircled{k}$), we say that $\varphi$ is $\textcircled{k}$-*pure*. Furthermore, if $\varphi$ is a $\textcircled{k}$-pure formula, we say that $\psi$ is a $\textcircled{k}$-*pure instance* of $\varphi$ if $\psi$ is obtained from $\varphi$ by uniformly substituting nominals for nominals. A formula $\varphi$ is *pure* if its atomic subformulas are only nominals.

The axiomatization we presented in Fig. 1 for $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$ has the following property: for any set of pure formulas $\Pi$, if $P$ is the logic obtained by adding the formulas in $\Pi$ as axioms, then $P$ is complete with respect to the class defined by $\Pi$.[2] This result can be extended to $\textcircled{k}$-pure formulas for the case of $\mathcal{HL}_\emptyset(@, \textcircled{r}, \textcircled{k})$, the logic obtained over the class $\{\langle D, (R_r)_{r \in \mathsf{Rel}}, L, S\rangle \mid S = \emptyset\}$ of models with no previously remembered states.

We first state a property that will help us achieve the completeness result for pure axioms.

**Lemma 8.** *Let $\mathcal{M} = \langle D, (R_r)_{r \in \mathbf{Rel}}, L, S\rangle$ be a named model.*

1. *Let $\varphi$ be a pure formula, and suppose that for all pure instances $\psi$ of $\varphi$, $\mathcal{M} \models \psi$. Then for any $L'$ and $S'$, $\langle D, (R_r)_{r \in \mathbf{Rel}}, L', S'\rangle \models \varphi$.*
2. *Let $S = \emptyset$ and $\varphi$ be a $\textcircled{k}$-pure formula. Suppose that for all $\textcircled{k}$-pure instances $\psi$ of $\varphi$, $\mathcal{M} \models \psi$. Then for any $L'$, $\langle D, (R_r)_{r \in \mathbf{Rel}}, L', S\rangle \models \varphi$.*

**Proof.** We only discuss item 2. Suppose that the hypothesis holds, but for some labeling $L'$, $\langle D, (R_r)_{r \in \mathbf{Rel}}, L', \emptyset\rangle \not\models \varphi$. We can take $\rho$, a $\textcircled{k}$-pure instance of $\varphi$, such that $\rho$ is obtained from $\varphi$ replacing each nominal $i$ by $j$, where $L'(i) = L(j)$. By an induction on the formula complexity, it is easy to see that $(D, (R_r)_{r \in \mathbf{Rel}}, L, \emptyset) \not\models \rho$. This is a contradiction. □

With the help of Lemma 8, and since we showed that we can build named models from $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$-MCSs, a wide range of strong completeness results can be established (with the same proof as the one given in [7]).

**Theorem 9.** *Let $\Pi$ be a set of pure formulas and let $\mathcal{A}$ be the axiomatization obtained by adding formulas in $\Pi$ as axioms to the axiomatization shown in Fig. 1. Then, every $\mathcal{A}$-consistent set of formulas is satisfiable in a countable named model in the class defined by $\Pi$.*

**Proof.** Given a $\mathcal{A}$-consistent set of formulas $\Omega$, we can use the Extended Lindenbaum's Lemma to extend it to a named and pasted $\mathcal{A}$-MCS $\Omega^+$. The named model $\mathcal{M}^\Omega$ that $\Omega^+$ gives rise to will satisfy $\Omega$ at $\Omega^+$. In addition, as every formula in $\Pi$ belongs to every $\mathcal{A}$-MCS, we have that $\mathcal{M}^\Omega \models \Pi$. Therefore, by Lemma 8, $\mathcal{M}^\Omega$ is in the class of models defined by $\Pi$. □

To finish this section we will discuss an extension of the axiomatization presented above to characterize $\mathcal{HL}_\emptyset(@, \textcircled{r}, \textcircled{k})$.

**Theorem 10.** *The system obtained by extending the axiomatization in Fig. 1 with the axiom (Empty) $\vdash \neg\textcircled{k}$ is sound and strongly complete for $\mathcal{HL}_\emptyset(@, \textcircled{r}, \textcircled{k})$.*

**Proof.** Soundness of *Empty* is obvious for the class of $\mathcal{HL}_\emptyset(@, \textcircled{r}, \textcircled{k})$-models. The completeness proof is as the one for $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$, but in addition, thanks to *Empty*, all maximal consistent sets $\Delta_i$ are such that $\neg\textcircled{k} \in \Delta_i$. Therefore, the final model yielded by $\Gamma$, $\mathcal{M}^\Gamma = \langle D^\Gamma, (R_r^\Gamma)_{r \in \mathbf{Rel}}, L^\Gamma, S^\Gamma\rangle$, is such that $S^\Gamma = \emptyset$, and thus it is a $\mathcal{HL}_\emptyset(@, \textcircled{r}, \textcircled{k})$-model. □

**Proposition 1.** *For the case of $\mathcal{HL}_\emptyset(@, \textcircled{r}, \textcircled{k})$, the result of adding $\Pi$, a set of pure formulas, can be extended to a set $\Pi$ of $\textcircled{k}$-pure formulas.*

**Proof.** Trivial, using Lemma 8, and the same proof as in Theorem 9. □

## 4. Completeness for the erase and forget operators

We now turn to languages containing the $\textcircled{e}$ and $\textcircled{f}$ operators. We will first discuss completeness for $\mathcal{HL}(@, \textcircled{r}, \textcircled{e}, \textcircled{k})$, then for $\mathcal{HL}(@, \textcircled{r}, \textcircled{f}, \textcircled{k})$, and finally for the language $\mathcal{HL}(@, \textcircled{r}, \textcircled{e}, \textcircled{f}, \textcircled{k})$.

### 4.1. Axiomatizing $\mathcal{HL}(@, \textcircled{r}, \textcircled{e}, \textcircled{k})$

We take as a starting point the axiomatization for $\mathcal{HL}(@, \textcircled{r}, \textcircled{k})$ presented in Fig. 1. The first thing we should notice is that the *Rem* axiom is no longer sound. For example, take the valid formula $@_i\textcircled{e}(\textcircled{k} \vee i)$ and use *Rem* to conclude $@_i\textcircled{r}\textcircled{e}\textcircled{k}$.

---

[2] These general completeness results are standard in hybrid logics (see [8]).

**Axioms**:
All the axioms from $\mathcal{HL}(@, \circledr, \circledk)$ except *Rem*
*Rem'*    $\vdash @_i(\circledr\varphi \leftrightarrow \varphi_i^\star)$
*Erase$_1$*    $\vdash \circlede\neg\circledk$
*Erase$_2$*    $\vdash \circlede s \leftrightarrow s \quad s \in \mathsf{Prop} \cup \mathsf{Nom}$
*Erase$_3$*    $\vdash \circlede\neg p \leftrightarrow \neg\circlede p$
*Erase$_4$*    $\vdash \circlede(p \wedge q) \leftrightarrow (\circlede p \wedge \circlede q)$
*Erase$_5$*    $\vdash \circlede\langle r\rangle p \leftrightarrow \langle r\rangle\circlede p$
*Erase$_6$*    $\vdash \circlede @_i p \leftrightarrow @_i\circlede p$
*Erase$_7$*    $\vdash @_i(\circlede\circledr\varphi \leftrightarrow \circlede\varphi_i^\star)$
**Rules**:
All the rules from $\mathcal{HL}(@, \circledr, \circledk)$

**Fig. 2.** Axiomatization for $\mathcal{HL}(@, \circledr, \circlede, \circledk)$.

This is clearly a contradiction, since after wiping out the memory, $\circledk$ cannot be true. Observe that the problem lies in the interaction between $\circledr$ and $\circlede$. The replacement operation defined by *Rem* cannot be carried out throughout the whole formula: it should avoid replacements within the scope of an $\circlede$. More formally, for each formula $\varphi$ and nominal $i$ we define the formula $\varphi_i^\star$ as follows:

$$p_i^\star = p \quad p \in \mathsf{Prop} \cup \mathsf{Nom}$$
$$\circledk_i^\star = \circledk \vee i$$
$$(\neg\varphi)_i^\star = \neg\varphi_i^\star$$
$$(\varphi_1 \wedge \varphi_2)_i^\star = \varphi_{1i}^\star \wedge \varphi_{2i}^\star$$
$$(\circledr\varphi)_i^\star = \circledr\varphi_i^\star$$
$$(\langle r\rangle\varphi)_i^\star = \langle r\rangle\varphi_i^\star$$
$$(@_j\varphi)_i^\star = @_j\varphi_i^\star$$
$$(\circlede\varphi)_i^\star = \circlede\varphi$$

Analogously to Lemma 1, we can use $(\cdot)^\star$ to characterize the behavior of the $\circledr$ operator and its interaction with the $\circlede$ operator.

**Lemma 11.** *Let $\mathcal{M}$ be a model and $w \in \mathcal{M}$ such that $\mathcal{M}, w \models i$. Then $\mathcal{M}, w \models \circledr\varphi$ iff $\mathcal{M}, w \models \varphi_i^\star$.*

This result naturally suggests an axiom *Rem'* (shown in Fig. 2) that replaces *Rem*. To characterize the $\circlede$ operator we should notice first that it behaves globally and that it does not change the evaluation point. This implies that there is no interaction between $\circlede$ and $\neg$, $\wedge$, $\langle r\rangle$ and $@$. To describe the interaction between $\circlede$ and $\circledr$ we can again make use of the operation $(\cdot)^\star$. The detailed axiomatization is in Fig. 2.

Soundness of this axiomatization is straightforward. The completeness proof uses the same techniques introduced in Section 3. The proof of the Truth Lemma is carried out by induction in the complexity of the formula, and the new axioms handle the case for $\circlede$ by appropriately reducing the complexity in order to use the inductive hypothesis, as it is done in Lemma 6.

So now we can give the strong completeness result for $\mathcal{HL}(@, \circledr, \circlede, \circledk)$. The proof of this theorem follows exactly the same technique used in Theorem 7.

**Theorem 12** (*Completeness for $\mathcal{HL}(@, \circledr, \circlede, \circledk)$*)**.** *Every MCS of formulas in $\mathcal{HL}(@, \circledr, \circlede, \circledk)$ is satisfiable in a countable named model.*

Since it is clear that Lemma 8 still holds in $\mathcal{HL}(@, \circledr, \circlede, \circledk)$, and the canonical model we built is named, it is easy to see that one can also establish a stronger completeness result in terms of pure formulas for $\mathcal{HL}(@, \circledr, \circlede, \circledk)$, in the same way as stated in Theorems 9 and 10.

**Theorem 13** (*Completeness for $\mathcal{HL}(@, \circledr, \circlede, \circledk)$*)**.** *Let $\Pi$ be a set of pure formulas and let $\mathcal{A}$ be the axiomatization obtained by adding formulas in $\Pi$ as axioms to the axiomatization shown in Fig. 2. Then, every $\mathcal{A}$-consistent set of formulas is satisfiable in a countable named model in the class defined by $\Pi$.*

*4.2. Axiomatizing $\mathcal{HL}(@, \circledr, \circledf, \circledk)$*

Let us consider an axiomatization for $\mathcal{HL}(@, \circledr, \circledf, \circledk)$. The main complication, compared with the case we just discussed, is that the $\circledf$ operator has a *local* behavior, and clearly depends on the point of evaluation. Hence, describing its interaction with the $\circledr$ operator will be more involved. We will require two rewriting functions $(\cdot)^r$ and $(\cdot)^f$. Using these two functions, we can obtain a very simple axiomatization of $\mathcal{HL}(@, \circledr, \circledf, \circledk)$ (see Fig. 3).

---

**Axioms**:
All the axioms from $\mathcal{HL}(@)$
*Rem*    $\vdash @_i(\textcircled{r}\varphi \leftrightarrow \varphi_i^r)$
*Forg*   $\vdash @_i(\textcircled{f}\varphi \leftrightarrow \varphi_i^f)$
**Rules**:
All the rules from $\mathcal{HL}(@)$

---

Fig. 3. Axiomatization for $\mathcal{HL}(@, \textcircled{r}, \textcircled{f}, \textcircled{k})$.

For each formula $\varphi \in \mathcal{HL}(@, \textcircled{r}, \textcircled{f}, \textcircled{k})$ and nominal $i$, we define the formula $\varphi_i^r$ as follows:

$$p_i^r = p \quad p \in \mathsf{Prop} \cup \mathsf{Nom}$$
$$\textcircled{k}_i^r = (\textcircled{k} \vee i)$$
$$(\neg\varphi)_i^r = \neg\varphi_i^r$$
$$(\varphi_1 \wedge \varphi_2)_i^r = (\varphi_{1i}^r \wedge \varphi_{2i}^r)$$
$$(\textcircled{r}\varphi)_i^r = \textcircled{r}\varphi_i^r$$
$$(\langle r\rangle\varphi)_i^r = \langle r\rangle\varphi_i^r$$
$$(@_j\varphi)_i^r = @_j\varphi_i^r$$
$$(\textcircled{f}\varphi)_i^r = \textcircled{f}((i \to \varphi) \wedge (\neg i \to \varphi_i^r))$$

**Lemma 14.** *For every pointed model $(\mathcal{M}, w)$ such that $\mathcal{M}, w \models i$, and for all $v \in \mathcal{M}$, $\mathcal{M}[+w], v \models \varphi$ iff $\mathcal{M}, v \models \varphi_i^r$.*

**Proof.** By induction on $\varphi$. For the base case, if $\varphi$ is a proposition symbol or a nominal, say $a$, then since $a_i^r = a$ we have $\mathcal{M}[+w], v \models a$ iff $\mathcal{M}, v \models a$. For the $\textcircled{k}$ case we have to prove $\mathcal{M}[+w], v \models \textcircled{k}$ iff $\mathcal{M}, v \models \textcircled{k} \vee i$.

($\Rightarrow$) Assume that $\mathcal{M}[+w], v \models \textcircled{k}$. If $v = w$, then $\mathcal{M}, v \models i$, and therefore $\mathcal{M}, v \models \textcircled{k} \vee i$. If $v \neq w$, then $\mathcal{M}, v \models \textcircled{k}$, and hence $\mathcal{M}, v \models \textcircled{k} \vee i$.

($\Leftarrow$) Let us assume that $\mathcal{M}, v \models \textcircled{k} \vee i$. If $v = w$, then $\mathcal{M}[+w], v \models \textcircled{k}$. On the other hand, if $v \neq w$, then we know that $\mathcal{M}[+w], v \models \neg i$, and therefore $\mathcal{M}, v \models \textcircled{k}$. We conclude $\mathcal{M}[+w], v \models \textcircled{k}$.

Let us analyze the $\varphi = \textcircled{f}\psi$ case. Suppose that $v = w$, therefore $\mathcal{M}[+w], w \models \textcircled{f}\psi$ iff $\mathcal{M}[+w, -w], w \models \psi$ iff $\mathcal{M}[-w], w \models \psi$ iff (because $\mathcal{M}[-w], w \models i$) $\mathcal{M}[-w], w \models (i \to \psi) \wedge (\neg i \to \psi_i^r)$ iff (by definition of $\textcircled{f}$) $\mathcal{M}, w \models \textcircled{f}((i \to \psi) \wedge (\neg i \to \psi_i^r))$. On the other hand, suppose $w \neq v$. Therefore, $\mathcal{M}[+w], v \models \textcircled{f}\psi$ iff $\mathcal{M}[+w, -v], v \models \psi$ iff (because $v$ and $w$ are different states) $\mathcal{M}[-v, +w, ], v \models \psi$ iff (by inductive hypothesis) $\mathcal{M}[-v], v \models \psi_i^r$ iff (because $\mathcal{M}[-v], v \models \neg i$) $\mathcal{M}[-v], v \models (i \to \psi) \wedge (\neg i \to \psi_i^r)$ iff (by definition of $\textcircled{f}$) $\mathcal{M}, v \models \textcircled{f}((i \to \psi) \wedge (\neg i \to \psi_i^r))$.

The conjunction, negation, diamond, @ and remember cases are straightforward, using the inductive hypothesis and the fact that the translation from $\varphi$ to $\varphi_i^r$ distributes over $\wedge, \neg, \langle r\rangle$, @ and $\textcircled{r}$. $\square$

**Corollary 15.** *Let $\mathcal{M}$ by a model, and $w \in \mathcal{M}$. Then $\mathcal{M}, w \models @_i(\textcircled{r}\varphi \leftrightarrow \varphi_i^r)$.*

In the same way, we can define a formula $\varphi_i^f$ to deal with the $\textcircled{f}$ case:

$$p_i^f = p \quad p \in \mathsf{Prop} \cup \mathsf{Nom}$$
$$\textcircled{k}_i^f = (\textcircled{k} \wedge \neg i)$$
$$(\neg\varphi)_i^f = \neg\varphi_i^f$$
$$(\varphi_1 \wedge \varphi_2)_i^f = (\varphi_{1i}^f \wedge \varphi_{2i}^f)$$
$$(\textcircled{f}\varphi)_i^f = \textcircled{f}\varphi_i^f$$
$$(\langle r\rangle\varphi)_i^f = \langle r\rangle\varphi_i^f$$
$$(@_j\varphi)_i^f = @_j\varphi_i^f$$
$$(\textcircled{r}\varphi)_i^f = \textcircled{r}((i \to \varphi) \wedge (\neg i \to \varphi_i^f))$$

**Lemma 16.** *For every pointed model $(\mathcal{M}, w)$ such that $\mathcal{M}, w \models i$, and for all $v \in \mathcal{M}$, $\mathcal{M}[-w], v \models \varphi$ iff $\mathcal{M}, v \models \varphi_i^f$.*

**Proof.** By induction on $\varphi$. The only cases that are worth analyzing are $\textcircled{k}$ and $\textcircled{r}\psi$. The other cases are equivalent to the proof of Lemma 14. For the $\textcircled{k}$ case we have to prove $\mathcal{M}[-w], v \models \textcircled{k}$ iff $\mathcal{M}, v \models \textcircled{k} \wedge \neg i$.

($\Rightarrow$) Assume that $\mathcal{M}[-w], v \models \textcircled{k}$. If $v = w$, this is absurd, so $v \neq w$. Therefore, $\mathcal{M}, v \models \textcircled{k}$, and hence $\mathcal{M}, v \models \textcircled{k} \wedge \neg i$.
($\Leftarrow$) Let us assume that $\mathcal{M}, v \models \textcircled{k} \wedge \neg i$. If $v = w$, then $\mathcal{M}, v \models i$, so this is absurd. Therefore $v \neq w$, and then we know that $\mathcal{M}[-w], v \models \textcircled{k}$, and therefore $\mathcal{M}[-w], v \models \textcircled{k} \wedge \neg i$.

Let us analyze the $\varphi = \textcircled{r}\psi$ case. Suppose that $v = w$, therefore $\mathcal{M}[-w], w \models \textcircled{r}\psi$ iff $\mathcal{M}[-w, +w], w \models \psi$ iff $\mathcal{M}[+w], w \models \psi$ iff (because $\mathcal{M}[+w], w \models i$) $\mathcal{M}[+w], w \models (i \to \psi) \wedge (\neg i \to \psi_i^f)$ iff (by definition of $\textcircled{r}$)

$\mathcal{M}, w \models \text{ⓡ}((i \to \psi) \wedge (\neg i \to \psi_i^f))$. On the other hand, suppose $w \neq v$. Therefore, $\mathcal{M}[-w], v \models \text{ⓡ}\psi$ iff $\mathcal{M}[-w, +v], v \models \psi$ iff (because $v$ and $w$ are different states) $\mathcal{M}[+v, -w], v \models \psi$ iff (by inductive hypothesis) $\mathcal{M}[+v], v \models \psi_i^f$ iff (because $\mathcal{M}[+v], v \models \neg i$) $\mathcal{M}[+v], v \models (i \to \psi) \wedge (\neg i \to \psi_i^f)$ iff (by definition of ⓡ) $\mathcal{M}, v \models \text{ⓡ}((i \to \psi) \wedge (\neg i \to \psi_i^f))$. □

**Corollary 17.** *Let $\mathcal{M}$ be a model, and $w \in \mathcal{M}$. Then $\mathcal{M}, w \models @_i(\text{ⓕ}\varphi \leftrightarrow \varphi_i^f)$.*

Soundness of *Rem* and *Forg* in the axiomatization of $\mathcal{HL}(@, \text{ⓡ}, \text{ⓕ}, \text{Ⓚ})$ in Fig. 3 are a direct consequence of Corollaries 15 and 17.

To achieve completeness, we first have to give an adequate notion of complexity of formulas in such a way that the Truth Lemma for this logic can be shown. As in Section 3, we look for a function *comp* : Forms $\to \mathbb{N}$ such that $comp(\text{ⓡ}\varphi) > comp(\varphi[\text{Ⓚ}/(\text{Ⓚ} \vee i)])$. But in this setting, to account for the new axioms of Fig. 3, we have stronger restrictions: we need to find a function such that $comp(\text{ⓡ}\varphi) > comp(\varphi_i^r)$ and $comp(\text{ⓕ}\varphi) > comp(\varphi_i^f)$. The complexity given in Definition 6 is not suitable because the lengths of $\varphi_i^r$ and $\varphi_i^f$ are much larger than the length of $\varphi$. We next show some upper bounds for the lengths of $\varphi_i^r$ and $\varphi_i^f$ and then we define a suitable complexity function.

Observe that some right-hand formulas in the definition of $\varphi_i^r$ and $\varphi_i^f$ are abbreviations of formulas using $\wedge$ and $\neg$ as the only boolean connectives. Having this in mind, it can easily be shown the following equalities concerning $|\varphi|$, the length of a formula $\varphi$:

$$|(\text{ⓕ}\varphi)_i^r| = 15 + |\varphi| + |\varphi_r^r|$$
$$|(\text{ⓡ}\varphi)_i^f| = 15 + |\varphi| + |\varphi_i^f|$$
$$|(\varphi \wedge \psi)_i^*| = 3 + |\varphi_i^*| + |\psi_i^*| \quad \text{for } * \in \{r, f\}$$
$$|(\dagger\varphi)_i^*| = 1 + |\varphi_i^*| \quad \text{for } \dagger \in \{\neg, \langle r \rangle, @_j\} \text{ and } * \in \{r, f\}$$
$$|\text{Ⓚ}_i^r| = 8$$
$$|\text{Ⓚ}_i^f| = 6$$

It can be shown by induction in $\varphi$ that $\max\{|\varphi_i^r|, |\varphi_i^f|\} \leq (|\varphi| + 7)^2$. Let $n_r(\varphi)$ denote the nesting depth of ⓡ in the formula $\varphi$, i.e. the maximum number of occurrences of ⓡ along the paths of the syntactic tree of $\varphi$. In the same way, let $n_f(\varphi)$ denote the nesting depth of ⓕ in $\varphi$. Observe that $n_r(\varphi) = n_r(\varphi_i^r)$ and $n_f(\varphi) = n_f(\varphi_i^f)$.

Let $c(\varphi)$ : Forms $\to \mathbb{R}$ be defined as

$$c(\varphi) = 2^{3(n_r(\varphi) + n_f(\varphi))} \cdot \log |\varphi|.$$

The reader may verify that $c(\text{ⓡ}\varphi) > c(\varphi_i^r)$ and $c(\text{ⓕ}\varphi) > c(\varphi_i^f)$. Furthermore, for all the subformulas $\psi$ of a formula $\varphi$, $c(\psi)$ is strictly increasing in $|\psi|$. Therefore, *comp* : Forms $\to \mathbb{N}$ defined as

$$comp(\varphi) = 2^{c(\varphi)} = |\varphi|^{2^{3(n_r(\varphi) + n_f(\varphi))}}$$

is a suitable complexity function.

With the complexity function properly defined, strong completeness follows using the same techniques introduced in Section 3. As in the case for $\mathcal{HL}(@, \text{ⓡ}, \text{ⓔ}, \text{Ⓚ})$, it is easy to see that the result holds for any pure axiomatic extension.

**Theorem 18** (*Completeness for $\mathcal{HL}(@, \text{ⓡ}, \text{ⓕ}, \text{Ⓚ})$*). *Let $\Pi$ be a set of pure formulas and let $\mathcal{A}$ be the axiomatization obtained by adding formulas in $\Pi$ as axioms to the axiomatization shown in Fig. 3. Then, every $\mathcal{A}$-consistent set of formulas is satisfiable in a countable named model in the class defined by $\Pi$.*

*4.3. Axiomatizing $\mathcal{HL}(@, \text{ⓡ}, \text{ⓔ}, \text{ⓕ}, \text{Ⓚ})$*

Finally, putting together the ideas from the previous two axiomatizations we obtain a sound and complete axiomatization for $\mathcal{HL}(@, \text{ⓡ}, \text{ⓔ}, \text{ⓕ}, \text{Ⓚ})$. The first step is to extend the definition of $\varphi_i^r$ and $\varphi_i^f$ to handle the case of ⓔ:

$$(\text{ⓔ}\varphi)_i^r = \text{ⓔ}\varphi$$
$$(\text{ⓔ}\varphi)_i^f = \text{ⓔ}\varphi$$

Note that Lemmas 14 and 16 still hold. Now we only need to add the axioms we used to characterize ⓔ with minor changes. Observe that the complexity function defined in Section 4.2 is appropriate for this case also. The final axiomatization is shown in Fig. 4.

**Theorem 19** (*Completeness for $\mathcal{HL}(@, \text{ⓡ}, \text{ⓔ}, \text{ⓕ}, \text{Ⓚ})$*). *Let $\Pi$ be a set of pure formulas and let $\mathcal{A}$ be the axiomatization obtained by adding formulas in $\Pi$ as axioms to the axiomatization shown in Fig. 4. Then, every $\mathcal{A}$-consistent set of formulas is satisfiable in a countable named model in the class defined by $\Pi$.*

**Axioms**:
All the axioms from $\mathcal{HL}(@)$
$Rem$ $\quad\vdash @_i(\text{ⓡ}\varphi \leftrightarrow \varphi_i^r)$
$Forg$ $\quad\vdash @_i(\text{ⓕ}\varphi \leftrightarrow \varphi_i^f)$
$Erase_1$ $\quad\vdash \text{ⓔ}\neg\text{ⓚ}$
$Erase_2$ $\quad\vdash \text{ⓔ}s \leftrightarrow s \quad s \in \mathsf{Prop} \cup \mathsf{Nom}$
$Erase_3$ $\quad\vdash \text{ⓔ}\neg p \leftrightarrow \neg\text{ⓔ}p$
$Erase_4$ $\quad\vdash \text{ⓔ}(p \wedge q) \leftrightarrow (\text{ⓔ}p \wedge \text{ⓔ}q)$
$Erase_5$ $\quad\vdash \text{ⓔ}\langle r\rangle p \leftrightarrow \langle r\rangle\text{ⓔ}p$
$Erase_6$ $\quad\vdash \text{ⓔ}@_i p \leftrightarrow @_i\text{ⓔ}p$
$Erase_7'$ $\quad\vdash @_i(\text{ⓔ}\text{ⓡ}\varphi \leftrightarrow \text{ⓔ}\varphi_i^r)$
$Erase_8$ $\quad\vdash \text{ⓔ}\text{ⓕ}\varphi \leftrightarrow \text{ⓔ}\varphi$
**Rules**:
All the rules from $\mathcal{HL}(@)$

Fig. 4. Axiomatization for $\mathcal{HL}(@, \text{ⓡ}, \text{ⓔ}, \text{ⓕ}, \text{ⓚ})$.

**Axioms**:
$CT$ $\quad$ All classical tautologies
$K_{[r]}$ $\quad\vdash [r](p \to q) \to ([r]p \to [r]q)$
$Rem^-$ $\quad\vdash \text{ⓡ}\varphi \leftrightarrow \varphi^\sharp$
**Rules**:
$MP$ $\quad$ If $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$
$Gen_{[r]}$ $\quad$ If $\vdash \varphi$ then $\vdash [r]\varphi$
$Sub$ $\quad$ If $\vdash \varphi$ then $\vdash \varphi[p/\psi]$ for any $p \in \mathsf{Prop}$

Fig. 5. Axiomatization for $\mathcal{ML}^-(\text{ⓡ}, \text{ⓚ})$.

## 5. The case for $\mathcal{ML}^-(\text{ⓡ}, \text{ⓚ})$

The last logic that we are going to discuss is $\mathcal{ML}^-(\text{ⓡ}, \text{ⓚ})$. In the previous section we mentioned the importance of nominals to describe the interaction between memory and modal operators. In this section we will show that if we restrict ourselves to the logic in which we are constrained to remember the current state every time we make a modal transition, it is possible to define a sound and complete axiomatization where nominals can be avoided. The key ingredient to do this is that in this logic we can describe the interaction between $\text{ⓡ}$ and $\text{ⓚ}$ at a propositional level. This is not a coincidence. Because this logic has the tree model property [3,4], we can assume that we evaluate $\mathcal{ML}^-(\text{ⓡ}, \text{ⓚ})$-formulas on trees, and since there are no cycles, the remember operator has no real effect beyond the current state.

Given a formula $\varphi$, we define the formula $\varphi^\sharp$ as the result of replacing all the occurrences of $\text{ⓚ}$ that are in $\varphi$ at modal depth zero by $\top$. Formally:

$$p^\sharp = p \quad p \in \mathsf{Prop}$$
$$\text{ⓚ}^\sharp = \top$$
$$(\neg\varphi)^\sharp = \neg\varphi^\sharp$$
$$(\varphi_1 \wedge \varphi_2)^\sharp = \varphi_1^\sharp \wedge \varphi_2^\sharp$$
$$(\text{ⓡ}\varphi)^\sharp = \text{ⓡ}\varphi^\sharp$$
$$(\langle r\rangle\varphi)^\sharp = \langle r\rangle\varphi$$

**Lemma 20.** $\mathcal{M}, w \models \text{ⓡ}\varphi$ iff $\mathcal{M}, w \models \varphi^\sharp$.

**Proof.** We proceed by induction. The case for $\text{ⓚ}$, the propositional symbols and Boolean connectives are straightforward. We analyze the other cases. For the case $\varphi = \text{ⓡ}\psi$. $\mathcal{M}, w \models \text{ⓡ}\text{ⓡ}\psi$ iff $\mathcal{M}, w \models \text{ⓡ}\psi$ iff (by inductive hypothesis) $\mathcal{M}, w \models \psi^\sharp$ iff $\mathcal{M}, w \models (\psi^\sharp)^\sharp$ iff (by inductive hypothesis) $\mathcal{M}, w \models \text{ⓡ}(\psi^\sharp)$ iff $\mathcal{M}, w \models (\text{ⓡ}\psi)^\sharp$. For the case $\varphi = \langle r\rangle\psi$. $\mathcal{M}, w \models \text{ⓡ}\langle r\rangle\psi$ iff (by definition) $\mathcal{M}[w], w \models \langle r\rangle\psi$ iff (by definition of $\langle r\rangle$) there is a $v \in \mathcal{M}, R_r(w, v)$ such that $\mathcal{M}[w], v \models \psi$ iff (by definition of $\langle r\rangle$) $\mathcal{M}, w \models \langle r\rangle\psi$ iff (by definition of $\sharp$) $\mathcal{M}, w \models (\langle r\rangle\psi)^\sharp$. $\square$

The axiomatization for $\mathcal{ML}^-(\text{ⓡ}, \text{ⓚ})$ (see Fig. 5) is an extension of the axiomatization for the basic modal logic [7], plus the axiom $Rem^- \vdash \text{ⓡ}\varphi \leftrightarrow \varphi^\sharp$.

Soundness of $Rem^-$ follows from Lemma 20. We will prove completeness with respect to the class of acyclic models, and therefore for the class of all models. We will use a step-by-step construction. I.e., instead of building the entire canonical model, we will carry out a stepwise selection from MCSs of the canonical model of $\mathcal{ML}^-(\text{ⓡ}, \text{ⓚ})$ as our basic building blocks.[3]

---

[3] Alternatively, one can take the standard canonical model and then unravel it to obtain a tree, and therefore acyclic, model.

We define $\mathcal{M}^c = \langle D^c, (R_r^c)_{r\in\mathsf{Rel}}, L^c, S^c \rangle$, the $\mathcal{ML}^-(\textcircled{r}, \textcircled{k})$ canonical model, in the usual sense (see [7] for details). That is, $D^c$ is the set of all maximal consistent sets of formulas of $\mathcal{ML}^-(\textcircled{r}, \textcircled{k})$, $R_r^c(\Gamma, \Delta)$ iff for all $\varphi \in \Delta$, $\langle r\rangle\varphi \in \Gamma$, $\Gamma \in L^c(p)$ iff $p \in \Gamma$ and $S^c = \{\Gamma \mid \textcircled{k} \in \Gamma\}$.

**Definition 8.** A network $\mathcal{N} = \langle N, (R_r)_{r\in\mathsf{Rel}}, v\rangle$ is a triple where $N$ is a countable non-empty set of elements, each $R_r$ is a binary relation on $N$, and $v$ is a function that maps elements in $N$ to maximal consistent sets.

We say that a network is *coherent* if (C1) $\bigcup_{r\in\mathsf{Rel}} R_r$ defines an acyclic graph and (C2) $R_r^c(v(s), v(t))$ for all $s, t \in N$ such that $R_r(s, t)$. A network is *saturated* if whenever $\langle r\rangle\psi \in v(s)$ for some $s \in N$, then there is a $t \in N$ such that $R_r(s, t)$ and $\psi \in v(t)$.

We want networks to play the role of models, so we have to check that we have imposed the right conditions on a network to achieve this.

**Definition 9.** Let $\mathcal{N} = \langle N, (R_r)_{r\in\mathsf{Rel}}, v\rangle$ be a network. We define the induced labeling $L_\mathcal{N}(p) = \{s \in N \mid p \in v(s)\}$, the induced set of remembered states $S_\mathcal{N} = \{s \in N \mid \textcircled{k} \in v(s)\}$, and the induced model $\mathcal{M}_\mathcal{N} = \langle N, (R_r)_{r\in\mathsf{Rel}}, L_\mathcal{N}, S_\mathcal{N}\rangle$. $\mathcal{F}_\mathcal{N} = \langle N, (R_r)_{r\in\mathsf{Rel}}\rangle$ is called the *underlying frame* of $\mathcal{N}$.

We are now ready to prove a Truth Lemma.

**Lemma 21** (*Truth Lemma*). *Let $\mathcal{N} = \langle N, (R_r)_{r\in\mathbf{Rel}}, v\rangle$ be a coherent and saturated network. Then, for all $\varphi$ and $s \in N$,*

$$\mathcal{M}_\mathcal{N}, s \models \varphi \text{ iff } \varphi \in v(s).$$

**Proof.** Before we prove this lemma, let us observe the following property: let $\mathcal{M} = \langle D, (R_r)_{r\in\mathsf{Rel}}, L, S\rangle$ be an acyclic model, and let $w, v \in D$ be such that $R_r(w, v)$. Then for all formulas $\varphi$, $\mathcal{M}[w], v \models \varphi$ iff $\mathcal{M}, v \models \varphi$.

We now proceed by induction on $\varphi$. The propositional case, the $\textcircled{k}$ case and the Boolean cases are straightforward, given the definition of $\mathcal{M}_\mathcal{N}$. Let us suppose that $\mathcal{M}_\mathcal{N}, s \models \textcircled{r}\psi$. This happens iff (by Lemma 20) $\mathcal{M}_\mathcal{N}, s \models \psi^\sharp$ iff (by the inductive hypothesis) $\psi^\sharp \in v(s)$ iff (by $Rem^-$ axiom) $\textcircled{r}\psi \in v(s)$.

The $\langle r\rangle$ case: for the left-to-right direction, if $\mathcal{M}_\mathcal{N}, s \models \langle r\rangle\psi$, then there exists $t \in N$ such that $R_r(s, t)$ and $\mathcal{M}_\mathcal{N}[s], t \models \psi$. Therefore, $\mathcal{M}_\mathcal{N}, t \models \psi$. By inductive hypothesis, $\psi \in v(t)$. Because the network is coherent, and $R_r(s, t)$, then $R_r^c(v(s), v(t))$, and we conclude $\langle r\rangle\psi \in v(s)$. For the other direction, let us suppose that $\langle r\rangle\psi \in v(s)$. Because the network is saturated, there is a $t \in N$ such that $\psi \in v(t)$ and $R_r(s, t)$. By the inductive hypothesis, $\mathcal{M}_\mathcal{N}, t \models \psi$, so $\mathcal{M}_\mathcal{N}[s], t \models \psi$, and therefore by definition, $\mathcal{M}_\mathcal{N}, s \models \langle r\rangle\psi$. □

Summing up, we have reduced the problem of finding a model for an MCS $\Delta$ to a search for a coherent and saturated network for $\Delta$. The idea here is to start with a coherent network and, one step at a time, remove the defects that are preventing the network from being saturated.

**Definition 10.** Let $\mathcal{N}$ be a network. We say that $\mathcal{N}$ has a saturation defect if there is a node $s \in N$ and a formula $\langle r\rangle\psi \in v(s)$ such that there is no $t \in N$ such that $R(s, t)$ and $\psi \in v(t)$.

Because a coherent network may have saturation defects, we have to say more about what is the meaning of repairing a defect. We are going to *extend* a network with a saturation defect with another where the defect is corrected.

**Definition 11.** Let $\mathcal{N}_0 = \langle N_0, R_0, v_0\rangle$ and $\mathcal{N}_1 = \langle N_1, R_1, v_1\rangle$ be two networks. We say that $\mathcal{N}_1$ extends $\mathcal{N}_0$ if $\mathcal{F}_{\mathcal{N}_0}$ is a subframe of $\mathcal{F}_{\mathcal{N}_1}$ and $v_0$ agrees with $v_1$ on $N_0$.

The following lemma states that a saturation defect of a finite coherent network can always be repaired.

**Lemma 22** (*Repair Lemma*). *Let $\mathcal{N}$ be a finite and coherent network with a saturation defect. Then there is a network $\mathcal{N}'$ extending $\mathcal{N}$ without that defect.*

**Proof.** Because $\mathcal{N}$ has a a saturation defect, there is a node $s \in N$ and a formula $\langle r\rangle\psi \in v(s)$ such that there is no $t \in N$, with $R_r(s, t)$ and $\psi \in v(t)$. We define $\mathcal{N}'$ as

$$N' = N \cup \{s'\} \quad \text{with } s' \notin N$$
$$R_r' = R_r \cup \{(s, s')\}$$
$$v' = v \cup \{(s', \Delta)\}$$

where $\Delta$ is an MCS containing $\psi$ such that $R_r^c(v(s), \Delta)$ (the existence of such $\Delta$ can be proved through an Existence Lemma similar to Lemma 5). Clearly, $\mathcal{N}'$ is a coherent network extending $\mathcal{N}$ and does not have the previous defect. □

Now we can prove the desired strong completeness result. We start with a singleton network, and we extend it step by step to a larger network using the Repair Lemma. We obtain the saturated network we want by taking the union of our sequence of networks.

**Theorem 23.** *The axiomatization is strongly complete with respect to the class of $\mathcal{ML}^-(\textcircled{r}, \textcircled{k})$ models.*

**Proof.** Let $S = \{s_i \mid i \in \omega\}$. Enumerate the potential saturation defects (the set $S \times$ Forms). Given a consistent set $\Sigma$, expand it to an MCS $\Sigma^+$. The initial network is $\mathcal{N}^0 = \langle \{s_0\}, \emptyset, (s_0, \Sigma^+) \rangle$, which is finite and coherent. Given a network $\mathcal{N}^i, i \geq 0$, where the minimal saturation defect is $D$, we define $\mathcal{N}^{i+1}$ as the extension of $\mathcal{N}^i$ (following the Repair Lemma) without that defect. If $\mathcal{N}^i$ has no saturation defects, then $\mathcal{N}^{i+1} = \mathcal{N}^i$. Let $\mathcal{N}^\omega = \langle N, (R_r)_{r \in \mathsf{Rel}}, v \rangle$ be:

$$N = \bigcup_{n \in \omega} N^n \qquad R_r = \bigcup_{n \in \omega} R_r^n \qquad v = \bigcup_{n \in \omega} v^n.$$

It is clear that $\mathcal{N}^\omega$ is saturated. Suppose this is not so; let $d$ be the minimal saturation defect (with respect to the enumeration) of $\mathcal{N}^\omega$, say $d = d_k$. By construction, there must be an approximation $\mathcal{N}^i$ of $\mathcal{N}^\omega$ of which $d$ is also a defect. There only can be $k$ defects that are less than $d$, so $d$ will be repaired before the stage $k + i$ of the construction. This is a contradiction, so $\mathcal{N}^\omega$ is a coherent and saturated network, and therefore $\mathcal{M}_{\mathcal{N}^\omega}, s_0 \models \Sigma$. □

## 6. Conclusions

In this paper we presented several axiomatizations for some members of the memory logic family. We showed how nominals can be an effective tool to achieve completeness: by allowing to describe the precise interaction between ⓡ and ⓚ we could give a completeness result for $\mathcal{HL}(@, ⓡ, ⓚ)$. Small variations of this axiomatization leads us to completeness results for other languages, as we showed for $\mathcal{HL}_\emptyset(@, ⓡ, ⓚ)$ and $\mathcal{HL}(@, ⓡ, ⓔ, ⓚ)$.

Our intention was to give the basic techniques to characterize memory operators using nominals, and not to exhaustively list all possible languages. Observe that, for example, the logic $\mathcal{HL}^-(@, ⓡ, ⓚ)$ can be easily axiomatized by replacing the *Back* axiom presented in Fig. 1 by $\vdash @_i \langle r \rangle @_j \varphi \rightarrow @_j \varphi[ⓚ/(ⓚ \vee i)]$ (and similarly with the *Paste* rule).

We also showed that nominals are not needed when we add constraints on how $\langle r \rangle$ interacts with ⓡ, giving a completeness result for $\mathcal{ML}^-(ⓡ, ⓚ)$. The idea behind this result lies in the fact that $\mathcal{ML}^-(ⓡ, ⓚ)$ has the tree model property and hence, we can describe the interaction between ⓡ and ⓚ at a propositional level, independently of the modal operators.

We have not yet found suitable axiomatizations for certain memory logics. Languages without the tree model property, and which do not have nominals seem to be particularly hard to axiomatize. For example, we have not yet been able to devise complete axiomatizations for $\mathcal{ML}(ⓡ, ⓚ)$ and $\mathcal{ML}_\emptyset^-(ⓡ, ⓚ)$.

## Acknowledgement

## References

[1] R. Alur, T. Henzinger, A really temporal logic, in: Journal of the ACM, IEEE Computer Society Press, 1989, pp. 164–169.
[2] C. Areces, Hybrid logics: the old and the new, in: Proceedings of LogKCA-07, San Sebastian, Spain, 2007, pp. 15–29.
[3] C. Areces, D. Figueira, S. Figueira, S. Mera, Expressive power and decidability for memory logics, in: Logic, Language, Information and Computation, in: LNCS, vol. 5110, Springer, Berlin, Heidelberg, 2008, pp. 56–68.
[4] C. Areces, D. Figueira, S. Figueira, S. Mera, The expressive power of memory logics, The Review of Symbolic Logic 4 (2) (2011) 290–318.
[5] C. Areces, B. ten Cate, Hybrid logics, in: Blackburn et al. [9], pp. 821–868.
[6] P. Blackburn, Representation, reasoning, and relational structures: a hybrid logic manifesto, Logic Journal of the IGPL 8 (3) (2000) 339–625.
[7] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, 2001.
[8] P. Blackburn, M. Tzakova, Hybrid completeness, Logic Journal of the IGPL 6 (4) (1998) 625–650.
[9] P. Blackburn, F. Wolter, J. van Benthem (Eds.), Handbook of Modal Logics, Elsevier, 2006.
[10] J. Gerbrandy, Bisimulations on planet Kripke, Ph.D. Thesis, University of Amsterdam, iLLC Dissertation series DS-1999-01, 1999.
[11] E. Harel, O. Lichtenstein, A. Pnueli, Explicit clock temporal logic, in: Proceedings of LICS'90, 1990, pp. 402–413.
[12] T. Henzinger, Half-order modal logic: how to prove real-time properties, in: Proceedings of the Ninth Annual Symposium on Principles of Distributed Computing, ACM Press, 1990, pp. 281–296.
[13] C. Lutz, The complexity of reasoning with concrete domains, Ph.D. Thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2002.
[14] J. Plaza, Logics of public communications, in: 4th International Symposium on Methodologies for Intelligent Systems, 1989, pp. 201–216.
[15] J. van Benthem, Logics for information update, in: TARK'01: Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge, Morgan Kaufmann Publishers Inc., 2001, pp. 51–67.
[16] J. van Benthem, J. van Eijck, B. Kooi, Logics of communication and change, Information and Computation 204 (11) (2006) 1620–1662.
[17] H. van Ditmarsch, W. van der Hoek, B. Kooi, Dynamic Epistemic Logic, Kluwer academic publishers, 2007.