Hybrid many-sorted modal logic with nominal terms

Bogdan Macovei

(joint work with Ioana Leuștean)

Faculty of Mathematics and Computer Science, University of Bucharest

Research Center for Logic, Optimization and Security

April 8, 2026
Będlewo, Poland

Content

Motivation

- Formalizing security protocols in Lean was one of my PhD thesis subjects. Our goal was to develop a logical system that: (1) has good theoretical logical properties (2) defines protocol execution by operational semantics (3) allows us to reason about agents' knowledge and beliefs.
- In order to benefit from the advantages of a general theory, we chose the many-sorted modal hybrid logic, previously used for defining a programming language and its operational semantics.
- In our examples we identified a common pattern: the nominals are used to define the universe of our problem, and the actual modal formulas are used to reason about the system's properties. The idea of adding structure on nominals such that we can represent our universe as a term algebra was natural in this context.

Motivation: from hybrid logic to nominal terms

Hybrid logic extends modal logic by adding *nominals*, i.e. atomic formulas that are true at exactly one world.

In standard hybrid logic, if r is a nominal and φ is a formula, then

$$@_r\varphi$$

means that φ is true at the world named by r.

Motivation: from hybrid logic to nominal terms

Hybrid logic extends modal logic by adding *nominals*, i.e. atomic formulas that are true at exactly one world.

In standard hybrid logic, if r is a nominal and φ is a formula, then

$$@_r\varphi$$

means that φ is true at the world named by r.

Question. What if the set of worlds has an algebraic structures?

Motivation: from hybrid logic to nominal terms

Hybrid logic extends modal logic by adding *nominals*, i.e. atomic formulas that are true at exactly one world.

In standard hybrid logic, if r is a nominal and φ is a formula, then

$$@_r\varphi$$

means that φ is true at the world named by r.

Question. What if the set of worlds has an algebraic structures?
The worlds are named not only by atomic nominals, but also by terms.

Motivation: from hybrid logic to nominal terms

Main ideas.

- worlds/states should admit algebraic structure;
- nominals should be generalized to *nominal terms*.

Main contributions

The paper has three main layers.

(1) Logical layer. We define a many-sorted hybrid modal logic

$$\mathcal{H}_{\Sigma\mathfrak{T}}(@)$$

based on a modal signature (S, Σ) and an algebraic signature $(S, \mathfrak{T})$.

Main contributions

The paper has three main layers.

(1) Logical layer. We define a many-sorted hybrid modal logic

$$\mathcal{H}_{\Sigma\mathfrak{T}}(@)$$

based on a modal signature (S, Σ) and an algebraic signature $(S, \mathfrak{T})$.

(2) Algebraic layer. We prove:

- completeness with respect to suitable classes of $\Sigma\mathfrak{T}$-frames;
- algebraic completeness with respect to suitable $\Sigma\mathfrak{T}$-structures.

Main contributions

The paper has three main layers.

(1) Logical layer. We define a many-sorted hybrid modal logic

$$\mathcal{H}_{\Sigma\mathfrak{T}}(@)$$

based on a modal signature (S, Σ) and an algebraic signature $(S, \mathfrak{T})$.

(2) Algebraic layer. We prove:

- completeness with respect to suitable classes of $\Sigma\mathfrak{T}$-frames;
- algebraic completeness with respect to suitable $\Sigma\mathfrak{T}$-structures.

(3) Application layer. We instantiate the framework as an epistemic logic with actions for specifying and analysing security protocols.

Implementation aspect. The protocol logic is implemented in Lean 4, and a concrete model can be generated automatically from a protocol description.

System $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$: overview

- Syntax is richer than in ordinary hybrid logic because *terms themselves are formulas*.
- Frame semantics uses singleton interpretation for terms.
- Algebraic semantics is defines by pairs "term algebra + Boolean algebra with operators" connected by an injective map.

Many-sorted modal algebraic signature

A *many-sorted modal algebraic signature* is a triple

$$(S, \Sigma, \mathfrak{T})$$

where:

- S is a nonempty set of sorts;
- (S, Σ) is a many-sorted modal signature;
- $(S, \mathfrak{T})$ is an algebraic S-sorted signature.

We also fix disjoint countable S-sorted sets

$$\mathrm{PROP} = \{\mathrm{PROP}_s\}_{s \in S}, \qquad \mathrm{NOM} = \{\mathrm{NOM}_s\}_{s \in S}.$$

Intuition.

- elements of PROP are the ordinary propositional variables;
- elements of NOM are the primitive nominals;
- operations in $\mathfrak{T}$ build *nominal terms*, by increasingly complex names for states;
- operations in Σ are the modal operators.

Syntax: terms and formulas

For every sort $s \in S$, the syntax is:

$$t_s ::= r \mid f(t_{s_1}, \ldots, t_{s_n})$$

$$\varphi_s ::= p \mid t_s \mid \neg\varphi_s \mid \varphi_s \to \varphi_s \mid \sigma(\varphi_{s_1}, \ldots, \varphi_{s_n}) \mid @^s_{t_{s'}}\varphi_{s'}$$

where

$$r \in \mathrm{NOM}_s, \quad p \in \mathrm{PROP}_s, \quad f \in \mathfrak{T}_{s_1 \cdots s_n, s}, \quad \sigma \in \Sigma_{s_1 \cdots s_n, s}.$$

1. Terms are built inductively exactly as in an ordinary many-sorted term algebra.
2. A term t_s is also admitted as a *formula of sort s*. This is essential because terms behave semantically like nominals: they are always evaluated in singleton sets.

For a modal operator $\sigma \in \Sigma_{s_1 \cdots s_n, s}$, its dual is defined by

$$\sigma^{\Box}(\varphi_1, \ldots, \varphi_n) := \neg\sigma(\neg\varphi_1, \ldots, \neg\varphi_n).$$

Sets of terms and formulas

For every sort $s \in S$, denote:

$$\mathrm{FORM}_s = \{\text{all formulas of sort } s\}, \qquad \mathrm{TERM}_s = \{\text{all terms of sort } s\}.$$

Hence we obtain two S-sorted families:

$$\mathrm{FORM} = \{\mathrm{FORM}_s\}_{s \in S}, \qquad \mathrm{TERM} = \{\mathrm{TERM}_s\}_{s \in S}.$$

Moreover, the term signature naturally induces the many-sorted term algebra generated by nominals:

$$(T_{\mathfrak{T}}(\mathrm{NOM}), \{F_f^T\}_{f \in \mathfrak{T}})$$

where, for each operation symbol $f \in \mathfrak{T}_{s_1 \cdots s_n, s}$,

$$F_f^T(\tau_1, \ldots, \tau_n) = f(\tau_1, \ldots, \tau_n).$$

Frame semantics

A $\Sigma\mathfrak{T}$-*frame* is a tuple

$$\mathcal{F} = (W, \{R_\sigma\}_{\sigma\in\Sigma}, \{F_f\}_{f\in\mathfrak{T}})$$

where:

- $(W, \{R_\sigma\}_{\sigma\in\Sigma})$ is a many-sorted modal frame;
- $(W, \{F_f\}_{f\in\mathfrak{T}})$ is a many-sorted algebra.

A $\Sigma\mathfrak{T}$-*model* is a pair

$$\mathcal{M} = (\mathcal{F}, V)$$

where $V : \mathrm{PROP} \cup \mathrm{TERM} \to \mathcal{P}(W)$ is an S-sorted valuation satisfying:

$$\begin{array}{ll} \text{(v1)} & V_s(t) \text{ is a singleton for every } t \in \mathrm{TERM}_s; \\ \text{(v2)} & V_s(f(t_1, \ldots, t_n)) = \{F_f(V_{s_1}(t_1), \ldots, V_{s_n}(t_n))\}. \end{array}$$

So, semantically, terms continue to behave like nominals, but now they do so *via algebraic evaluation*.

Satisfaction relation

For a model $\mathcal{M} = (\mathcal{F}, V)$, the satisfaction relation is defined by:

$$
\begin{aligned}
\mathcal{M}, w \models^{s} p &\iff w \in V_s(p), \qquad p \in \mathrm{PROP}_s, \\
\mathcal{M}, w \models^{s} t &\iff V_s(t) = \{w\}, \qquad t \in \mathrm{TERM}_s, \\
\mathcal{M}, w \models^{s} \neg\varphi &\iff \mathcal{M}, w \not\models^{s} \varphi, \\
\mathcal{M}, w \models^{s} \varphi \to \psi &\iff (\mathcal{M}, w \models^{s} \varphi \text{ implies } \mathcal{M}, w \models^{s} \psi), \\
\mathcal{M}, w \models^{s} \sigma(\varphi_1, \ldots, \varphi_n) &\iff \exists(w_1, \ldots, w_n)\,(R_\sigma w w_1 \ldots w_n \wedge \bigwedge_{i=1}^{n} \mathcal{M}, w_i \models^{s_i} \varphi_i), \\
\mathcal{M}, w \models^{s} @_t^s \varphi &\iff \mathcal{M}, u \models^{s'} \varphi \text{ where } V_{s'}(t) = \{u\}.
\end{aligned}
$$

The formula $@_t^s\varphi$ is evaluated at a world of sort s, but it asserts that φ holds at the world named by t in sort s'. $@$ is a mechanism that transports local truth to a chosen named state.

Pure formulas and named models

A formula is *pure* if it contains no propositional variables.

A model

$$\mathcal{M} = (W, \{R_\sigma\}_{\sigma \in \Sigma}, \{F_f\}_{f \in \mathfrak{T}}, V)$$

is *named* if, for every sort $s \in S$ and every world $w \in W_s$, there exists a term $t \in \mathrm{TERM}_s$ such that

$$V_s(t) = \{w\}.$$

Pure formulas proposition. If $\mathcal{M} = (\mathcal{F}, V)$ is named and φ is pure of sort s, then

$$\mathcal{F} \models^s \varphi \quad \Longleftrightarrow \quad \mathcal{M} \models^s \psi$$

for every pure instance ψ of φ obtained by uniformly replacing nominals with nominal terms of the same sort.

Deductive system $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$ I

Propositional basis. Every propositional tautology in sort s is an axiom of sort s.

Modal axioms.

$$(K_\sigma)\ \sigma^{\Box}(\dots,\varphi_{i-1},\varphi\to\psi,\varphi_{i+1},\dots)\\ \to(\sigma^{\Box}(\dots,\varphi_{i-1},\varphi,\varphi_{i+1},\dots)\to\sigma^{\Box}(\dots,\varphi_{i-1},\psi,\varphi_{i+1},\dots)),$$

$$(Dual_\sigma)\ \sigma(\varphi_1,\dots,\varphi_n)\leftrightarrow\neg\sigma^{\Box}(\neg\varphi_1,\dots,\neg\varphi_n).$$

Deductive system $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$ II

Hybrid axioms.

$$
\begin{aligned}
(K@)\ & @_t^s(\varphi \to \psi) \to (@_t^s\varphi \to @_t^s\psi),\\
(Agree)\ & @_t^s @_{t'}^{s'}\varphi \leftrightarrow @_{t'}^s\varphi,\\
(SDual)\ & @_t^s\varphi \leftrightarrow \neg @_t^s \neg\varphi,\\
(Intro)\ & t \to (\varphi \leftrightarrow @_t^s\varphi),\\
(Back)\ & \sigma(\dots,\varphi_{i-1}, @_t^{s_i}\psi, \varphi_{i+1},\dots)_s \to @_t^s\psi,\\
(Ref)\ & @_t^s t.
\end{aligned}
$$

Deductive system $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$ III

Rules.

$$(MP): \quad \frac{\vdash^s \varphi \qquad \vdash^s (\varphi \to \psi)}{\vdash^s \psi},$$

$$(UG): \quad \frac{\vdash^{s_i} \varphi}{\vdash^s \sigma^{\Box}(\varphi_1, \ldots, \varphi, \ldots, \varphi_n)},$$

$(Subst)$: uniform substitution of formulas for propositional variables and terms for nominals,

$$(Bcast_S): \quad \frac{\vdash^s @_t^s \varphi}{\vdash^{s'} @_t^{s'} \varphi},$$

$$(Gen@): \quad \frac{\vdash^{s'} \varphi}{\vdash^s @_t^s \varphi},$$

$$(TRepl): \quad \frac{\vdash^s @_t^s t'}{\vdash^s @^s_{f(\ldots,t,\ldots)} f(\ldots, t', \ldots)}.$$

Unorthodox rules and their role

Name@ rule

$$\frac{\vdash^{s} @^{s}_{r}\varphi}{\vdash^{s'}\varphi}$$ where r is a nominal not occurring in φ.

Paste rule

$$\frac{\vdash^{s} (@^{s}_{t}\sigma(\dots, r, \dots) \wedge @^{s}_{r}\varphi \to \psi)}{\vdash^{s} (@^{s}_{t}\sigma(\dots, \varphi, \dots) \to \psi)}$$

where r is a fresh nominal.

Interpretation.

- *Name@* says that if a formula holds at a fresh named point, we may discharge the name.
- *Paste* is the usual hybrid mechanism allowing us to decompose modal accessibility through a freshly named witness.

Basic derived facts

Among the useful derived formulas are:

$$
\begin{aligned}
(\mathit{Nom})\ & @_t^s t' \to (@_t^s \varphi \leftrightarrow @_{t'}^s \varphi),\\
(\mathit{Sym})\ & @_t^s t' \to @_{t'}^{s'} t,\\
(\mathit{Bridge})\ & \sigma(\dots, \varphi_{i-1}, t, \varphi_{i+1}, \dots) \wedge @_t^s \varphi\\
& \qquad \to \sigma(\dots, \varphi_{i-1}, \varphi, \varphi_{i+1}, \dots).
\end{aligned}
$$

If t_1, t_2 are terms of the same sort, one may define an equational notation by

$$t_1 \overset{s}{\approx} t_2 \ := \ @_{t_1}^s t_2.$$

Then the usual equational rules become available on the term side:

- reflexivity from (Ref),
- symmetry from (Sym),
- transitivity from (Nom),
- replacement from (TRepl),
- substitution from (Subst).

Named and pasted sets. Henkin model

A consistent set $\Gamma \subseteq \mathrm{FORM}_s$ is called *named* if one of its elements is a nominal term, and *pasted* if every formula of the form

$$@_t^s \sigma(\dots, \varphi, \dots)$$

can be accompanied by a fresh nominal witness r with

$$@_t^s \sigma(\dots, r, \dots) \in \Gamma \qquad \text{and} \qquad @_r^s \varphi \in \Gamma.$$

Extended Lindenbaum lemma. Every Λ-consistent set can be extended to a named, pasted, maximal Λ-consistent set after adding countably many nominals.

From such a set Γ, the Henkin model is built as follows:

$$W_{s'}^{\Gamma} = \{|t| : t \in \mathrm{TERM}_{s'}\} \qquad \text{where} \qquad |t| = \{t' \in \mathrm{TERM}_{s'} : @_t^s t' \in \Gamma\}.$$

Relations and operations are defined canonically by membership in Γ:

$$(|t|, |t_1|, \dots, |t_n|) \in R_\sigma^{\Gamma} \iff @_t^s \sigma(t_1, \dots, t_n) \in \Gamma,$$
$$F_f^{\Gamma}(|t_1|, \dots, |t_n|) := |f(t_1, \dots, t_n)|.$$

Truth lemma and completeness

Truth lemma. For every sort $s' \in S$, every term $t \in \mathrm{TERM}_{s'}$ and every formula $\varphi \in \mathrm{FORM}_{s'}$,

$$\mathcal{M}_\Gamma, |t| \models^{s'} \varphi \qquad \Longleftrightarrow \qquad @_t^s \varphi \in \Gamma.$$

Moreover, if $t_0 \in \Gamma \cap \mathrm{TERM}$, then

$$\mathcal{M}_\Gamma, |t_0| \models^{s} \Gamma.$$

Main logical completeness theorem.

1. $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$ is sound and complete with respect to the class of all $\Sigma\mathfrak{T}$-frames.
2. If Λ is an S-sorted set of pure formulas and $s \in S$, then any Λ-consistent set $\Gamma \subseteq \mathrm{FORM}_s$ has a countable named model based on a $\Sigma\mathfrak{T}$-frame validating all formulas from Λ.

Why an algebraic semantics?

The logical semantics is given by frames and models. The next step is to identify the appropriate algebraic structures corresponding to the logic.

For ordinary modal logic, the classical algebraic semantics is given by Boolean algebras with operators (BAO's).

For hybrid logic, one typically enriches BAO's with designated atoms or additional operators. In the present setting, however, we need something richer because:

- terms are not only atomic names, but elements of a many-sorted algebra;
- formulas still live on the Boolean/modal side;
- the satisfaction operator must connect the term side to the formula side.

Therefore the right algebraic object is not just a single algebra, but a *tuple* combining:

$$\text{a term algebra} \quad + \quad \text{a Boolean algebra with operators} \quad + \quad \text{an interaction map } @.$$

Definition of a $\Sigma\mathfrak{T}$-structure

A $\Sigma\mathfrak{T}$-structure is a tuple

$$\mathcal{AB} = (A, B, f, \{@^{s,s'}\}_{s,s' \in S})$$

such that:

- $A = (A, \{f^A\}_f)$ is a $\mathfrak{T}$-algebra;
- $B = (B, \vee, \neg, \bot_B, \{m^B_\sigma\}_{\sigma \in \Sigma})$ is a Σ-BAO;
- $f : A \to B$ is an injective S-sorted map satisfying:

$$f_s(a) \neq \bot^B_s, \qquad f_s(a) \wedge f_s(a') = \bot^B_s \text{ whenever } a \neq a';$$

- for every $s, s' \in S$, there is a map

$$@^{s,s'} : A_{s'} \times B_{s'} \to B_s.$$

Thus A stores the algebra of terms, B stores the algebra of formulas.

Algebraic axioms for @

Writing $@^s_a b$ instead of $@^{s,s'}(a, b)$ when $a \in A_{s'}$ and $b \in B_{s'}$, the following axioms are imposed:

$$
\begin{aligned}
(K@)\ & @^s_a(\neg b_1 \vee b_2) \leq \neg @^s_a b_1 \vee @^s_a b_2, \\
(Agree \leq)\ & @^{s,s'}_{a_1} @^{s',s''}_{a_2} b \leq @^{s,s''}_{a_2} b, \\
(Ref)\ & @^{s,s'}_a f_{s'}(a) = \top^B_s, \\
(SDual)\ & \neg @^s_a b = @^s_a \neg b, \\
(Intro \leq)\ & f_s(a) \wedge b \leq @^{s,s}_a b, \\
(Gen@)\ & @^{s,s'}_a \top^B_{s'} = \top^B_s, \\
(Back)\ & m_\sigma(\dots, b_{i-1}, @^{s_i,s'}_a b, b_{i+1}, \dots) \leq @^{s,s'}_a b.
\end{aligned}
$$

Some useful derived algebraic laws

From the basic axioms one derives:

$$
\begin{aligned}
@_a^s(b_1 \vee b_2) &= @_a^s(b_1) \vee @_a^s(b_2),\\
@_a^s(b_1 \wedge b_2) &= @_a^s(b_1) \wedge @_a^s(b_2),\\
@_{a_2}^{s,s''} b &\leq @_{a_1}^{s,s'} @_{a_2}^{s',s''} b,\\
f_s(a) \wedge @_a^{s,s} b &\leq b,\\
@_a^{s,s'} f_{s'}(a') &= @_{a'}^{s,s'} f_{s'}(a),\\
@_a^s b = \top_s^B &\iff f_{s'}(a) \leq b,\\
@_a^s b = \bot_s^B &\iff f_{s'}(a) \leq \neg b.
\end{aligned}
$$

Permeated structures

A $\Sigma\mathfrak{T}$-structure $\mathcal{AB}$ is *permeated* if the following hold:

($p1$) $\forall b \in B_s \backslash \{\bot_s^B\}$ $\exists a \in A_s$ such that $\mathfrak{f}_s(a) \leq b$,

($p2$) for any $a \in A_s, b \in B_s$, if $\mathfrak{f}(a) \leq \sigma(\dots, b_{i-1}, b, b_{i+1}, \dots)$,
then $\exists a' \in A_{s'}$ such that $\mathfrak{f}(a') \leq b$ and $\mathfrak{f}(a) \leq \sigma(\dots, b_{i-1}, \mathfrak{f}(a'), b_{i+1}, \dots)$.

Canonical examples of $\Sigma\mathfrak{T}$-structures

(e1) Full complex algebra of a frame. If

$$\mathcal{F} = (A, \{R_\sigma\}_{\sigma\in\Sigma}, \{F_f\}_{f\in\mathfrak{T}})$$

is a $\Sigma\mathfrak{T}$-frame, then one obtains a structure

$$A\mathcal{F}_A = (A, \mathcal{F}_A^+, f, \mathbb{@})$$

by taking the full complex algebra of the relational reduct and defining

$$f_s(a) = \{a\}, \qquad \mathbb{@}_a^{s,s'}X = \begin{cases} A_s, & a \in X, \\ \varnothing, & a \notin X. \end{cases}$$

This example is permeated.

Canonical examples of $\Sigma\mathfrak{T}$-structures

(e2) Lindembaum-Tarski algebra. We consider the logic $\mathcal{L}_{\Sigma\mathfrak{T}}(@) = \mathcal{H}_{\Sigma\mathfrak{T}}(@) \setminus \{(\mathrm{Name}@), (\mathrm{Paste})\}$.

For any $\phi, \psi \in \mathrm{FORM}_s$, we define: $\phi \equiv_s \psi$ iff $\vdash^s_{\mathcal{L}_{\Sigma\mathfrak{T}}} \phi \leftrightarrow \psi$.
Let $[\phi]$ be the equivalence class of ϕ, for any $\phi \in \mathrm{FORM}_s$.

For any $t \in \mathrm{TERM}_s$, we define: $\hat{t} := \left\{ t' \in \mathrm{TERM}_s \mid \vdash^s_{\mathcal{L}_{\Sigma\mathfrak{T}}} t \overset{s}{\approx} t' \right\}$.

We construct $AT := \{\hat{t} \mid t \in \mathrm{TERM}\}$, $\quad LT := \{[\phi] \mid \phi \in \mathrm{FORM}\}$
Then:

- AT is a $\mathfrak{T}$-algebra,
- LT is a Σ-BAO.

We define: $\mathfrak{f}_s(\hat{t}) := [t]$, $\quad @^{s',s}_{\hat{t}}[\phi] := [@^{s'}_t \phi]$

Hence $\mathcal{ALT} = (\mathcal{AT}, \mathcal{LT}, \mathfrak{f}, \{@^{s,s'}\}_{s,s' \in S})$ is a $\Sigma\mathfrak{T}$-structure.

Canonical examples of $\Sigma\mathfrak{T}$-structures

(e3) Herbrand algebra. Let $(S, \Sigma, \mathfrak{T})$ be a modal algebraic signature and let NOM be the set of nominals.

Then:

$$(T_{\mathfrak{T}}(\mathrm{NOM}), \{F_f^T\}_{f \in \mathfrak{T}})$$

is the $\mathfrak{T}$-term algebra generated by NOM.

A $\Sigma\mathfrak{T}$-**Herbrand algebra** is defined as the full complex algebra of the frame:

$$\left(T_{\mathfrak{T}}(\mathrm{NOM}), \{R_\sigma^T\}_{\sigma \in \Sigma}, \{F_f^T\}_{f \in \mathfrak{T}}\right),$$

where:

- $\{R_\sigma^T\}_{\sigma \in \Sigma}$ is a concrete family of relations on terms,
- $\{F_f^T\}_{f \in \mathfrak{T}}$ are the algebraic operations on terms.

Assignments, evaluation and satisfaction

An assignment in a $\Sigma\mathfrak{T}$-structure is a pair

$$(h, V)$$

where

$$h : \mathrm{NOM} \to A, \qquad V : \mathrm{PROP} \to B.$$

The map h extends uniquely to terms, and the map V extends uniquely to formulas, with

$$V(@_t^{s'}\varphi) = @^{s,s'}_{h(t)} V(\varphi).$$

For terms $t_1, t_2 \in \mathrm{TERM}_s$, one has

$$h_s(t_1) = h_s(t_2) \iff V_s(@^s_{t_1} t_2) = \top^{\mathcal{B}}_s.$$

So the hybrid formula $@^s_{t_1} t_2$ really behaves as an equality formula.

The satisfaction relation is then defined by:

$$\mathcal{AB} \models^s t_1 \overset{s}{\approx} t_2 \iff h_s(t_1) = h_s(t_2),$$
$$\mathcal{AB} \models^s \varphi \iff V_s(\varphi) = \top^{\mathcal{B}}_s.$$

Algebraic soundness and completeness

Let

$$L_{\Sigma\mathfrak{T}}(@) = \mathcal{H}_{\Sigma\mathfrak{T}}(@) \setminus \{(Name@), (Paste)\}.$$

Then the main algebraic results are:

Soundness.

- $L_{\Sigma\mathfrak{T}}(@)$ is sound with respect to all $\Sigma\mathfrak{T}$-structures;
- $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$ is sound with respect to all permeated $\Sigma\mathfrak{T}$-structures.

Completeness.

1. $L_{\Sigma\mathfrak{T}}(@)$ is sound and complete with respect to the class of all $\Sigma\mathfrak{T}$-structures.
2. For every set Λ of pure formulas, $\mathcal{H}_{\Sigma\mathfrak{T}}(@) + \Lambda$ is sound and complete with respect to the class of permeated $\Sigma\mathfrak{T}$-structures validating Λ.

Application to security protocols

The final section specializes the general framework to protocol analysis.

This is a natural fit because protocol analysis needs, at the same time:

- **state structure**: what each agent explicitly knows at each stage;
- **actions**: send / receive;
- **epistemic operators**: belief and knowledge.

Sorts for protocol analysis

The protocol logic uses four sorts:

$$S = \{msg, act, st, prot\}.$$

Message terms:

$$\mu_{msg} ::= m \mid (\mu_{msg}, \mu_{msg}) \mid \{\mu_{msg}\}_k$$

where m ranges over atomic messages and k over cryptographic keys.

Action terms:

$$\alpha_{act} ::= 0_{act} \mid send_{a,b}\, \mu_{msg} \mid recv_a\, \mu_{msg} \mid \alpha_{act}; \alpha_{act}.$$

State nominal terms:

$$\gamma_{st_{nom}} ::= a \triangleleft \mu_{msg} \mid (a \triangleleft \mu_{msg}, \gamma_{st_{nom}}).$$

Protocol formulas:

$$\varphi_{prot} ::= p \mid \neg\varphi \mid \varphi \to \varphi \mid B_a\varphi \mid \langle\gamma_{st}\rangle \mid [\alpha_{act}]\varphi \mid X_a\mu_{msg}.$$

Explicit knowledge, belief, and knowledge

$X_a m$: agent *a explicitly knows* the message m.

This is distinct from implicit logical consequence and is used to avoid logical omniscience.

Belief is represented by modal operators B_a, satisfying the KD45 axioms:

$$
\begin{aligned}
(K_B) \quad & B_a(\varphi \to \psi) \to (B_a\varphi \to B_a\psi), \\
(D) \quad & B_a\varphi \to \neg B_a \neg \varphi, \\
(4) \quad & B_a\varphi \to B_a B_a \varphi, \\
(5) \quad & \neg B_a\varphi \to B_a \neg B_a \varphi.
\end{aligned}
$$

Knowledge is then defined by truth plus belief:

$$K_a\varphi := B_a\varphi \wedge \varphi.$$

Explicit knowledge, belief, and knowledge

The system also includes explicit knowledge principles such as:

$$X_a(m_1, m_2) \leftrightarrow X_a m_1 \wedge X_a m_2,$$
$$X_a\{m\}_{pk(b)} \wedge X_a sk_b \rightarrow X_a m,$$
$$X_a\{m\}_{sk(b)} \wedge X_a pk_b \rightarrow X_a m.$$

These capture tuple decomposition and cryptographic deduction.

Dynamic and state axioms

The protocol logic contains dynamic laws for actions and state evolution.

Sequential composition:

$$[\alpha;\alpha']\varphi \leftrightarrow [\alpha][\alpha']\varphi.$$

State evolution via send/receive:

$$(H1)\ \langle a \triangleleft m, \gamma\rangle \rightarrow [send_{a,b}m]\langle a \triangleleft m, \gamma\rangle,$$
$$(H2)\ \langle \gamma\rangle \rightarrow [recv_a m]\langle a \triangleleft m, \gamma\rangle.$$

If an adversary e is explicitly tracked, these become:

$$(H_1^e)\ \langle a \triangleleft m, \gamma\rangle \rightarrow [send_{a,b}m]\langle e \triangleleft m, a \triangleleft m, \gamma\rangle,$$
$$(H_2^e)\ \langle e \triangleleft m, \gamma\rangle \rightarrow [recv_a m]\langle a \triangleleft m, e \triangleleft m, \gamma\rangle.$$

We have a Dolev-Yao style reasoning: the adversary can intercept, store, and resend messages, but cannot decrypt without the appropriate keys.

Example: the OSS protocol

The paper first illustrates the framework on the very simple protocol:

$$i \to r : \{i, n\}_{pk(r)}.$$

Two protocol-specific axioms encode its epistemic effect:

$$(oss1)\ \langle a \triangleleft m, \gamma \rangle \to [send_{a,b}\{a, m\}_{pk(b)}; \alpha] B_a X_b m,$$
$$(oss2)\ \langle \gamma \rangle \to [recv_b\{a, m\}_{pk(b)}; \alpha] B_b X_a m.$$

From these axioms one derives the epistemic outcome:

$$\vdash \langle i \triangleleft n, i \triangleleft \{i, n\}_{pk(r)}, \gamma_0 \rangle \to [send_{i,r}\{i, n\}_{pk(r)}; recv_r\{i, n\}_{pk(r)}](K_i X_r n \wedge K_r X_i n).$$

So, after the send/receive sequence, the initiator knows that the responder explicitly knows the nonce, and the responder knows that the initiator explicitly knows it as well.

Automatically generated model

The implementation aspect of the paper is particularly nice.

A protocol specification has the form

$$\text{Protocol} = \langle Ag, Keys, Actions \rangle$$

where

- $Ag = \{a_1, \ldots, a_n\}$ is the set of agents,
- $Keys$ is the finite set of keys,
- $Actions = [\alpha_1, \ldots, \alpha_m]$ is a sequence of send-actions.

A state at time t is

$$w^{(t)} = (X_{a_1}^{(t)}, \ldots, X_{a_n}^{(t)})$$

where $X_a^{(t)}$ is the set of messages explicitly known by a at time t.

Initial knowledge is defined from public/private/shared keys, and then the state sequence

$$w^{(0)}, w^{(1)}, \ldots, w^{(m)}$$

is generated algorithmically by processing the actions one by one.

Accessibility relations in the generated model

For each action $\alpha_t = \mathit{send}(s_t, r_t, m_t)$, the dynamic relation contains

$$(w^{(t)}, w^{(t+1)}) \in R_{\alpha_t}.$$

For beliefs, the relation of agent a is defined by the first time t_a at which the agent becomes active:

$$t_a := \min\{t : a \text{ appears in } \alpha_t\}.$$

Then

$$R_a^B := \{(w^{(i)}, w^{(j)}) : i \geq t_a,\ i < j\} \cup \{(w^{(m)}, w^{(m)})\}.$$

This guarantees the KD45 shape of the accessibility relation:

- seriality,
- transitivity,
- Euclidean property.

So the logical axioms are matched by a concrete model construction that can be generated from the execution trace itself.

Automatically Generated Model

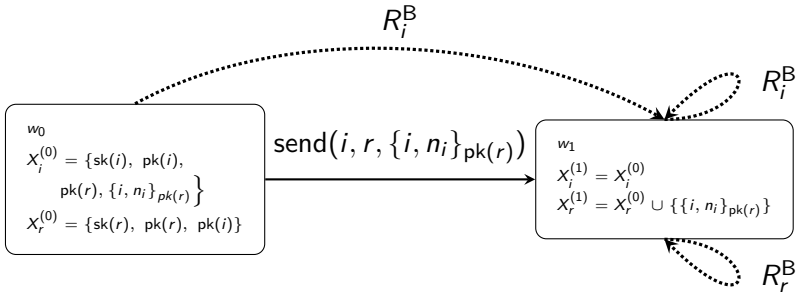

Figure: Automatically generated model for OSS

Why the framework is interesting conceptually

1. Hybrid logic becomes more structural. Nominals are replaced by nominal terms, so states are no longer merely named-they are algebraically generated.

2. The semantics two-sided.

- a frame semantics with operations on worlds,
- an algebraic semantics with a term algebra and a BAO tied together by @.

3. Real-world applications. Security protocols naturally require explicit states, actions, epistemic operators, and symbolic terms for messages.

4. The framework is implementable. The Lean 4 formalization and the automatic model extraction show that the theory is suitable not only for abstract metatheory, but also for mechanized reasoning.

Conclusion and possible talking points for discussion

Conclusion. The paper extends many-sorted hybrid logic by equipping nominals with algebraic structure. This yields:

- a new logic $\mathcal{H}_{\Sigma\mathfrak{T}}(@)$ with terms as formulas,
- completeness with respect to $\Sigma\mathfrak{T}$-frames,
- algebraic completeness with respect to suitable $\Sigma\mathfrak{T}$-structures,
- a concrete epistemic/action logic for security protocols.

Future work.

- richer protocol theories,
- more general operational semantics encoded in the same style,
- further Lean formalization for the generic many-sorted hybrid setting.

Thank you for your attention!