Bimodal Reference Manual

A Logic for Tense and Modality

Benjamin Brast-McKie

www.benbrastmckie.com

— January 21, 2026 —

Primary Reference:

"The Construction of Possible Worlds", Brast-McKie, (under review), 2025.

Abstract

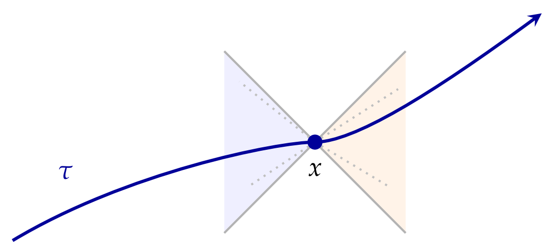
This reference manual provides the formal specification of the Bimodal logic **TM** for tense and modality as implemented in the ProofChecker project. **TM** is a bimodal logic combining an S5 historical necessity operator with linear temporal operators for the past and future tenses. Soundness and the deduction theorem are established. Completeness is proven via the semantic canonical model approach: the Lindenbaum lemma, truth lemma, and weak completeness theorem are all proven. The key result `semantic_weak_completeness` demonstrates that validity implies derivability.

Contents

1 Introduction

The bimodal logic **TM** combines S5 historical modal operators for necessity ($\Box$) and possibility ($\Diamond$) with linear temporal operators for past (H) and future (G). Implementing **TM** in Lean 4 provides a positive RL signal for training AI systems to conduct verified reasoning about past and future contingency in an interpreted formal language.

The semantics is based on *task frames*, which extend Kripke frames with temporal structure. A task frame consists of world states connected by a *task relation* indexed by temporal durations. World histories are temporal slices through a task frame, representing the possible evolution of a system.



The diagram above illustrates the conceptual structure underlying TM logic. The solid curve τ represents a single world history—a temporal sequence of states. From any point x along a history, the past and future light cones contain all states that are modally accessible. The necessity operator $\Box$ quantifies over all possible histories, though we may often restrict to those histories that pass through the world state $\tau(x)$. The temporal operators H and G quantify over past and future times which, given a particular history, determines the range of past and future world states that history occupies relative to a given time. These primitive operators may then be used to define a host of combined operators of interest.

Project Structure

The Lean 4 implementation is in the `Bimodal/` directory:

- `Syntax/` – Defines the formula language with 6 primitive constructors and derived operators. **Complete.**
- `ProofSystem/` – Axioms (14 schemata) and inference rules (7 rules) forming a Hilbert-style proof system. **Complete.**
- `Semantics/` – Task frames model possible worlds; world histories model time; truth conditions define meaning. **Complete.**

- `Metalogic/` – Soundness theorem (proven: all axioms valid, rules preserve validity), deduction theorem (proven: enables assumption introduction), completeness via the semantic canonical model (Lindenbaum lemma proven, truth lemma proven, weak completeness proven), and decidability via tableau-based decision procedure (soundness and completeness proven). **Complete.**
- `Theorems/` – Perpetuity principles and modal theorems derived from the axiom system. **Partial.**

2 Syntax

2.1 Formulas

Formulas are defined inductively with six primitive constructors.

Definition 2.1 (Formula)**.** The type `Formula` is defined by:

$$\varphi, \psi ::= p \mid \bot \mid \varphi \to \psi \mid \Box\varphi \mid H\varphi \mid G\varphi$$

where p ranges over propositional atoms (type `String`).

Symbol	Name	Lean	Reading
p, q, r	Atom	`atom s`	propositional atom
$\bot$	Bottom	`bot`	falsity
$\varphi \to \psi$	Implication	`imp`	"if φ then ψ"
$\Box\varphi$	Necessity	`box`	"necessarily φ"
$H\varphi$	Always past	`all_past`	"always in the past, φ"
$G\varphi$	Always future	`all_future`	"always in the future, φ"

2.2 Derived Operators

The following operators are defined in terms of the primitives.

Definition 2.2 (Propositional)**.**

$$\begin{aligned} \neg\varphi &:= \varphi \to \bot \\ \varphi \wedge \psi &:= \neg(\varphi \to \neg\psi) \\ \varphi \vee \psi &:= \neg\varphi \to \psi \end{aligned}$$

Symbol	Name	Lean	Reading
$\neg\varphi$	Negation	`neg`	"not φ"
$\varphi \wedge \psi$	Conjunction	`and`	"φ and ψ"
$\varphi \vee \psi$	Disjunction	`or`	"φ or ψ"

Definition 2.3 (Modal)**.**

$$\Diamond\varphi := \neg\Box\neg\varphi$$

Symbol	Name	Lean	Reading
$\Diamond\varphi$	Possibility	`pos`	"possibly φ"

Definition 2.4 (Temporal)**.**

$$\begin{aligned} P\varphi &:= \neg H\neg\varphi \\ F\varphi &:= \neg G\neg\varphi \\ \triangle\varphi &:= H\varphi \wedge \varphi \wedge G\varphi \\ \triangledown\varphi &:= P\varphi \vee \varphi \vee F\varphi \end{aligned}$$

Symbol	Name	Lean	Reading
$P\varphi$	Sometime past	`some_past`	"at some past time, φ"
$F\varphi$	Sometime future	`some_future`	"at some future time, φ"
$\triangle\varphi$	Always	`always`	"at all times, φ"
$\triangledown\varphi$	Sometimes	`sometimes`	"at some time, φ"

2.3 Temporal Duality

The `swap_temporal` function exchanges past and future operators.

Definition 2.5 (Temporal Swap)**.** The function $\langle S\rangle$: `Formula` $\to$ `Formula` is defined by structural induction on formulas:

$$\begin{aligned} \langle S\rangle p &= p \\ \langle S\rangle\bot &= \bot \\ \langle S\rangle(\varphi \to \psi) &= (\langle S\rangle\varphi \to \langle S\rangle\psi) \\ \langle S\rangle\Box\varphi &= \Box\langle S\rangle\varphi \\ \langle S\rangle H\varphi &= G\langle S\rangle\varphi \\ \langle S\rangle G\varphi &= H\langle S\rangle\varphi \end{aligned}$$

Theorem 2.6 (Involution)**.** $\langle S\rangle\langle S\rangle\varphi = \varphi$

3 Task Semantics

3.1 Task Frames

Task frames are the fundamental semantic structures for **TM**. They abstract from universal laws governing transitions between world states while still retaining the temporal duration for a transition to complete.

The following primitives are required to define a task frame:

Primitive	Type	Description
W	Type	World states
D	Type	Temporal durations
$w \Rightarrow_x u$	$W \to D \to W \to \text{Prop}$	Task relation

Definition 3.1 (Task Frame)**.** A **task frame** over temporal type D is a triple $\mathcal{F} = (W, D, \Rightarrow)$ satisfying:

1. **Nullity**: For all $w : W$, we have $w \Rightarrow_0 w$.
2. **Compositionality**: For all $w, u, v : W$ and $x, y : D$, if $w \Rightarrow_x u$ and $u \Rightarrow_y v$, then $w \Rightarrow_{x+y} v$.

Nullity ensures that zero-duration tasks leave the world state unchanged. Compositionality ensures that executing tasks sequentially yields results consistent with a single task of combined duration.

3.2 World Histories

A world history is a function from times to world states that respects the task relation over a convex temporal domain. World histories represent possible paths through the space of world states.

Definition 3.2 (Convex Domain)**.** A domain $\text{dom} : D \to \text{Prop}$ is **convex** if whenever $a, c \in \text{dom}$ with $a \leq c$, every time b with $a \leq b \leq c$ is also in dom. More precisely, for all $a, b, c : D$, if $\text{dom}(a)$ and $\text{dom}(c)$ and $a \leq b \leq c$, then $\text{dom}(b)$. Convexity ensures the domain has no temporal gaps.

Definition 3.3 (World History)**.** A **world history** in a task frame $\mathcal{F}$ is a dependent function $\tau : (x : D) \to \text{dom}(x) \to W$ where $\text{dom} : D \to \text{Prop}$ is a convex subset of D and $\tau(x) \Rightarrow_{y-x} \tau(y)$ for all times $x, y : D$ with $\text{dom}(x)$, $\text{dom}(y)$, and $x \leq y$. We write $H_{\mathcal{F}}$ for all world histories over frame $\mathcal{F}$.

3.3 Task Models

A task model extends a task frame with an interpretation function that assigns truth values to sentence letters at world states. **Propositions** are subsets of W representing instantaneous ways for the system to be. Sentence letters express propositions, which can be realized by zero or more world states. World states themselves are specific configurations of the total system at an instant.

Definition 3.4 (Task Model)**.** A **task model** defined over a frame $\mathcal{F}$ is a pair $\mathcal{M} = (\mathcal{F}, I)$ where the **interpretation function** $I : W \to \text{String} \to \text{Prop}$ assigns to each world state $w : W$ and sentence letter $p : \text{String}$ a truth value $I(w, p) : \text{Prop}$. We write $I(w, p)$ to indicate that sentence letter p is true at w.

3.4 Truth Conditions

Truth is evaluated relative to a model $\mathcal{M}$ providing the interpretation, a world history τ representing a possible path through the space of world states, and a time $x : D$. Whereas the model fixes the interpretation of the language, the contextual parameters τ and x determine the truth value of every sentence of the langauge.

Definition 3.5 (Truth)**.** For model $\mathcal{M}$, history $\tau \in H_{\mathcal{F}}$, and time $x : D$:

$$
\begin{aligned}
\mathcal{M}, \tau, x &\vDash p \textit{ iff } x \in \text{dom}(\tau) \text{ and } I(\tau(x), p) \\
\mathcal{M}, \tau, x &\nvDash \bot \\
\mathcal{M}, \tau, x &\vDash \varphi \to \psi \textit{ iff } \mathcal{M}, \tau, x \nvDash \varphi \text{ or } \mathcal{M}, \tau, x \vDash \psi \\
\mathcal{M}, \tau, x &\vDash \Box\varphi \textit{ iff } \mathcal{M}, \sigma, x \vDash \varphi \text{ for all } \sigma \in H_{\mathcal{F}} \\
\mathcal{M}, \tau, x &\vDash H\varphi \textit{ iff } \mathcal{M}, \tau, y \vDash \varphi \text{ for all } y : D \text{ where } y < x \\
\mathcal{M}, \tau, x &\vDash G\varphi \textit{ iff } \mathcal{M}, \tau, y \vDash \varphi \text{ for all } y : D \text{ where } x < y
\end{aligned}
$$

The modal operator $\Box$ quantifies over all world histories $\sigma : H_{\mathcal{F}}$ at the current time $x : D$. The temporal operators H and G quantify over all earlier and later times $y : D$.

3.5 Time-Shift

The time-shift operation is used to establish the **perpetuity principles**:

- P1: $\Box\varphi \to \triangle\varphi$ (what is necessary is always the case)
- P2: $\triangledown\varphi \to \Diamond\varphi$ (what is sometimes the case is possible)

It is natural to assume that whatever is necessary is always the case, or equivalently, whatever is sometimes the case is possible. Time-shift enables proofs of the bimodal axioms MF ($\Box\varphi \to \Box G\varphi$) and TF ($\Box\varphi \to G\Box\varphi$) which together imply the perpetuity principles.

Definition 3.6 (Time-Shift)**.** For $\tau, \sigma \in H_{\mathcal{F}}$ and $x, y : D$, world histories τ and σ are **time-shifted from** y **to** x, written $\tau \approx^x_y \sigma$, if and only if there exists an order automorphism $\bar{a} : D \to D$ where $y = \bar{a}(x)$, $\mathrm{dom}_\sigma = \bar{a}^{-1}(\mathrm{dom}_\tau)$, and $\sigma(z) = \tau(\bar{a}(z))$ for all $z \in \mathrm{dom}_\sigma$.

Time-shifting preserves the essential structure of histories:

Theorem 3.7 (Convexity Preservation)**.** *If τ has a convex domain and $\tau \approx^x_y \sigma$, then σ has a convex domain.*

Theorem 3.8 (Task Preservation)**.** *If τ respects the task relation and $\tau \approx^x_y \sigma$, then σ respects the task relation.*

3.6 Logical Consequence and Validity

Logical consequence and validity are defined uniformly across all temporal types, frames, models, histories, and times.

Definition 3.9 (Logical Consequence)**.** A formula φ is a **logical consequence** of Γ (written $\Gamma \models \varphi$) just in case for every temporal type D : Type, frame $\mathcal{F}$: TaskFrame(D), model $\mathcal{M}$: TaskModel$(\mathcal{F})$, history $\tau \in H_{\mathcal{F}}$, and time $x : D$, if $\mathcal{M}, \tau, x \models \psi$ for all $\psi \in \Gamma$, then $\mathcal{M}, \tau, x \models \varphi$.

Definition 3.10 (Validity)**.** A formula φ is **valid** (written $\models \varphi$) just in case φ is a logical consequence of the empty set: $\varnothing \models \varphi$. Equivalently, φ is true at every model-history-time triple.

Definition 3.11 (Satisfiability)**.** A context Γ is **satisfiable** in temporal type D : Type if there exist a frame $\mathcal{F}$: TaskFrame(D), model $\mathcal{M}$: TaskModel$(\mathcal{F})$, history $\tau \in H_{\mathcal{F}}$, and time $x : D$ such that $\mathcal{M}, \tau, x \models \psi$ for all $\psi \in \Gamma$.

Theorem 3.12 (Monotonicity)**.** *If $\Gamma \subseteq \Delta$ and $\Gamma \models \varphi$, then $\Delta \models \varphi$.*

4 Proof Theory

4.1 Axiom Schemata

The TM proof system has 14 axiom schemata.

4.1.1 Propositional Axioms

Axiom 4.1 (K - Distribution). $(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$

Axiom 4.2 (S - Weakening). $\varphi \to (\psi \to \varphi)$

Axiom 4.3 (EFQ - Ex Falso Quodlibet). $\bot \to \varphi$

Axiom 4.4 (Peirce). $((\varphi \to \psi) \to \varphi) \to \varphi$

4.1.2 Modal Axioms (S5)

Axiom 4.5 (MT - Modal T). $\Box\varphi \to \varphi$

Axiom 4.6 (M4 - Modal 4). $\Box\varphi \to \Box\Box\varphi$

Axiom 4.7 (MB - Modal B). $\varphi \to \Box\Diamond\varphi$

Axiom 4.8 (M5 - Modal 5 Collapse). $\Diamond\Box\varphi \to \Box\varphi$

Axiom 4.9 (MK - Modal Distribution). $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$

4.1.3 Temporal Axioms

Axiom 4.10 (TK - Temporal Distribution). $G(\varphi \to \psi) \to (G\varphi \to G\psi)$

Axiom 4.11 (T4 - Temporal 4). $G\varphi \to GG\varphi$

Axiom 4.12 (TA - Temporal A). $\varphi \to GP\varphi$

Axiom 4.13 (TL - Temporal L). $\triangle\varphi \to GH\varphi$

4.1.4 Modal-Temporal Interaction

Axiom 4.14 (MF - Modal-Future). $\Box\varphi \to \Box G\varphi$

Axiom 4.15 (TF - Temporal-Future). $\Box\varphi \to G\Box\varphi$

Axiom	Lean Constructor	Pattern
K	`Axiom.prop_k`	Distribution
S	`Axiom.prop_s`	Weakening
EFQ	`Axiom.ex_falso`	Explosion
Peirce	`Axiom.peirce`	Classical
MT	`Axiom.modal_t`	Reflexivity
M4	`Axiom.modal_4`	Transitivity
MB	`Axiom.modal_b`	Symmetry
M5	`Axiom.modal_5_collapse`	S5 collapse
MK	`Axiom.modal_k_dist`	Modal distribution
TK	`Axiom.temp_k_dist`	Temporal distribution
T4	`Axiom.temp_4`	Temporal transitivity
TA	`Axiom.temp_a`	Connectedness
TL	`Axiom.temp_l`	Introspection
MF	`Axiom.modal_future`	Modal-future
TF	`Axiom.temp_future`	Temporal-modal

4.2 Inference Rules

The proof system has 7 inference rules.

Definition 4.16 (Axiom Rule)**.** If φ matches an axiom schema, then $\Gamma \vdash \varphi$.

Definition 4.17 (Assumption Rule)**.** If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.

Definition 4.18 (Modus Ponens)**.**

$$\frac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

Definition 4.19 (Necessitation)**.**

$$\frac{\vdash \varphi}{\vdash \Box\varphi}$$

Applies only to theorems (empty context).

Definition 4.20 (Temporal Necessitation)**.**

$$\frac{\vdash \varphi}{\vdash G\varphi}$$

Applies only to theorems (empty context).

Definition 4.21 (Temporal Duality)**.**

$$\frac{\vdash \varphi}{\vdash \langle S \rangle \varphi}$$

Applies only to theorems (empty context).

Definition 4.22 (Weakening)**.**

$$\frac{\Gamma \vdash \varphi \qquad \Gamma \subseteq \Delta}{\Delta \vdash \varphi}$$

Rule	Lean Constructor	Context Requirement
Axiom	`DerivationTree.axiom`	Any
Assumption	`DerivationTree.assumption`	Any
Modus Ponens	`DerivationTree.modus_ponens`	Any
Necessitation	`DerivationTree.necessitation`	Empty only
Temp. Necessitation	`DerivationTree.temporal_necessitation`	Empty only
Temporal Duality	`DerivationTree.temporal_duality`	Empty only
Weakening	`DerivationTree.weakening`	Any

4.3 Derivation Trees

Derivations are represented as inductive trees.

Definition 4.23 (Derivation Tree)**.** `DerivationTree` Γ φ (written $\Gamma \vdash \varphi$) is an inductive type representing a derivation of φ from context Γ.

Definition 4.24 (Height)**.** The height of a derivation tree:

- Base cases (axiom, assumption): height 0
- Unary rules: height of subderivation + 1
- Modus ponens: max of both subderivations + 1

The height measure enables well-founded recursion in metalogical proofs.

4.4 Notation

Notation	Lean
$\Gamma \vdash \varphi$	`DerivationTree` Γ φ
$\vdash \varphi$	`DerivationTree []` φ

5 Metalogic

The metalogic for the bimodal logic **TM** establishes that the proof system and semantics describe the same space of inferences. This chapter presents a representation theorem from which completeness and compactness follow as corollaries. The chapter also proves that **TM** is decidable.

5.1 Soundness

This theorem establishes that only logical consequences are derivable.

Theorem 5.1 (Soundness). *If* $\Gamma \vdash \varphi$ *then* $\Gamma \vDash \varphi$.

The proof proceeds by induction on the derivation structure:

- **Axioms**: Each of the 14 axiom schemata is proven valid
- **Assumptions**: Assumed formulas are true by hypothesis
- **Modus ponens**: Validity preserved under implication elimination
- **Necessitation**: Valid formulas become necessarily valid
- **Temporal necessitation**: Valid formulas become always-future valid
- **Temporal duality**: Past-future swap preserves validity
- **Weakening**: Adding premises preserves semantic consequence

Axiom Validity	Lean Theorem	Technique
`prop_k_valid`	Propositional K	Propositional reasoning
`prop_s_valid`	Propositional S	Propositional reasoning
`ex_falso_valid`	EFQ	Vacuous implication
`peirce_valid`	Peirce	Classical case analysis
`modal_t_valid`	MT	Reflexivity of accessibility
`modal_4_valid`	M4	Transitivity of accessibility
`modal_b_valid`	MB	Symmetry of accessibility
`modal_5_collapse_valid`	M5	S5 equivalence structure
`modal_k_dist_valid`	MK	Distribution
`temp_k_dist_valid`	TK	Temporal distribution
`temp_4_valid`	T4	Transitivity of time
`temp_a_valid`	TA	Temporal connectedness
`temp_l_valid`	TL	Always implies recurrence
`modal_future_valid`	MF	Time-shift invariance
`temp_future_valid`	TF	Time-shift invariance

The MF and TF axioms use time-shift invariance to relate truth at different times (via `WorldHistory.time_shift`).

5.2 Core Infrastructure

The completeness proof requires three foundational components including the deduction theorem, maximal consistent sets, and Lindenbaum's lemma. These provide the infrastructure for constructing canonical models.

5.2.1 Deduction Theorem

Theorem 5.2 (Deduction Theorem)**.** *If* $A :: \Gamma \vdash B$ *then* $\Gamma \vdash A \to B$*.*

The proof uses well-founded induction on derivation height, handling each of the following rules:

- **Axiom**: Use S axiom to weaken
- **Assumption**: Identity if same, S axiom if different
- **Modus ponens**: Use K axiom distribution
- **Weakening**: Case analysis on assumption membership
- **Modal/temporal rules**: Do not apply (require empty context)

5.2.2 Consistency

Definition 5.3 (Consistent)**.** A context Γ is **consistent** if $\Gamma \nvdash \bot$.

Definition 5.4 (Maximal Consistent)**.** A context Γ is **maximal consistent** if it is consistent and for all $\varphi \notin \Gamma$, the context $\varphi :: \Gamma$ is inconsistent.

Definition 5.5 (Negation-Complete)**.** A set of formulas S is **negation-complete** if for every formula φ, exactly one of φ or $\neg\varphi$ is in S.

Maximal consistent sets (MCS) are negation-complete. This property is essential for defining canonical world states, as it ensures that every formula has a definite truth value in each MCS.

5.2.3 Lindenbaum's Lemma

Lemma 5.6 (`set_lindenbaum`)**.** *Every consistent context can be extended to a maximal consistent context.*

The proof applies Zorn's lemma to the partially ordered set of consistent supersets of the given context. Note that contexts (finite lists of formulas) embed naturally into sets, so "consistent context" and "consistent set" are used interchangeably here. The key step is showing that the union of any chain of consistent sets is itself consistent. This follows because any derivation uses only finitely many premises, so a derivation of $\perp$ from the union would have to come from some finite subset, which is contained in some member of the chain, contradicting that member's consistency.

5.3 Representation Theory

The *Representation Theorem* is the core of the metalogic, providing the bridge between syntactic consistency and semantic satisfiability. The subsequent *Completeness Theorems* follow directly from this result.

5.3.1 Canonical World States

The semantic approach constructs world states from histories and times. We first define these constituent concepts before assembling them into canonical structures.

Definition 5.7 (History)**.** A **history** is a function $\tau : \mathbb{Z} \to \text{MCS}$ mapping each time to a maximal consistent set, satisfying temporal coherence: for each time t, the formulas in $\tau(t)$ are consistent with the temporal operators applied to formulas in $\tau(t')$ for related times t'.

Definition 5.8 (Time)**.** **Times** are integers ($\mathbb{Z}$), providing a discrete linear order with both past and future directions.

Definition 5.9 (Task Relation)**.** The **task relation** (`SemanticTaskRelV2`) relates world states based on the existence of histories connecting them: world state w is task-related to w' if there exists a history τ such that w and w' both occur along τ.

A **canonical world state** is derived from a maximal consistent set. The semantic approach defines world states as equivalence classes of (history, time) pairs.

Definition 5.10 (Semantic World State)**.** A world state is an equivalence class of (history, time) pairs under the relation where two pairs are equivalent iff they denote the same underlying world state.[1]

[1] This is formalized as `SemanticWorldState`.

Definition 5.11 (`SemanticCanonicalFrame`)**.** The **canonical semantic frame** has:

- World states: Equivalence classes of history-time pairs over maximal consistent sets
- Times: Integers ($\mathbb{Z}$)
- Task relation: Defined via history existence (`SemanticTaskRelV2`)

The frame satisfies nullity and compositionality.

Definition 5.12 (Canonical Valuation)**.** An atom p is true at world state $[\tau, x]$ iff $p \in \mathsf{MCS}(\tau(x))$.

Lemma 5.13 (Truth Lemma)**.** *In the semantic canonical model, $\varphi \in \mathsf{MCS}(\tau(x))$ iff φ is true at world state $[\tau, x]$.*[2]

The *quotient construction* identifies (history, time) pairs that agree on which formulas hold, forming equivalence classes. Two pairs (τ_1, t_1) and (τ_2, t_2) are equivalent when $\tau_1(t_1) = \tau_2(t_2)$ as maximal consistent sets. The *Truth Lemma* follows directly from this construction: membership in a maximal consistent set corresponds to truth by definition of the equivalence class.

5.3.2 Representation Theorem

Theorem 5.14 (Representation Theorem (`representation_theorem`))**.** *Every consistent context is satisfiable in the canonical model.*

This theorem is the pivotal result linking syntax to semantics. The proof strategy is:

1. Given a consistent context Γ, convert it to a set $S = \texttt{contextToSet}(\Gamma)$.
2. Apply Lindenbaum's lemma to extend S to a maximal consistent set M.
3. View M as a canonical world state.
4. By the truth lemma, all formulas in Γ are satisfied at this world.

The elegance of this approach is that the MCS construction makes the *Truth Lemma* essentially trivial since truth *is* membership in the MCS.

Theorem 5.15 (Strong Representation Theorem)**.** *If $\Gamma \nvdash \varphi$, then $\Gamma \cup \{\neg\varphi\}$ is satisfiable in the canonical model.*[3]

[2]This is proven as `semantic_truth_lemma_v2`.
[3]This is proven as `strong_representation_theorem`.

This strengthening is crucial for completeness: it says that any formula unprovable from a context can be made false while satisfying the context. The proof adds $\neg\varphi$ to the context and applies the standard *Representation Theorem* to the resulting consistent set.

5.3.3 Theorem Dependency Structure

Figure 1 illustrates the dependency structure of the completeness proof. The core infrastructure (top row) feeds into the central *Representation Theorem*, from which the completeness corollaries follow.

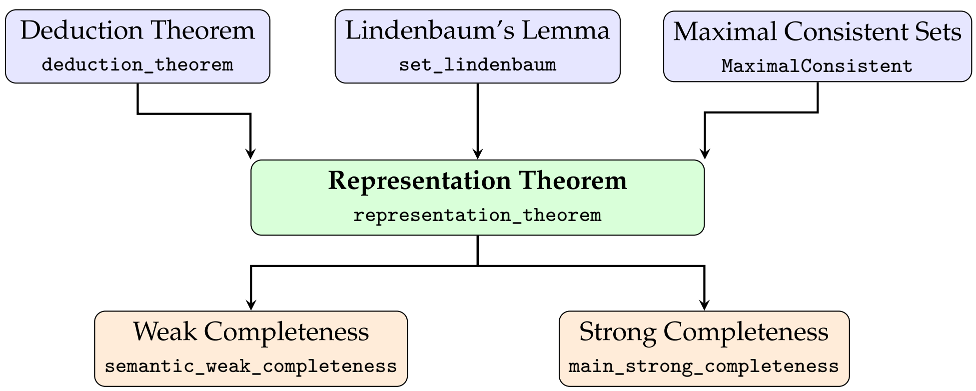


Figure 1: Theorem dependency structure for completeness.

The three foundational components—the *Deduction Theorem*, *Lindenbaum's Lemma*, and *Maximal Consistent Sets*—provide the infrastructure for the *Representation Theorem*. Both weak and strong completeness then follow as direct corollaries via contrapositive arguments.

5.4 Completeness as Corollary

The *Completeness Theorems* follow directly from the *Representation Theorem* via contrapositive arguments.

5.4.1 Weak Completeness

Theorem 5.16 (Weak Completeness). *If* $\vDash \varphi$*, then* $\vdash \varphi$.[4]

The proof proceeds by contraposition:

[4] This is proven as `semantic_weak_completeness`.

1. Assume $\nvdash \varphi$ (i.e., the empty context does not derive φ).
2. Then $\{\neg\varphi\}$ is consistent (otherwise we could derive φ).
3. By the *Representation Theorem*, $\{\neg\varphi\}$ is satisfiable in the canonical model.
4. So there exists a world where $\neg\varphi$ is true, meaning φ is false.
5. Hence φ is not valid.

By contraposition, validity implies provability.

5.4.2 Strong Completeness

Theorem 5.17 (Strong Completeness)**.** *If* $\Gamma \vDash \varphi$*, then* $\Gamma \vdash \varphi$*.*[5]

The proof extends weak completeness using an implication chain technique:

1. Assume semantic consequence: $\Gamma \vDash \varphi$.
2. For context $\Gamma = [\psi_1, \ldots, \psi_n]$, build the implication chain $\psi_1 \to (\psi_2 \to \cdots (\psi_n \to \varphi))$.
3. Show this chain is valid (from the semantic consequence assumption).
4. By weak completeness, the chain is provable.
5. Unfold the chain with repeated modus ponens applications to obtain $\Gamma \vdash \varphi$.

Theorem 5.18 (Provable iff Valid)**.** $\vdash \varphi$ *iff* $\vDash \varphi$*.*[6]

This biconditional shows that the proof system and semantics align perfectly. Soundness (left-to-right) ensures no non-logical consequences are derivable, while completeness (right-to-left) ensures all logical consequences are captured by the proof system. Together, they establish that **TM** provides an exact syntactic characterization of semantic validity.

5.4.3 Two Canonical Model Approaches

The codebase contains two canonical model constructions. Understanding their differences explains why the semantic approach is primary.

Syntactic Approach. World states are directly identified with maximal consistent sets. Accessibility is defined via modal witnesses: $\Box\varphi \in w$

[5]This is proven as `main_strong_completeness` with bridge sorries for the generalization.
[6]This is proven as `main_provable_iff_valid`, establishing completeness of **TM**.

implies $\varphi \in w'$ for all accessible w'. This approach requires explicit negation-completeness proofs for locally consistent sets. The syntactic approach is archived in `Boneyard/` for historical reference.

Semantic Approach. World states are equivalence classes of (history, time) pairs, where two pairs are equivalent iff they denote the same underlying world state. This approach offers key advantages:

- **Truth lemma**: Follows trivially from the quotient construction.
- **Compositionality**: The task relation is defined via history concatenation, making compositionality proofs straightforward.
- **Negation-completeness**: The semantic approach does not require proving negation-completeness of arbitrary locally consistent sets, a property that caused difficulties in the syntactic approach.

5.5 Decidability

The decidability of TM bimodal logic rests on the *Finite Model Property*: if a formula is satisfiable, it is satisfiable in a finite model bounded by the formula's modal and temporal depth. The bound on model size is $2^{|\mathsf{closure}(\varphi)|}$, where the subformula closure contains all relevant formulas for determining the truth of φ. This property connects the representation infrastructure to decidability by ensuring that satisfiability checking can terminate.

The decidability proof proceeds via a tableau-based decision procedure.

Theorem 5.19 (Decidability)**.** *Validity in TM bimodal logic is decidable: for any formula φ, either $\vdash \varphi$ or $\neg \vdash \varphi$.*

Theorem 5.20 (Decision Soundness)**.** *If the decision procedure returns "valid" with proof π, then $\vdash \varphi$.*[7]

Here π is a derivation term witnessing $\vdash \varphi$, constructed from the closed tableau. In Lean 4, this is a term of type `ContextDerivable [] `φ, representing a formal proof tree.

The decision procedure operates as follows:

1. **Direct axiom proof**: Check if φ matches one of the axiom schemata directly, yielding an immediate derivation.
2. **Proof search**: Apply Lean 4 tactics (`decide`, `simp`) with bounded recursion depth to find a derivation automatically.

[7]This is proven as `decide_sound`.

3. **Tableau construction**: Build a systematic tree that decomposes φ into simpler signed formulas.
4. If all branches close: formula is valid, extract proof.
5. If open saturated branch: formula is invalid, extract countermodel.

A *tableau* is a tree-structured refutation method: to prove φ valid, we assume φ is false and systematically derive contradictions. Each branch represents a possible scenario; if all branches lead to contradictions, the original assumption was impossible, so φ must be valid.

5.5.1 Tableau Structure

The tableau uses **signed formulas** with annotations:

- $T(\varphi)$: formula φ is assumed true
- $F(\varphi)$: formula φ is assumed false

Expansion rules are categorized as:

- **Propositional**: $T(\varphi \wedge \psi)$ splits, $F(\varphi \rightarrow \psi)$ splits, etc.
- **Modal**: $T(\Box\varphi)$ propagates to accessible worlds, $F(\Diamond\varphi)$ creates witness
- **Temporal**: $T(\triangle\varphi)$ propagates to future times, $F(\triangledown\varphi)$ creates witness

A branch **closes** when it contains both $T(\varphi)$ and $F(\varphi)$ for some formula. A branch is **saturated** when no expansion rules apply.

5.5.2 Complexity

Measure	Complexity
Time	$O(2^n)$ where n is formula size
Space	$O(n)$
Class	PSPACE-complete

The exponential time bound means formulas of modest size (30–50 symbols) remain tractable on modern hardware. PSPACE-completeness implies that modal satisfiability is among the hardest problems solvable with polynomial space, but the linear space bound makes memory usage manageable even for larger formulas.

5.5.3 Decision Result Types

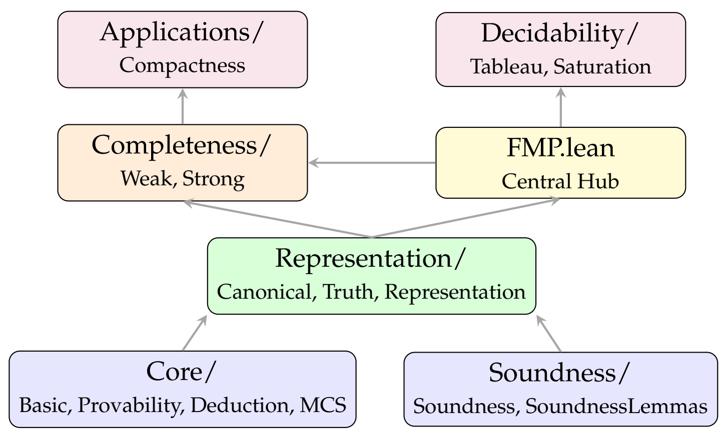
The decision procedure returns one of three outcomes:

- `valid proof`: Formula is valid with derivation tree
- `invalid counter`: Formula is invalid with countermodel
- `timeout`: Resources exhausted before decision

Despite computational limitations, decidability is practically valuable: small formulas (most proof obligations in practice) resolve quickly, invalid formulas are often rejected early without full exploration, and countermodels provide concrete feedback for debugging specifications.

5.6 File Organization and Dependencies

The `Metalogic_v2` directory contains 27 Lean files organized into six subdirectories. The following diagram illustrates the import structure, showing how files depend on each other.



Directory descriptions:

- **`Core/`**: Foundational definitions including provability (`ContextDerivable`), deduction theorem, and maximal consistent sets.
- **`Soundness/`**: Validity proofs for all 15 axiom schemata and the main soundness theorem.
- **`Representation/`**: Canonical model construction, truth lemma, and the central representation theorems.
- **`Completeness/`**: Weak and strong completeness theorems derived from representation.
- **`Applications/`**: Compactness theorem (trivial for list-based contexts).
- **`Decidability/`**: Tableau-based decision procedure with proof/countermodel extraction.

5.7 Implementation Status

5.7.1 Sorry Status

The `Metalogic_v2` codebase has three remaining `sorry` statements, none of which block the core completeness results:

1. `semantic_task_rel_compositionality` (SemanticCanonicalModel.lean:236) — Finite time domain limitation; a fundamental issue with Int-valued durations exceeding finite time bounds.
2. `main_provable_iff_valid_v2` completeness direction (SemanticCanonicalModel.lean:647) — Requires truth bridge from general validity to finite model truth.
3. `finite_model_property_constructive` truth bridge (FiniteModelProperty.lean:481) — Same truth bridge issue.

The core completeness result `semantic_weak_completeness` is fully proven without sorries.

5.7.2 Decidability Implementation

Submodule	Status	Notes
SignedFormula	Complete	Sign, SignedFormula, Branch types
Tableau	Complete	Expansion rules
Closure	Complete	Branch closure detection
Saturation	Complete	Fuel-based termination
ProofExtraction	Partial	Axiom instances only
CountermodelExtraction	Complete	From open branches
DecisionProcedure	Complete	Main decide function
Soundness	Proven	`decide_sound`
Completeness	Partial	Requires Finite Model Property

5.7.3 Metalogic Implementation

Component	Status	Lean
Soundness	Proven	`soundness`
Deduction Theorem	Proven	`deduction_theorem`
Lindenbaum Lemma	Proven	`set_lindenbaum`
Canonical Frame	Proven	`SemanticCanonicalFrame`
Truth Lemma	Proven	`semantic_truth_lemma_v2`
Representation Theorem	Proven	`representation_theorem`
Weak Completeness	Proven	`semantic_weak_completeness`
Strong Completeness	Proven*	`main_strong_completeness`
Provable iff Valid	Proven	`main_provable_iff_valid`
Finite Model Property	Statement	`finite_model_property`
Decidability Soundness	Proven	`decide_sound`

* Strong completeness uses the weak completeness result, which is fully proven. The three sorries listed above affect only the finite model property path.

6 Theorems

6.1 Perpetuity Principles

The perpetuity principles establish deep connections between modal necessity ($\Box$) and temporal operators ($\triangle$, $\triangledown$).

Theorem 6.1 (P1: Necessity Implies Always). $\vdash \Box\varphi \rightarrow \triangle\varphi$

Theorem 6.2 (P2: Sometimes Implies Possible). $\vdash \triangledown\varphi \rightarrow \Diamond\varphi$

Theorem 6.3 (P3: Necessity of Perpetuity). $\vdash \Box\varphi \rightarrow \Box\triangle\varphi$

Theorem 6.4 (P4: Possibility of Occurrence). $\vdash \Diamond\triangledown\varphi \rightarrow \Diamond\varphi$

Theorem 6.5 (P5: Persistent Possibility). $\vdash \Diamond\triangledown\varphi \rightarrow \triangle\Diamond\varphi$

Theorem 6.6 (P6: Occurrent Necessity is Perpetual). $\vdash \triangledown\Box\varphi \rightarrow \Box\triangle\varphi$

All six perpetuity principles are fully proven in the Lean implementation.

Principle	Lean Theorem	Key Lemmas
P1	`perpetuity_1`	MF, TF, MT
P2	`perpetuity_2`	Contraposition of P1
P3	`perpetuity_3`	P1, box_mono
P4	`perpetuity_4`	Contraposition
P5	`perpetuity_5`	modal_5, temporal K
P6	`perpetuity_6`	P5, bridge lemmas

6.2 Modal S5 Theorems

Theorem 6.7 (T-Box-to-Diamond). $\vdash \Box\varphi \to \Diamond\varphi$

Theorem 6.8 (Box Distributes Over Disjunction). $\vdash (\Box\varphi \vee \Box\psi) \to \Box(\varphi \vee \psi)$

Theorem 6.9 (Box Preserves Contraposition). $\vdash \Box(\varphi \to \psi) \to \Box(\neg\psi \to \neg\varphi)$

Theorem 6.10 (K Distribution for Diamond). $\vdash \Box(\varphi \to \psi) \to (\Diamond\varphi \to \Diamond\psi)$

Theorem 6.11 (S5 Collapse). $\vdash \Diamond\Box\varphi \leftrightarrow \Box\varphi$

Theorem 6.12 (Box-Conjunction). $\vdash \Box(\varphi \wedge \psi) \leftrightarrow (\Box\varphi \wedge \Box\psi)$

Theorem 6.13 (Diamond-Disjunction). $\vdash \Diamond(\varphi \vee \psi) \leftrightarrow (\Diamond\varphi \vee \Diamond\psi)$

Theorem 6.14 (S5 Diamond-Box to Truth). $\vdash \Diamond\Box\varphi \to \varphi$

Theorem 6.15 (T-Box Consistency). $\vdash \Box(\varphi \wedge \neg\varphi) \to \bot$

6.3 Modal S4 Properties

The following S4 properties are derived from the TM axiom system.

Theorem 6.16 (Modal 5). $\vdash \Diamond\varphi \to \Box\Diamond\varphi$

Theorem 6.17 (Diamond 4). $\vdash \Diamond\Diamond\varphi \to \Diamond\varphi$

Theorem 6.18 (Box Monotonicity). *If* $\vdash \varphi \to \psi$ *then* $\vdash \Box\varphi \to \Box\psi$.

Theorem 6.19 (Diamond Monotonicity). *If* $\vdash \varphi \to \psi$ *then* $\vdash \Diamond\varphi \to \Diamond\psi$.

6.4 Propositional Theorems

Theorem 6.20 (Identity). $\vdash \varphi \to \varphi$

Theorem 6.21 (Double Negation Introduction). $\vdash \varphi \to \neg\neg\varphi$

Theorem 6.22 (Double Negation Elimination). $\vdash \neg\neg\varphi \to \varphi$

Theorem 6.23 (Contraposition). *If* $\vdash \varphi \to \psi$ *then* $\vdash \neg\psi \to \neg\varphi$.

Theorem 6.24 (De Morgan Disjunction). $\vdash \neg(\varphi \vee \psi) \leftrightarrow (\neg\varphi \wedge \neg\psi)$

Theorem 6.25 (De Morgan Conjunction). $\vdash \neg(\varphi \wedge \psi) \leftrightarrow (\neg\varphi \vee \neg\psi)$

6.5 Combinator Infrastructure

The combinator infrastructure provides Hilbert-style proof tools.

Theorem 6.26 (B Combinator). $\vdash (B \to C) \to ((A \to B) \to (A \to C))$

Theorem 6.27 (Implication Transitivity). *If* $\vdash A \to B$ *and* $\vdash B \to C$ *then* $\vdash A \to C$.

Theorem 6.28 (Pairing). $\vdash A \to (B \to (A \wedge B))$

Theorem 6.29 (Classical Merge). $\vdash (P \to Q) \to ((\neg P \to Q) \to Q)$

6.6 Generalized Necessitation

Theorem 6.30 (Generalized Modal Necessitation). *If* $\Gamma \vdash \varphi$ *then* $\Box\Gamma \vdash \Box\varphi$ *where* $\Box\Gamma = [\Box\psi \mid \psi \in \Gamma]$.

Theorem 6.31 (Generalized Temporal Necessitation). *If* $\Gamma \vdash \varphi$ *then* $G\Gamma \vdash G\varphi$ *where* $G\Gamma = [G\psi \mid \psi \in \Gamma]$.

6.7 Module Organization

Module	Contents
`Perpetuity.lean`	P1-P6 principles
`ModalS5.lean`	S5 characteristic theorems (24 theorems)
`ModalS4.lean`	S4 properties (modal_5, diamond_4)
`Propositional.lean`	Classical propositional theorems
`Combinators.lean`	B, I, S combinators, imp_trans
`GeneralizedNecessitation.lean`	Context-level necessitation

Total theorem count: 228 theorems/lemmas across the Theorems/ directory.

7 Notes

7.1 Implementation Status

Component	Status	Notes
Syntax	Complete	6 primitives, derived operators
Semantics	Complete	Task frames, world histories, truth
Proof System	Complete	14 axioms, 7 inference rules
Soundness	Proven	All 14 axioms valid, 7 rules sound
Deduction Theorem	Proven	Well-founded recursion on height
Completeness	Proven (Semantic)	Lindenbaum, truth lemma, weak completeness
Decidability	Soundness Proven	Tableau-based, FMP for full completeness
Perpetuity Principles	Proven	P1-P6 all proven

7.2 Discrepancy Notes

This section documents differences between the paper "The Construction of Possible Worlds" and the Lean implementation.

7.2.1 Terminology

- The paper uses "perpetuity principles" for P1-P6; the Lean code uses the same terminology.
- The paper's notation $\triangle$ and $\triangledown$ for "always" and "sometimes" is preserved in the Lean implementation as `always` and `sometimes`.

7.2.2 Axiom Naming

Paper Name	Lean Name	Notes
MT (Modal T)	`Axiom.modal_t`	$\Box\varphi \to \varphi$
M4 (Modal 4)	`Axiom.modal_4`	$\Box\varphi \to \Box\Box\varphi$
MB (Modal B)	`Axiom.modal_b`	$\varphi \to \Box\Diamond\varphi$
MK	`Axiom.modal_k_dist`	K distribution
TK	`Axiom.temp_k_dist`	Temporal K distribution
T4	`Axiom.temp_4`	Temporal transitivity
TA	`Axiom.temp_a`	Temporal connectedness
TL	`Axiom.temp_l`	Temporal introspection
MF	`Axiom.modal_future`	Modal-future interaction
TF	`Axiom.temp_future`	Temporal-future interaction

7.2.3 M5 Collapse Axiom

The implementation includes an explicit M5 collapse axiom (`Axiom.modal_5_collapse`):

$$\Diamond\Box\varphi \to \Box\varphi$$

This is derivable from the other S5 axioms (MB + M4) but is included as a primitive for proof convenience in the S5 collapse theorem.

7.2.4 Completeness Status

The paper proves completeness via canonical model construction. The Lean implementation establishes completeness via the semantic canonical model approach. The key results are:

- `set_lindenbaum`: Every consistent set extends to a maximal consistent set
- `semantic_truth_lemma_v2`: Membership corresponds to truth in the semantic model
- `semantic_weak_completeness`: Validity implies derivability
- `main_provable_iff_valid`: Derivability and validity coincide

The semantic approach defines world states as equivalence classes of history-time pairs, making the truth lemma straightforward by construction. Bridge sorries remain for connecting general validity to frame validity in strong completeness.

7.2.5 Decidability Implementation

The implementation includes a tableau-based decision procedure for validity that provides an alternative to the canonical model approach. The decidability module establishes that validity is decidable via constructive tableau expansion and branch closure. Soundness is proven: if the procedure returns "valid", the formula is semantically valid. Completeness requires the Finite Model Property (FMP), which is stated but not yet fully formalized. The FMP states that if a formula is satisfiable, it is satisfiable in a finite model. Full formalization of the FMP completes decidability.