A meta-modal logic for bisimulations

Alfredo Burrieza Fernando Soler-Toscano
Antonio Yuste-Ginel

February 6, 2026

Abstract

Bisimulations are a fundamental formal tool in the model theory of standard modal logic. Roughly speaking, bisimulations provide a clear answer to a foundational model-theoretical question: Given two (Kripke-style) models, what conditions are sufficient and necessary for them to satisfy the same modal formulas? We propose a modal study of the notion of bisimulation. We extend the basic modal language with a new modality $[b]$, whose intended meaning is universal quantification over all states that are bisimilar to the current one. We provide a sound and complete axiomatisation of the class of all pairs of Kripke models linked by bisimulations.

Contents

This theory introduces a modal logic for reasoning about bisimulations (more details on [1]). The proofs rely on various results concerning maximally consistent sets, drawn from the APF entry *Synthetic Completeness* by Asta Halkjær From [2].

theory *Bisimulation-Logic*
imports *Synthetic-Completeness.Derivations*
begin

1 Syntax

First, the language $\mathcal{L}_{\Box[b]}$ is introduced:

```
datatype 'p fm
  = Fls (‹⊥›)
  | Pro 'p (‹•›)
  | Imp ‹'p fm› ‹'p fm› (infixr ‹⟶› 55)
  | Box ‹'p fm› (‹□ -› [70] 70)
  | FrB ‹'p fm› (‹[b] -› [70] 70)
```

Defined connectives.

```
abbreviation Not (‹¬ -› [70] 70) where
  ‹¬ p ≡ p ⟶ ⊥›
abbreviation Tru (‹⊤›) where
  ‹⊤ ≡ ¬⊥›
abbreviation Dis (infixr ‹∨› 60) where
  ‹A ∨ B ≡ ¬A ⟶ B›
abbreviation Con (infixr ‹∧› 65) where
  ‹A ∧ B ≡ ¬(A ⟶ ¬B)›
abbreviation Iff (infixr ‹⟷› 55) where
  ‹A ⟷ B ≡ (A ⟶ B) ∧ (B ⟶ A)›
abbreviation Dia (‹◇ -› [70] 70) where
  ‹◇A ≡ ¬□¬A›
abbreviation FrD (‹⟨b⟩ -› [70] 70) where
  ‹⟨b⟩A ≡ ¬[b]¬A›
```

Iteration of modal operators □ and ◇.

```
primrec chain-b :: ‹nat ⇒ 'p fm ⇒ 'p fm› (‹□^- -› [70, 70] 70) where
  ‹□^0 f = f›
```

| ‹□^(Suc n) f = □(□^n f)›

primrec *chain-d* :: ‹nat ⇒ 'p fm ⇒ 'p fm› (‹◇^- -› [70, 70] 70) **where**
‹◇^0 f = f›
| ‹◇^(Suc n) f = ◇(◇^n f)›

lemma *chain-bd-sum*:
‹□^n (□^m F) = □^(n+m) F› **and**
‹◇^n (◇^m F) = ◇^(n+m) F›
⟨*proof*⟩

2 Semantics

This is the type of both left and right models:

datatype ('p, 'w) *model* =
Model (W: ‹'w set›) (R: ‹('w × 'w) set›) (V: ‹'w ⇒ 'p ⇒ bool›)

Given a model $\mathcal{M}$ W $\mathcal{M}$ denotes its set of worlds, R $\mathcal{M}$ the accessibility relation and V $\mathcal{M}$ the valuation function.

This is the type of a model in $\mathcal{L}_{\Box[b]}$:

datatype ('p, 'w) *modelLb* =
ModelLb (M1: ‹('p, 'w) model›) (M2: ‹('p, 'w) model›) (Z : ‹('w × 'w) set›)

Given a model $\mathfrak{M}$ of $\mathcal{L}_{\Box[b]}$, *M1* $\mathfrak{M}$ denotes the model on the left, *M2* $\mathfrak{M}$ the model on the right and Z $\mathfrak{M}$ the bisimulation relation.

Bi-models are a relevant class of $\mathcal{L}_{\Box[b]}$, as we will prove soundness and completeness of the proof system $\vdash_B$ for bi-models. First, the conditions for $\mathfrak{M}$ to be a *bi−model* are introduced.

definition *bi-model* :: ‹('p, 'w) modelLb ⇒ bool› **where**
‹bi-model 𝔐 ≡
— M1 and M2 have non-empty domains
W (M1 𝔐) ≠ {} ∧ W (M2 𝔐) ≠ {} ∧
— R1 and R2 are defined in the corresponding domains
R (M1 𝔐) ⊆ (W (M1 𝔐)) × (W (M1 𝔐)) ∧
R (M2 𝔐) ⊆ (W (M2 𝔐)) × (W (M2 𝔐)) ∧
— Z is a non-empty relation from W (M1 𝔐) to W (M2 𝔐)
Z 𝔐 ≠ {} ∧ Z 𝔐 ⊆ (W (M1 𝔐)) × (W (M2 𝔐)) ∧
— Atomic harmony
(∀ w w' . (w, w') ∈ Z 𝔐 ⟶ ((V (M1 𝔐)) w) = ((V (M2 𝔐)) w')) ∧
— Forth
(∀ w w' v . (w, w') ∈ Z 𝔐 ∧ (w,v) ∈ R (M1 𝔐) ⟶
(∃ v' . (v,v') ∈ Z 𝔐 ∧ (w', v') ∈ R (M2 𝔐))) ∧
— Back
(∀ w w' v' . (w, w') ∈ Z 𝔐 ∧ (w',v') ∈ R (M2 𝔐) ⟶ (∃ v . (v,v') ∈ Z 𝔐 ∧ (w,v) ∈ R (M1 𝔐)))›

In the semantics, formulas are evaluated differently depending on the pointed world is on the left ($\mathcal{M}$) or on the right ($\mathcal{M}'$). Datatype *ep* (evaluation point) indicates the side of a model in which a given formula is evaluated.

datatype *ep* = m | m′

— Pointed model ($\mathcal{M}$,w).
type-synonym (′*p*, ′*w*) *Mctx* = ‹(′*p*, ′*w*) *model* × ′*w*›

— Pointed $\mathcal{L}_{\Box[b]}$-modelLb ($\mathfrak{M}$,$\mathcal{M}^{(\prime)}$,w).
type-synonym (′*p*, ′*w*) *MLbCtx* = ‹(′*p*, ′*w*) *modelLb* × *ep* × ′*w*›

Definition of truth in a pointed $\mathcal{L}_{\Box[b]}$-modelLb.

fun *semantics* :: ‹(′*p*, ′*w*) *MLbCtx* ⇒ ′*p fm* ⇒ *bool*› (**infix** ‹$\models_B$› *50*) **where**
‹- $\models_B$(⊥::(′*p fm*)) ⟷ *False*›
| ‹($\mathfrak{M}$, m, *w*) $\models_B$ ⋅*P* ⟷ *V* (*M1* $\mathfrak{M}$) *w P*›
| ‹($\mathfrak{M}$, m′, *w*) $\models_B$ ⋅*P* ⟷ *V* (*M2* $\mathfrak{M}$) *w P*›
| ‹($\mathfrak{M}$, *e*, *w*) $\models_B$ *A* ⟶ *B* ⟷ ($\mathfrak{M}$, *e*, *w*) $\models_B$ *A* ⟶ ($\mathfrak{M}$, *e*, *w*) $\models_B$ *B*›
| ‹($\mathfrak{M}$, m, *w*) $\models_B$ □*A* ⟷ (∀ *v* ∈ *W* (*M1* $\mathfrak{M}$) . (*w*,*v*) ∈ *R* (*M1* $\mathfrak{M}$) ⟶ ($\mathfrak{M}$, m, *v*) $\models_B$ *A*)›
| ‹($\mathfrak{M}$, m′, *w*) $\models_B$ □*A* ⟷ (∀ *v* ∈ *W* (*M2* $\mathfrak{M}$) . (*w*,*v*) ∈ *R* (*M2* $\mathfrak{M}$) ⟶ ($\mathfrak{M}$, m′, *v*) $\models_B$ *A*)›
| ‹($\mathfrak{M}$, m, *w*) $\models_B$ [*b*]*A* ⟷ (∀ *w*′ ∈ *W* (*M2* $\mathfrak{M}$) . (*w*,*w*′) ∈ (*Z* $\mathfrak{M}$) ⟶ ($\mathfrak{M}$, m′, *w*′) $\models_B$ *A*)›
| ‹($\mathfrak{M}$, m′, *w*) $\models_B$ [*b*]*A* ⟷ *True*›

3 Calculus

Function *eval* and *tautology* define what is a propositional tautology.

primrec *eval* :: ‹(′*p* ⇒ *bool*) ⇒ (′*p fm* ⇒ *bool*) ⇒ ′*p fm* ⇒ *bool*› **where**
‹*eval* - - ⊥ = *False*›
| ‹*eval g* - (⋅*P*) = *g P*›
| ‹*eval g h* (*A* ⟶ *B*) = (*eval g h A* ⟶ *eval g h B*)›
| ‹*eval* - *h* (□*A*) = *h* (□*A*)›
| ‹*eval* - *h* ([*b*]*A*) = *h* ([*b*]*A*)›

abbreviation ‹*tautology p* ≡ ∀ *g h*. *eval g h p*›

— Example of propositional tautology
lemma ‹*tautology* ([*b*]*A* ∨ ¬[*b*]*A*)› ⟨*proof*⟩

Finally, the axiom system $\vdash_B$ is presented.

inductive *Calculus* :: ‹′*p fm* ⇒ *bool*› (‹$\vdash_B$ -› [*50*] *50*) **where**
TAU: ‹*tautology A* ⟹ $\vdash_B$ *A*›
| *KSq*: ‹$\vdash_B$ □ (*A* ⟶ *B*) ⟶ (□*A* ⟶ □*B*)›
| *Kb*: ‹$\vdash_B$ [*b*](*A* ⟶ *B*) ⟶ ([*b*]*A* ⟶[*b*]*B*)›
| *FORTH*: ‹$\vdash_B$ (⟨*b*⟩*A* ∧ ◊[*b*]*B*) ⟶ ⟨*b*⟩(*A* ∧ ◊*B*)›
| *BACK*: ‹$\vdash_B$ ⟨*b*⟩◊*A* ⟶ ◊⟨*b*⟩*A*›

| *HARM*: ‹$(l = \cdot p \vee l = \neg\cdot p) \Longrightarrow \vdash_B l \longrightarrow [b]l$›
| *NTS*: ‹$\vdash_B [b][b]\bot$›
| *MP*: ‹$\vdash_B A \longrightarrow B \Longrightarrow \vdash_B A \Longrightarrow \vdash_B B$›
| *NSq*: ‹$\vdash_B A \Longrightarrow \vdash_B \Box A$›
| *Nb*: ‹$\vdash_B A \Longrightarrow \vdash_B [b]A$›

Proofs use nested conditionals. Given a list $A = [A_1, ..., A_n]$ of formulas, $A \leadsto B$ represents $A_1 \longrightarrow (A_2 \longrightarrow ... (A_n \longrightarrow B))$.

primrec *imply* :: ‹$'p\ fm\ list \Rightarrow 'p\ fm \Rightarrow 'p\ fm$› (**infixr** ‹$\leadsto$› *56*) **where**
‹$([] \leadsto B) = B$›
| ‹$(A \# \Lambda \leadsto B) = (A \longrightarrow \Lambda \leadsto B)$›

abbreviation *Calculus-assms* (**infix** ‹$\vdash_B$› *50*) **where**
‹$\Lambda \vdash_B A \equiv \vdash_B \Lambda \leadsto A$›

4 Soundness

These lemmas will be used to prove soundness.

lemma *atomic-harm*:
assumes ‹*bi-model* $\mathfrak{M}$›
and $(w,w') \in Z\ \mathfrak{M}$
shows $((V\ (M1\ \mathfrak{M}))\ w\ p) = ((V\ (M2\ \mathfrak{M}))\ w'\ p)$ ⟨*proof*⟩

lemma *eval-semantics*:
assumes ‹*bi-model* $\mathfrak{M}$›
shows ‹$eval\ (V\ (M1\ \mathfrak{M})\ w)\ (\lambda q.\ (\mathfrak{M}, \mathrm{m}, w) \models_B q)\ p = ((\mathfrak{M}, \mathrm{m}, w) \models_B p)$› **and**
‹$eval\ (V\ (M2\ \mathfrak{M})\ w)\ (\lambda q.\ (\mathfrak{M}, \mathrm{m}', w) \models_B q)\ p = ((\mathfrak{M}, \mathrm{m}', w) \models_B p)$›
⟨*proof*⟩

Tautologies are always true.

lemma *tautology*:
assumes ‹*tautology* A›
and ‹*bi-model* $\mathfrak{M}$›
shows ‹$(\mathfrak{M}, e, w) \models_B A$›
⟨*proof*⟩

Axiom FORTH is valid in all worlds in $\mathcal{M}$ of bi-models.

lemma *b-forth*:
assumes ‹*bi-model* $\mathfrak{M}$› **and**
‹$w \in W\ (M1\ \mathfrak{M})$›
shows
‹$(\mathfrak{M}, \mathrm{m}, w) \models_B (\langle b\rangle F \wedge \Diamond[b]G) \longrightarrow \langle b\rangle(F \wedge \Diamond G)$›
⟨*proof*⟩

Axiom FORTH is valid in all worlds on $\mathcal{M}'$ (bi-models).

lemma *b-forth2*:
assumes ‹*bi-model* $\mathfrak{M}$› **and**

‹$w \in W$ ($M2$ $\mathfrak{M}$)›
shows
‹($\mathfrak{M}$, m′, w) $\models_B$ ($\langle b\rangle F \wedge \Diamond[b]G$) $\longrightarrow$ $\langle b\rangle(F \wedge \Diamond G)$›
⟨*proof*⟩

Axiom BACK is valid in all worlds on $\mathcal{M}$ (bi-models).

lemma *b-back*:
assumes ‹*bi-model* $\mathfrak{M}$› **and**
‹$w \in W$ ($M1$ $\mathfrak{M}$)›
shows
‹($\mathfrak{M}$, m, w) $\models_B$ $\langle b\rangle\Diamond F \longrightarrow \Diamond\langle b\rangle F$›
⟨*proof*⟩

Axiom BACK is valid in all worlds on $\mathcal{M}'$ (bi-models).

lemma *b-back2*:
assumes ‹*bi-model* $\mathfrak{M}$› **and**
‹$w \in W$ ($M2$ $\mathfrak{M}$)›
shows
‹($\mathfrak{M}$, m′, w) $\models_B$ $\langle b\rangle\Diamond F \longrightarrow \Diamond\langle b\rangle F$› ⟨*proof*⟩

Soundness theorem

theorem *soundness*:
‹ $\vdash_B A \Longrightarrow$ *bi-model* $\mathfrak{M} \Longrightarrow$
($w \in W$ ($M1$ $\mathfrak{M}$) $\longrightarrow$ ($\mathfrak{M}$, m, w) $\models_B A$) $\wedge$
($w \in W$ ($M2$ $\mathfrak{M}$) $\longrightarrow$ ($\mathfrak{M}$, m′, w) $\models_B A$)›
⟨*proof*⟩

5 Admissible rules

These lemmas are mostly from the AFP entry "Synthetic Completeness" by Asta Halkjær From.

lemma *K-imply-head*: ‹p # $A \vdash_B p$›
⟨*proof*⟩

lemma *K-imply-Cons*:
assumes ‹$A \vdash_B q$›
shows ‹p # $A \vdash_B q$›
⟨*proof*⟩

lemma *K-right-mp*:
assumes ‹$A \vdash_B p$› ‹$A \vdash_B p \longrightarrow q$›
shows ‹$A \vdash_B q$›
⟨*proof*⟩

lemma *deduct1*: ‹$A \vdash_B p \longrightarrow q \Longrightarrow p$ # $A \vdash_B q$›
⟨*proof*⟩

lemma *imply-append* [*iff*]: ‹$(A \ @ \ B \rightsquigarrow r) = (A \rightsquigarrow B \rightsquigarrow r)$›
⟨*proof*⟩

lemma *imply-swap-append*: ‹$A \ @ \ B \vdash_B r \Longrightarrow B \ @ \ A \vdash_B r$›
⟨*proof*⟩

lemma *K-ImpI*: ‹$p \ \# \ A \vdash_B q \Longrightarrow A \vdash_B p \longrightarrow q$›
⟨*proof*⟩

lemma *imply-mem* [*simp*]: ‹$p \in set\ A \Longrightarrow A \vdash_B p$›
⟨*proof*⟩

lemma *add-imply* [*simp*]: ‹$\vdash_B q \Longrightarrow A \vdash_B q$›
⟨*proof*⟩

lemma *K-imply-weaken*: ‹$A \vdash_B q \Longrightarrow set\ A \subseteq set\ A' \Longrightarrow A' \vdash_B q$›
⟨*proof*⟩

lemma *K-Boole*:
assumes ‹$(\neg\ p) \ \# \ A \vdash_B \bot$›
shows ‹$A \vdash_B p$›
⟨*proof*⟩

lemma *MP-chain*:
assumes ‹$\vdash_B A \longrightarrow B$›
and ‹$\vdash_B B \longrightarrow C$›
shows ‹$\vdash_B A \longrightarrow C$›
⟨*proof*⟩

This locale is use to prove common results of normal modal operators. As both □ and [*b*] are normal, result involving **K** in *Kop* will be applied to them.

locale *Kop* =
fixes K :: *'p fm* ⇒ *'p fm* (‹**K** -› [*70*] *70*)
assumes *Kax*: $\vdash_B \mathbf{K}\ (A \longrightarrow B) \longrightarrow (\mathbf{K}\ A \longrightarrow \mathbf{K}\ B)$
and *KN*: $\vdash_B A \Longrightarrow \vdash_B \mathbf{K}\ A$

context *Kop* **begin**

abbreviation P (‹**P** -› [*70*] *70*) **where** ‹$\mathbf{P}\ A \equiv \neg\mathbf{K}\neg A$›

lemma *K-distrib-K-imp*:
assumes ‹$\vdash_B \mathbf{K}\ (A \rightsquigarrow q)$›
shows ‹$map\ (\lambda\ x\ .\ \mathbf{K}\ x)\ A \vdash_B \mathbf{K}\ q$›
⟨*proof*⟩

lemma *Kpos*:
shows ‹$\vdash_B \mathbf{K}(A \longrightarrow B) \longrightarrow (\mathbf{P}A \longrightarrow \mathbf{P}B)$›
⟨*proof*⟩

end

Both $\Box$ and $[b]$ are normal modal operators.

interpretation *KBox*: *Kop* $\lambda\ A\ .\ \Box\ A$
⟨*proof*⟩

interpretation *KFrB*: *Kop* $\lambda\ A\ .\ [b]\ A$
⟨*proof*⟩

Some other useful theorems of $\vdash_B$ that are used in later proofs.

First, the box-version of BACK axiom.

lemma *BACK-rev*:
‹$\vdash_B \Box[b]F \longrightarrow [b]\Box F$›
⟨*proof*⟩
lemma *NTSgen*:
‹$\vdash_B [b]\ \Box\widehat{\ }n\ [b]\bot$›
⟨*proof*⟩

6 Maximal Consistent Sets

These definitions and lemmas are mostly from the AFP entry "Synthetic Completeness" by Asta Halkjær From.

definition *consistent* :: ‹$'p\ fm\ set \Rightarrow bool$› **where**
‹$consistent\ S \equiv \forall A.\ set\ A \subseteq S \longrightarrow \neg\ A \vdash_B \bot$›

interpretation *MCS-No-Witness-UNIV consistent*
⟨*proof*⟩

interpretation *Derivations-Cut-MCS consistent Calculus-assms*
⟨*proof*⟩

interpretation *Derivations-Bot consistent Calculus-assms* ‹$\bot$›
⟨*proof*⟩

interpretation *Derivations-Imp consistent Calculus-assms* ‹$\lambda p\ q.\ p \longrightarrow q$›
⟨*proof*⟩

theorem *deriv-in-maximal*:
assumes ‹*consistent S*› ‹*maximal S*› ‹$\vdash_B p$›
shows ‹$p \in S$›
⟨*proof*⟩

lemma *dia-not-box-bot*:
assumes ‹*consistent S*› ‹*maximal S*› ‹$\langle b\rangle\ F \in S$›
shows ‹$\neg[b]\bot \in S$›
⟨*proof*⟩

Some other useful lemmas that are repeatedly used in proofs.

lemma *not-empty*:
assumes ‹$a \in A$›
shows ‹$A \neq \{\}$›
⟨*proof*⟩

lemma *MPcons*:
assumes ‹$\vdash_B A \longrightarrow (B \longrightarrow C)$›
and ‹$\vdash_B B$›
shows ‹$\vdash_B A \longrightarrow C$›
⟨*proof*⟩

lemma *multiple-MP-MCS*:
assumes ‹*MCS* S›
and ‹*set* $A \subseteq S$›
and ‹$A \leadsto f \in S$›
shows ‹$f \in S$› ⟨*proof*⟩

lemma *not-imp-to-conj*:
assumes ‹*MCS* A›
and ‹$\neg(B \leadsto \bot) \in A$›
shows ‹*set* $B \subseteq A$›
⟨*proof*⟩

Several lemmas of *Kop*, valid for normal modal operators.

context *Kop* **begin**

lemma *not-pos-to-nec-not*:
shows ‹$\vdash_B \neg\mathbf{P}F \longrightarrow \mathbf{K}\neg F$›
⟨*proof*⟩

lemma *not-pos-to-nec-not-deriv*:
assumes ‹$\vdash_B \neg F \longrightarrow G$›
shows ‹$\vdash_B \neg\mathbf{P}F \longrightarrow \mathbf{K}G$›
⟨*proof*⟩

lemma *pos-not-to-not-nec*:
shows ‹$\vdash_B \mathbf{P}\neg F \longrightarrow \neg\mathbf{K}F$›
⟨*proof*⟩

lemma *not-nec-to-pos-not*:
shows ‹$\vdash_B \neg\mathbf{K}F \longrightarrow \mathbf{P}\neg F$›
⟨*proof*⟩

lemma *pos-not-to-not-nec-MCS*:
assumes ‹*MCS* A›
and ‹$\mathbf{P}\neg F \in A$›
shows ‹$\neg\mathbf{K}F \in A$› ⟨*proof*⟩

lemma *pos-subset*:
  assumes ‹*MCS A*› **and** ‹*MCS B*›
  shows ‹$\{ F \mid F . \mathbf{K}\ F \in A\} \subseteq B \longleftrightarrow \{\mathbf{P}F \mid F . F \in B\} \subseteq A$›
⟨*proof*⟩

end

Lemmas involving the negation of a chain of $\Box$ or $\Diamond$.

lemma *not-chain-d-to-chain-b-not*:
  assumes ‹$\vdash_B \neg F \longrightarrow G$›
  shows ‹$\vdash_B \neg (\Diamond\hat{}n\ F) \longrightarrow (\Box\hat{}n\ G)$›
⟨*proof*⟩

lemma *not-chain-b-to-chain-d-not*:
  assumes ‹$\vdash_B \neg F \longrightarrow G$›
  shows ‹$\vdash_B \neg (\Box\hat{}n\ F) \longrightarrow (\Diamond\hat{}n\ G)$›
⟨*proof*⟩

lemma *not-chain-b-to-chain-d-not-rev*:
  assumes ‹$\vdash_B F \longrightarrow \neg G$›
  shows ‹$\vdash_B (\Diamond\hat{}n\ G) \longrightarrow \neg (\Box\hat{}n\ F)$›
⟨*proof*⟩

7 Elements for the Canonical model

First, we introduce some relations that will by used to define the components of the Canonical Model. The first one is the chain relation *Chn*:

abbreviation *Chn* :: ‹(*'p fm set* × *'p fm set*) *set*› **where**
  ‹$Chn \equiv \{(Sa,Sb) . MCS\ Sa \wedge MCS\ Sb \wedge \{f . \Box f \in Sa\} \subseteq Sb\}$›

Now, the eelation *Zmc* linking MCS that will produce the bisimilarity relation:

abbreviation *Zmc* :: ‹(*'p fm set* × *'p fm set*) *set*› **where**
  ‹$Zmc \equiv \{(Sa,Sb) . MCS\ Sa \wedge MCS\ Sb \wedge \{f. [b]f \in Sa\} \subseteq Sb\}$›

Truth of propositional variables in MCS:

abbreviation *Vmc* :: ‹*'p fm set* ⇒ *'p* ⇒ *bool*› **where**
  ‹$Vmc \equiv (\lambda\ S\ P. \cdot P \in S)$›

Sets *MC1* and *MC2* will constitute the worlds on the left and on the right model of the canonical model for $\mathcal{L}_{\Box[b]}$. All mc-sets are in *MC1*, while *MC2* contains only mc-sets containing $\Box\hat{}n[b]\bot$ for all n.

abbreviation *MC1* :: ‹*'p fm set set*› **where**
  ‹$MC1 \equiv \{A . MCS\ A\}$›

abbreviation *MC2* :: ‹′*p fm set set*› **where**
‹$MC2 \equiv \{A\ .\ MCS\ A \land (\forall\ n\ .\ \square\hat{}n\ [b]\bot \in A)\}$›

This lemma shows that *Zmc* goes from worlds not in *MC2* to worlds in *MC2*.

lemma *Z-from-MC1-to-MC2*:
assumes ‹$(A,B) \in Zmc$›
shows ‹$A \notin MC2 \land B \in MC2$›
⟨*proof*⟩

The following lemma is important for the proof of existence. It proves that if $\{f\ .\ \square f \in A\} \subseteq B$ and $A \in MC2$, then $B \in MC2$.

lemma *Chn-from-to-2*:
assumes ‹$(A,B) \in Chn$› **and** ‹$A \in MC2$›
shows ‹$B \in MC2$›
⟨*proof*⟩

Lemmas used to prove existence.

lemma *pos-r1-sub*:
assumes ‹$A \in MC1$› **and** ‹$B \in MC1$›
shows ‹$(A,B) \in Chn \longleftrightarrow \{\Diamond F \mid F\ .\ F \in B\} \subseteq A$›
⟨*proof*⟩

lemma *pos-r2-sub*:
assumes ‹$A \in MC2$› **and** ‹$B \in MC2$›
shows ‹$(A,B) \in Chn \longleftrightarrow \{\Diamond F \mid F\ .\ F \in B\} \subseteq A$›
⟨*proof*⟩

lemma *pos-b-r2-sub*:
assumes ‹$A \in MC1$› **and** ‹$B \in MC2$›
shows ‹$(A,B) \in Zmc \longleftrightarrow \{\langle b\rangle F \mid F\ .\ F \in B\} \subseteq A$›
⟨*proof*⟩

All mc-sets in *MC2* contain $[b]F$ for every F.

lemma *all-box-b-in-MC2*:
assumes ‹$S \in MC2$›
shows ‹$[b]F \in S$›
⟨*proof*⟩

8 Existence

First, we prove a general result for all normal modal operators.

context *Kop* **begin**

lemma *Kop-existence*:
assumes ‹*MCS A*›
and ‹$\mathbf{P}F \in A$›
shows ‹*consistent* $(\{F\} \cup \{G\ .\ \mathbf{K}G \in A\})$›

⟨*proof*⟩

end

lemma *Chn-iff*:
  assumes ‹*MCS A*›
  shows ‹$\square F \in A \longleftrightarrow (\forall\ B\ .\ (A,B) \in Chn \longrightarrow F{\in}B)$›
⟨*proof*⟩

Existence for $\Diamond$ in *MC1*.

lemma *existenceChn-1*:
  assumes ‹$\Diamond F \in A$› **and** ‹$A \in MC1$›
  shows ‹$\exists\ B\ .\ B \in MC1 \wedge \{F\} \cup \{G\ .\ \square G \in A\}{\subseteq}B$›
⟨*proof*⟩

Existence for $\Diamond$ in *MC2*.

lemma *existenceChn-2*:
  assumes ‹$\Diamond F \in A$› **and** ‹$A \in MC2$›
  shows ‹$\exists\ B\ .\ B \in MC2 \wedge \{F\} \cup \{G\ .\ \square G \in A\}{\subseteq}B$›
⟨*proof*⟩

Existence for $\langle b \rangle$ in *MC1*.

lemma *existenceZmc*:
  assumes ‹$\langle b \rangle F \in A$› **and** ‹$A \in MC1$›
  shows ‹$\exists\ B\ .\ B \in MC2 \wedge \{F\} \cup \{G\ .\ [b]G \in A\}{\subseteq}B$›
⟨*proof*⟩

9 Atomic harmony of *Zmc*.

MCS linked by *Zmc* contain the same propositional variables.

lemma *Zmc-atomic-harmony*:
  assumes ‹$(A,B) \in Zmc$›
  shows ‹$\cdot l \in A \longleftrightarrow \cdot l \in B$›
⟨*proof*⟩

10 Forth and Back

First, an auxiliary lemma used in the proofs of forth and back properties of the Canonical Model.

lemma *nec-As-to-nec-conj*:
  assumes ‹$S \in MC1$›
  and ‹$\{[b]f \mid f.\ f \in set\ A\} \subseteq S$›
  shows ‹$[b]\neg(A \rightsquigarrow \bot) \in S$›
⟨*proof*⟩

Lemmas that will be used to prove that the Canonical Model satisfies forth and back properties.

lemma *forth-cm*:
  assumes ‹$(G1, G2) \in Zmc$›
  and ‹$(G1, G3) \in Chn$›
shows ‹$\exists\ G4\ .\ (G3, G4) \in Zmc \wedge (G2, G4) \in Chn$›
⟨*proof*⟩

lemma *back-cm*:
  assumes ‹$(G1, G2) \in Zmc$›
  and ‹$(G2, G3) \in Chn$›
shows ‹$\exists\ G4\ .\ (G1, G4) \in Chn \wedge (G4, G3) \in Zmc$›
⟨*proof*⟩

11 Existence of elements in *Zmc*.

lemma *Zmc-not-empty*:
  ‹$Zmc \neq \{\}$›
⟨*proof*⟩

12 Canonical Model

The Canonical Model is defined in terms of *MC1*, *MC2*, *Chn* and *Zmc*.

R1 and *R2* are the modal acessibility relations of the models on the left and right of the Canonical Model for $\mathcal{L}_{\Box[b]}$. They are defined as restrictions of *Chn* for *MC1* and *MC2*, respectively. The valuation function *Vc* is common for two models, it assigns True to a variable iff it belongs to the corresponding world. The bisimulation relation *Zc* is defined from *Zmc*.

abbreviation *R1* :: ‹$('p\ fm\ set \times 'p\ fm\ set)\ set$› **where**
  ‹$R1 \equiv \{(w1, w2)\ .\ w1 \in MC1 \wedge w2 \in MC1 \wedge (w1, w2) \in Chn\}$›

abbreviation *R2* :: ‹$('p\ fm\ set \times 'p\ fm\ set)\ set$› **where**
  ‹$R2 \equiv \{(w1, w2)\ .\ w1 \in MC2 \wedge w2 \in MC2 \wedge (w1, w2) \in Chn\}$›

abbreviation *Vc* :: ‹$'p\ fm\ set \Rightarrow 'p \Rightarrow bool$› **where**
  ‹$Vc\ w\ p \equiv \cdot p \in w$›

abbreviation *Zc* :: ‹$('p\ fm\ set \times 'p\ fm\ set)\ set$› **where**
  ‹$Zc \equiv \{(w1, w2)\ .\ w1 \in MC1 \wedge w2 \in MC2 \wedge (w1, w2) \in Zmc\}$›

Now, models *Mc1* and *Mc2* are introduced. These are the models on the left and right, of the Canonical Model.

abbreviation *Mc1* :: ‹$('p, 'p\ fm\ set)\ model$› **where**
  ‹$Mc1 \equiv Model\ MC1\ R1\ Vc$›

abbreviation *Mc2* :: ‹(′*p*, ′*p fm set*) *model*› **where**
‹*Mc2* ≡ *Model MC2 R2 Vc*›

Finally, the Canonical Model is introduced.

abbreviation *CanMod* :: ‹(′*p*, ′*p fm set*) *modelLb*› **where**
‹*CanMod* ≡ *ModelLb Mc1 Mc2 Zc*›

lemma *Chn-Rc2*:
‹((*S*,*T*) ∈ *Chn* ∧ *S* ∈ *MC2* ∧ *T* ∈ *MC2*) ⟷ (*S*,*T*) ∈ *R2*› (**is** ‹*?L* ⟷ *?R*›)
⟨*proof*⟩

13 Canonocity

The Canonical Model is a bi-model.

lemma *bi-model-CM*:
‹*bi-model CanMod*›
⟨*proof*⟩

14 Truth Lemma

This is the key lemma for Completeness: a formula F is true in a given world w of the Canonical Model iff $F \in w$.

lemma *Truth-Lemma*:
‹∀ (*S*::′*p fm set*) . (*MCS S* ⟶ ((*S* ∈ *MC1* ⟶ ((*CanMod*, m, *S*)$\models_B$ *F* ⟷ *F* ∈ *S*)) ∧
(*S* ∈ *MC2* ⟶ ((*CanMod*, m′, *S*)$\models_B$ *F* ⟷ *F* ∈ *S*))))› (**is** ‹*?TL F*›)
⟨*proof*⟩

corollary *truth-lemma-MC1*:
assumes ‹*S* ∈ *MC1*›
shows ‹∀ *F* . *F* ∈ *S* ⟷ (*CanMod*, m, *S*) $\models_B$ *F*›
⟨*proof*⟩

corollary *truth-lemma-MC2*:
assumes ‹*S* ∈ *MC2*›
shows ‹∀ *F* . *F* ∈ *S* ⟷ (*CanMod*, m′, *S*) $\models_B$ *F*›
⟨*proof*⟩

15 Completeness

Proof of strong completeness.

theorem *strong-completeness*:
assumes ‹∀ (*M* :: (′*p*, ′*p fm set*) *modelLb*) *ep w* .

($bi\text{-}model\ M \longrightarrow$ (
($w \in W\ (M1\ M) \longrightarrow ((\forall\ \gamma \in set\ \Gamma\ .\ (M,\ \mathrm{m},\ w) \models_B \gamma) \longrightarrow (M,\ \mathrm{m},\ w) \models_B G)) \land$
($w \in W\ (M2\ M) \longrightarrow ((\forall\ \gamma \in set\ \Gamma\ .\ (M,\ \mathrm{m}',\ w) \models_B \gamma) \longrightarrow (M,\ \mathrm{m}',\ w) \models_B G))))$›
shows ‹$\Gamma \vdash_B G$›
⟨*proof*⟩

Definition of validity in bi-models:

abbreviation *bi-model-valid* :: ‹$'p\ fm \Rightarrow bool$› **where**
‹$bi\text{-}model\text{-}valid\ p \equiv \forall\ (M :: ('p,\ 'p\ fm\ set)\ modelLb)\ w.\ bi\text{-}model\ M \longrightarrow$
$((w \in W\ (M1\ M) \longrightarrow (M,\ \mathrm{m},\ w) \models_B p) \land$
$(w \in W\ (M2\ M) \longrightarrow (M,\ \mathrm{m}',\ w) \models_B p))$›

Weak completeness and main result:

corollary *completeness*: ‹$bi\text{-}model\text{-}valid\ p \Longrightarrow \vdash_B p$›
⟨*proof*⟩

theorem *main*: ‹$(bi\text{-}model\text{-}valid\ p) \longleftrightarrow \vdash_B p$›
⟨*proof*⟩

16 Extension of atomic harmony to all formulas in $\mathcal{L}_\Box$

Set of formulas in $\mathcal{L}_\Box$.

inductive-set *Lbox* :: ‹$'p\ fm\ set$› **where**
Fls: ‹$\bot \in Lbox$›
| *Pro*: ‹$\cdot l \in Lbox$›
| *Imp*: ‹$A \in Lbox \Longrightarrow B \in Lbox \Longrightarrow A \longrightarrow B \in Lbox$›
| *Box*: ‹$A \in Lbox \Longrightarrow \Box A \in Lbox$›

Auxiliary lemmas for the induction.

lemma *BotPos*:
shows ‹$\vdash_B \bot \longrightarrow [b]\bot$› ⟨*proof*⟩

lemma *BotNeg*:
shows ‹$\vdash_B \neg\bot \longrightarrow [b]\neg\bot$›
⟨*proof*⟩

lemma *impPos*:
assumes ‹$\vdash_B \neg A \longrightarrow [b]\neg A$›
and ‹$\vdash_B B \longrightarrow [b]B$›
shows ‹$\vdash_B (A \longrightarrow B) \longrightarrow [b](A \longrightarrow B)$›
⟨*proof*⟩

lemma *impNeg*:
assumes ‹$\vdash_B A \longrightarrow [b]A$›

and ‹$\vdash_B \neg B \longrightarrow [b]\neg B$›
shows ‹$\vdash_B \neg(A \longrightarrow B) \longrightarrow [b]\neg(A \longrightarrow B)$›
⟨*proof*⟩

lemma *NSqPos*:
assumes ‹$\vdash_B A \longrightarrow [b]A$›
shows ‹$\vdash_B \Box A \longrightarrow [b]\Box A$›
⟨*proof*⟩

lemma *NSqNeg*:
assumes ‹$\vdash_B \neg A \longrightarrow [b]\neg A$›
shows ‹$\vdash_B \neg\Box A \longrightarrow [b]\neg\Box A$›
⟨*proof*⟩

The following lemma extends atomic harmony (HARM) to all formulas in $\mathcal{L}_\Box$.

lemma *Lbox-harm*:
assumes ‹$A \in Lbox$›
shows‹$\vdash_B A \longrightarrow [b]A$›
⟨*proof*⟩

end

References

[1] A. Burrieza, F. Soler-Toscano, and A. Yuste-Ginel. A meta-modal logic for bisimulations, 2025. https://arxiv.org/abs/2507.15117.

[2] A. H. From. Synthetic completeness. *Archive of Formal Proofs*, January 2023. https://www.isa-afp.org/entries/Synthetic_Completeness.html, Formal proof development.