

Synthetic Completeness

Asta Halkjær From

February 6, 2026

Abstract

In this work, I provide an abstract framework for proving the completeness of a logical calculus using the synthetic method. The synthetic method is based on maximal consistent witnessed sets (MCSs). A set of formulas is consistent (with respect to the calculus) when we cannot derive a contradiction from it. It is maximally consistent when it contains every formula that is consistent with it. For logics where it is relevant, it is witnessed when it contains a witness for every existential formula. To prove completeness using these maximal consistent witnessed sets, we prove a truth lemma: every formula in an MCS has a satisfying model. Here, saturated sets provide a useful stepping stone. These can be seen as characterizations of the MCSs based on simple subformula conditions rather than via the calculus. We then prove that every saturated set gives rise to a satisfying model and that MCSs are saturated sets. Now, assume a valid formula cannot be derived. Then its negation must be consistent and therefore satisfiable. This contradicts validity and the original formula must be derivable.

To start, I build maximal consistent witnessed sets for any logic that satisfies a small set of assumptions. I do this using a transfinite version of Lindenbaum's lemma, which allows me to support languages of any cardinality. I then prove useful abstract results about derivations and refutations as they relate to MCSs. Finally, I show how saturated sets can be derived from the logic's semantics, outlining one way to prove the required truth lemma.

To demonstrate the versatility of the framework, I instantiate it with five different examples. The formalization contains soundness and completeness results for: a propositional tableau calculus, a propositional sequent calculus, an axiomatic system for modal logic, a labelled natural deduction system for hybrid logic and a natural deduction system for first-order logic. The tableau example uses custom Hintikka (downwards saturated) sets based on the calculus, but the other four examples derive their notion of saturation from the semantics in the style of the framework. The hybrid and first-order logic examples rely on witnessed MCSs. This places requirements on the cardinalities of their languages to ensure that there are enough witnesses available. In both cases, the type of witnesses must be infinite and have cardinality at least that of the type of propositional/predicate symbols.

Contents

Abstract	2
Contents	3
1 Maximal Consistent Sets	5
1.1 Utility	5
1.2 Base Locales	6
1.3 Ordinal Locale	6
1.3.1 Lindenbaum Extension	7
1.3.2 Consistency	7
1.3.3 Maximality	8
1.3.4 Witnessing	8
1.4 Locales for Universe Well-Order	8
1.5 Truth Lemma	9
2 Derivations	11
2.1 Derivations	11
2.2 MCSs and Explosion	11
2.3 MCSs and Derivability	12
2.4 Proof Rules	12
3 Refutations	17
3.1 Rearranging Refutations	17
3.2 MCSs and Refutability	17
4 Example: Propositional Tableau Calculus	19
4.1 Syntax	19
4.2 Semantics	19
4.3 Calculus	19
4.4 Soundness	19
4.5 Maximal Consistent Sets	20
4.6 Truth Lemma	20
4.7 Completeness	20
5 Example: Propositional Sequent Calculus	22
5.1 Syntax	22
5.2 Semantics	22
5.3 Calculus	22
5.4 Soundness	23
5.5 Maximal Consistent Sets	23

5.6	Truth Lemma	23
5.7	Completeness	24
6	Example: Modal Logic	25
6.1	Syntax	25
6.2	Semantics	25
6.3	Calculus	25
6.4	Soundness	26
6.5	Admissible rules	26
6.6	Maximal Consistent Sets	27
6.7	Truth Lemma	27
6.8	Completeness	28
7	Example: Hybrid Logic	29
7.1	Syntax	29
7.2	Semantics	29
7.3	Calculus	30
7.4	Soundness	30
7.5	Admissible Rules	30
7.6	Maximal Consistent Sets	31
7.7	Nominals	32
7.8	Truth Lemma	33
7.9	Cardinalities	34
7.10	Completeness	35
8	Example: First-Order Logic	37
8.1	Syntax	37
8.2	Semantics	37
8.3	Operations	37
8.4	Calculus	39
	8.4.1 Weakening	39
8.5	Soundness	40
8.6	Admissible Rules	40
8.7	Maximal Consistent Sets	40
8.8	Truth Lemma	41
8.9	Cardinalities	42
8.10	Completeness	43
	Bibliography	44

Chapter 1

Maximal Consistent Sets

theory *Maximal-Consistent-Sets* **imports** *HOL-Cardinals.Cardinal-Order-Relation* **begin**

1.1 Utility

lemma *Set-Diff-Un*: $\langle X - (Y \cup Z) = X - Y - Z \rangle$

<proof>

lemma *infinite-Diff-fin-Un*: $\langle \text{infinite } (X - Y) \implies \text{finite } Z \implies \text{infinite } (X - (Z \cup Y)) \rangle$

<proof>

lemma *infinite-Diff-subset*: $\langle \text{infinite } (X - A) \implies B \subseteq A \implies \text{infinite } (X - B) \rangle$

<proof>

lemma *finite-bound*:

fixes $X :: \langle 'a :: \text{size} \rangle \text{ set} \rangle$

assumes $\langle \text{finite } X \rangle \langle X \neq \{\} \rangle$

shows $\langle \exists x \in X. \forall y \in X. \text{size } y \leq \text{size } x \rangle$

<proof>

lemma *infinite-UNIV-size*:

fixes $f :: \langle 'a :: \text{size} \rangle \Rightarrow 'a \rangle$

assumes $\langle \bigwedge x. \text{size } x < \text{size } (f x) \rangle$

shows $\langle \text{infinite } (\text{UNIV} :: 'a \text{ set}) \rangle$

<proof>

context *wo-rel* **begin**

lemma *underS-bound*: $\langle a \in \text{underS } c \implies b \in \text{underS } c \implies a \in \text{under } b \vee b \in \text{under } a \rangle$

<proof>

lemma *finite-underS-bound*:

assumes $\langle \text{finite } X \rangle \langle X \subseteq \text{underS } c \rangle \langle X \neq \{\} \rangle$

shows $\langle \exists a \in X. \forall b \in X. b \in \text{under } a \rangle$

<proof>

lemma *finite-bound-under*:

assumes $\langle \text{finite } p \rangle \langle p \subseteq (\bigcup a \in \text{Field } r. f a) \rangle$

shows $\langle \exists b. p \subseteq (\bigcup a \in \text{under } b. f a) \rangle$

<proof>

lemma *underS-trans*: $\langle a \in \text{underS } b \implies b \in \text{underS } c \implies a \in \text{underS } c \rangle$
 $\langle \text{proof} \rangle$

end

lemma *card-of-infinite-smaller-Union*:
assumes $\langle \forall x. |f x| <_o |X| \rangle \langle \text{infinite } X \rangle$
shows $\langle |\bigcup x \in X. f x| \leq_o |X| \rangle$
 $\langle \text{proof} \rangle$

lemma *card-of-params-marker-lists*:
assumes $\langle \text{infinite } (UNIV :: 'i \text{ set}) \rangle \langle |\text{UNIV} :: 'm \text{ set}| \leq_o |\text{UNIV} :: \text{nat set}| \rangle$
shows $\langle |\text{UNIV} :: ('i + 'm \times \text{nat}) \text{ list set}| \leq_o |\text{UNIV} :: 'i \text{ set}| \rangle$
 $\langle \text{proof} \rangle$

1.2 Base Locales

locale *MCS-Base* =
fixes *consistent* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$
assumes *consistent-hereditary*: $\langle \bigwedge S S'. \text{consistent } S \implies S' \subseteq S \implies \text{consistent } S' \rangle$
and *inconsistent-finite*: $\langle \bigwedge S. \neg \text{consistent } S \implies \exists S' \subseteq S. \text{finite } S' \wedge \neg \text{consistent } S' \rangle$
begin

definition *maximal* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$ **where**
 $\langle \text{maximal } S \equiv \forall p. \text{consistent } (\{p\} \cup S) \longrightarrow p \in S \rangle$

end

locale *MCS-Witness* = *MCS-Base consistent*
for *consistent* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$ +
fixes *witness* :: $\langle 'a \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set} \rangle$
and *params* :: $\langle 'a \Rightarrow 'i \text{ set} \rangle$
assumes *finite-params*: $\langle \bigwedge p. \text{finite } (\text{params } p) \rangle$
and *finite-witness-params*: $\langle \bigwedge p S. \text{finite } (\bigcup q \in \text{witness } p S. \text{params } q) \rangle$
and *consistent-witness*: $\langle \bigwedge p S. \text{consistent } (\{p\} \cup S) \implies \text{infinite } (UNIV - (\bigcup q \in S. \text{params } q)) \implies \text{consistent } (\{p\} \cup S \cup \text{witness } p S) \rangle$
begin

definition *witnessed* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$ **where**
 $\langle \text{witnessed } S \equiv \forall p \in S. \exists S'. \text{witness } p S' \subseteq S \rangle$

abbreviation *MCS* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$ **where**
 $\langle \text{MCS } S \equiv \text{consistent } S \wedge \text{maximal } S \wedge \text{witnessed } S \rangle$

end

locale *MCS-No-Witness* = *MCS-Base consistent* **for** *consistent* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$

sublocale *MCS-No-Witness* \subseteq *MCS-Witness consistent* $\langle \lambda - . \{\} \rangle \langle \lambda - . \{\} \rangle$
 $\langle \text{proof} \rangle$

1.3 Ordinal Locale

locale *MCS-Lim-Ord* = *MCS-Witness consistent witness params*

for *consistent* :: $\langle 'a \text{ set} \Rightarrow \text{bool} \rangle$
and *witness* :: $\langle 'a \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set} \rangle$
and *params* :: $\langle 'a \Rightarrow 'i \text{ set} \rangle +$
fixes *r* :: $\langle 'a \text{ rel} \rangle$
assumes *Cinfinite-r*: $\langle \text{Cinfinite } r \rangle$
begin

lemma *WELL*: $\langle \text{Well-order } r \rangle$
 $\langle \text{proof} \rangle$

lemma *wo-rel-r*: $\langle \text{wo-rel } r \rangle$
 $\langle \text{proof} \rangle$

lemma *isLimOrd-r*: $\langle \text{isLimOrd } r \rangle$
 $\langle \text{proof} \rangle$

lemma *nonempty-Field-r*: $\langle \text{Field } r \neq \{\} \rangle$
 $\langle \text{proof} \rangle$

1.3.1 Lindenbaum Extension

abbreviation *paramss* :: $\langle 'a \text{ set} \Rightarrow 'i \text{ set} \rangle$ **where**
 $\langle \text{paramss } S \equiv \bigcup p \in S. \text{ params } p \rangle$

definition *extendS* :: $\langle 'a \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set} \rangle$ **where**
 $\langle \text{extendS } a \text{ prev} \equiv \text{if consistent } (\{a\} \cup \text{prev}) \text{ then } \{a\} \cup \text{prev} \cup \text{witness } a \text{ prev else prev} \rangle$

definition *extendL* :: $\langle ('a \Rightarrow 'a \text{ set}) \Rightarrow 'a \Rightarrow 'a \text{ set} \rangle$ **where**
 $\langle \text{extendL } \text{rec } a \equiv \bigcup b \in \text{underS } r \ a. \text{ rec } b \rangle$

definition *extend* :: $\langle 'a \text{ set} \Rightarrow 'a \Rightarrow 'a \text{ set} \rangle$ **where**
 $\langle \text{extend } S \ a \equiv \text{worecZSL } r \ S \ \text{extendS } \ \text{extendL } \ a \rangle$

lemma *adm-woL-extendL*: $\langle \text{adm-woL } r \ \text{extendL} \rangle$
 $\langle \text{proof} \rangle$

definition *Extend* :: $\langle 'a \text{ set} \Rightarrow 'a \text{ set} \rangle$ **where**
 $\langle \text{Extend } S \equiv \bigcup a \in \text{Field } r. \ \text{extend } S \ a \rangle$

lemma *extend-subset*: $\langle a \in \text{Field } r \Longrightarrow S \subseteq \text{extend } S \ a \rangle$
 $\langle \text{proof} \rangle$

lemma *Extend-subset*: $\langle S \subseteq \text{Extend } S \rangle$
 $\langle \text{proof} \rangle$

lemma *extend-underS*: $\langle b \in \text{underS } r \ a \Longrightarrow \text{extend } S \ b \subseteq \text{extend } S \ a \rangle$
 $\langle \text{proof} \rangle$

lemma *extend-under*: $\langle b \in \text{under } r \ a \Longrightarrow \text{extend } S \ b \subseteq \text{extend } S \ a \rangle$
 $\langle \text{proof} \rangle$

1.3.2 Consistency

lemma *params-origin*:
assumes $\langle x \in \text{paramss } (\text{extend } S \ a) \rangle$
shows $\langle x \in \text{paramss } S \vee (\exists b \in \text{underS } r \ a. x \in \text{paramss } (\{b\} \cup \text{witness } b \ (\text{extend } S \ b))) \rangle$

⟨proof⟩

lemma *consistent-extend*:

assumes ⟨consistent S ⟩ ⟨ $r \leq o \mid UNIV - params\ S$ ⟩

shows ⟨consistent (extend S a)⟩

⟨proof⟩

lemma *consistent-Extend*:

assumes ⟨consistent S ⟩ ⟨ $r \leq o \mid UNIV - params\ S$ ⟩

shows ⟨consistent (Extend S)⟩

⟨proof⟩

lemma *Extend-bound*: ⟨ $a \in Field\ r \implies extend\ S\ a \subseteq Extend\ S$ ⟩

⟨proof⟩

1.3.3 Maximality

definition *maximal'* :: ⟨'a set \implies bool⟩ **where**

⟨*maximal'* $S \equiv \forall p \in Field\ r. consistent\ (\{p\} \cup S) \implies p \in S$ ⟩

lemma *maximal'-Extend*: ⟨*maximal'* (Extend S)⟩

⟨proof⟩

1.3.4 Witnessing

definition *witnessed'* :: ⟨'a set \implies bool⟩ **where**

⟨*witnessed'* $S \equiv \forall p \in Field\ r. p \in S \implies (\exists S'. witness\ p\ S' \subseteq S)$ ⟩

lemma *witnessed'-Extend*:

assumes ⟨consistent (Extend S)⟩

shows ⟨*witnessed'* (Extend S)⟩

⟨proof⟩

end

1.4 Locales for Universe Well-Order

locale *MCS-Witness-UNIV* = *MCS-Witness consistent witness params*

for *consistent* :: ⟨'a set \implies bool⟩

and *witness* :: ⟨'a \implies 'a set \implies 'a set⟩

and *params* :: ⟨'a \implies 'i set⟩ +

assumes *infinite-UNIV*: ⟨*infinite* (UNIV :: 'a set)⟩

sublocale *MCS-Witness-UNIV* \subseteq *MCS-Lim-Ord consistent witness params* ⟨ $|UNIV|$ ⟩

⟨proof⟩

context *MCS-Witness-UNIV* **begin**

lemma *maximal-maximal'*: ⟨*maximal* $S \longleftrightarrow$ *maximal'* S ⟩

⟨proof⟩

lemma *maximal-Extend*: ⟨*maximal* (Extend S)⟩

⟨proof⟩

lemma *witnessed-witnessed'*: ⟨*witnessed* $S \longleftrightarrow$ *witnessed'* S ⟩

⟨proof⟩

lemma *witnessed-Extend*:

assumes ⟨consistent (Extend S)⟩

shows ⟨witnessed (Extend S)⟩

⟨proof⟩

theorem *MCS-Extend*:

assumes ⟨consistent S⟩ ⟨|UNIV :: 'a set| ≤ o |UNIV - paramss S|⟩

shows ⟨MCS (Extend S)⟩

⟨proof⟩

end

locale *MCS-No-Witness-UNIV = MCS-No-Witness consistent*

for consistent :: ⟨'a set ⇒ bool⟩ +

assumes infinite-UNIV' [simp]: ⟨infinite (UNIV :: 'a set)⟩

sublocale *MCS-No-Witness-UNIV ⊆ MCS-Witness-UNIV consistent* ⟨λ-. {}⟩ ⟨λ-. {}⟩

⟨proof⟩

context *MCS-No-Witness-UNIV*

begin

theorem *MCS-Extend'*:

assumes ⟨consistent S⟩

shows ⟨MCS (Extend S)⟩

⟨proof⟩

end

1.5 Truth Lemma

locale *Truth-Base =*

fixes semics :: ⟨'model ⇒ ('model ⇒ 'fm ⇒ bool) ⇒ 'fm ⇒ bool⟩ (⟨(- [[-]] -)⟩ [55, 0, 55] 55)

and semantics :: ⟨'model ⇒ 'fm ⇒ bool⟩ (**infix** <|=⟩ 50)

and M :: ⟨'a set ⇒ 'model set⟩

and R :: ⟨'a set ⇒ 'model ⇒ 'fm ⇒ bool⟩

assumes semics-semantics: ⟨M ⊨ p ⟷ M [[|=]] p⟩

begin

abbreviation saturated :: ⟨'a set ⇒ bool⟩ **where**

⟨saturated S ≡ ∀ p. ∀ M ∈ M(S). M [[R(S)]] p ⟷ R(S) M p⟩

end

locale *Truth-Witness = Truth-Base semics semantics M R + MCS-Witness consistent witness params*

for semics :: ⟨'model ⇒ ('model ⇒ 'fm ⇒ bool) ⇒ 'fm ⇒ bool⟩ (⟨(- [[-]] -)⟩ [55, 0, 55] 55)

and semantics :: ⟨'model ⇒ 'fm ⇒ bool⟩ (**infix** <|=⟩ 50)

and M :: ⟨'a set ⇒ 'model set⟩

and R :: ⟨'a set ⇒ 'model ⇒ 'fm ⇒ bool⟩

and consistent :: ⟨'a set ⇒ bool⟩

and witness :: ⟨'a ⇒ 'a set ⇒ 'a set⟩

and params :: ⟨'a ⇒ 'i set⟩ +

assumes saturated-semantics: ⟨∧ S M p. saturated S ⇒ M ∈ M(S) ⇒ M ⊨ p ⟷ R(S) M p⟩

and *MCS-saturated*: $\langle \bigwedge S. \text{MCS } S \implies \text{saturated } S \rangle$
begin

theorem *truth-lemma*:

assumes $\langle \text{MCS } S \rangle \langle M \in \mathcal{M}(S) \rangle$

shows $\langle M \models p \longleftrightarrow \mathcal{R}(S) \ M \ p \rangle$

$\langle \text{proof} \rangle$

end

locale *Truth-No-Witness = Truth-Witness semics semantics* $\mathcal{M} \ \mathcal{R} \ \text{consistent}$ $\langle \lambda-. \{\} \rangle \langle \lambda-. \{\} \rangle$

for *semics* :: $\langle 'model \Rightarrow ('model \Rightarrow 'fm \Rightarrow bool) \Rightarrow 'fm \Rightarrow bool \rangle$

and *semantics* :: $\langle 'model \Rightarrow 'fm \Rightarrow bool \rangle$

and \mathcal{M} :: $\langle 'a \ \text{set} \Rightarrow 'model \ \text{set} \rangle$

and \mathcal{R} :: $\langle 'a \ \text{set} \Rightarrow 'model \Rightarrow 'fm \Rightarrow bool \rangle$

and *consistent* :: $\langle 'a \ \text{set} \Rightarrow bool \rangle$

end

Chapter 2

Derivations

theory *Derivations* **imports** *Maximal-Consistent-Sets* **begin**

lemma *split-finite-sets*:

assumes $\langle \text{finite } A \rangle \langle \text{finite } B \rangle$

and $\langle A \subseteq B \cup S \rangle$

shows $\langle \exists B' C. \text{finite } C \wedge A = B' \cup C \wedge B' = A \cap B \wedge C \subseteq S \rangle$

$\langle \text{proof} \rangle$

lemma *split-list*:

assumes $\langle \text{set } A \subseteq \text{set } B \cup S \rangle$

shows $\langle \exists B' C. \text{set } (B' @ C) = \text{set } A \wedge \text{set } B' = \text{set } A \cap \text{set } B \wedge \text{set } C \subseteq S \rangle$

$\langle \text{proof} \rangle$

2.1 Derivations

locale *Derivations* =

fixes *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50)

assumes *derive-assm* [*simp*]: $\langle \bigwedge A p. p \in \text{set } A \Longrightarrow A \vdash p \rangle$

and *derive-set*: $\langle \bigwedge A B r. A \vdash r \Longrightarrow \text{set } A = \text{set } B \Longrightarrow B \vdash r \rangle$

begin

theorem *derive-split*:

assumes $\langle \text{set } A \subseteq \text{set } B \cup X \rangle \langle A \vdash p \rangle$

shows $\langle \exists B' C. \text{set } B' = \text{set } A \cap \text{set } B \wedge \text{set } C \subseteq X \wedge B' @ C \vdash p \rangle$

$\langle \text{proof} \rangle$

corollary *derive-split1*:

assumes $\langle \text{set } A \subseteq \{q\} \cup X \rangle \langle A \vdash p \rangle \langle q \in \text{set } A \rangle$

shows $\langle \exists C. \text{set } C \subseteq X \wedge q \# C \vdash p \rangle$

$\langle \text{proof} \rangle$

end

2.2 MCSs and Explosion

locale *Derivations-MCS* = *MCS-Base consistent* + *Derivations derive*

for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$

and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50) +

assumes *consistent-underivable*: $\langle \bigwedge S. \text{consistent } S \longleftrightarrow (\forall A. \text{set } A \subseteq S \longrightarrow (\exists q. \neg A \vdash q)) \rangle$

begin

theorem *MCS-explode*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle p \notin S \iff (\exists A. \text{set } A \subseteq S \wedge (\forall q. p \# A \vdash q)) \rangle$
 $\langle \text{proof} \rangle$

end

2.3 MCSs and Derivability

locale *Derivations-Cut-MCS = Derivations-MCS consistent derive*
for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$
and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50) +
assumes *derive-cut*: $\langle \bigwedge A B p q. A \vdash p \implies p \# B \vdash q \implies A @ B \vdash q \rangle$
begin

theorem *MCS-derive*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle p \in S \iff (\exists A. \text{set } A \subseteq S \wedge A \vdash p) \rangle$
 $\langle \text{proof} \rangle$

end

2.4 Proof Rules

locale *Derivations-Bot = Derivations-Cut-MCS consistent derive*
for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$
and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50) +
fixes *bot* :: $\langle 'fm \rangle$ ($\langle \perp \rangle$)
assumes *botE*: $\langle \bigwedge A p. A \vdash \perp \implies A \vdash p \rangle$
begin

corollary *MCS-botE [elim]*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
and $\langle \perp \in S \rangle$
shows $\langle p \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-bot [simp]*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle \perp \notin S \rangle$
 $\langle \text{proof} \rangle$

end

locale *Derivations-Top = Derivations-Cut-MCS +*
fixes *top* ($\langle \top \rangle$)
assumes *topI*: $\langle \bigwedge A. A \vdash \top \rangle$
begin

corollary *MCS-topI [simp]*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle \top \in S \rangle$
 $\langle \text{proof} \rangle$

end

locale *Derivations-Not = Derivations-Bot consistent derive bot*

for *consistent* :: $\langle 'fm\ set \Rightarrow bool \rangle$
and *derive* :: $\langle 'fm\ list \Rightarrow 'fm \Rightarrow bool \rangle$ (**infix** $\langle \vdash \rangle$ 50)
and *bot* :: $\langle 'fm \langle \perp \rangle \rangle +$
fixes *not* :: $\langle 'fm \Rightarrow 'fm \rangle \langle \neg \rangle$
assumes
notI: $\langle \bigwedge A\ p.\ p \# A \vdash \perp \Longrightarrow A \vdash \neg p \rangle$ **and**
notE: $\langle \bigwedge A\ p.\ A \vdash p \Longrightarrow A \vdash \neg p \Longrightarrow A \vdash \perp \rangle$

begin

corollary *MCS-not-xor*:

assumes $\langle consistent\ S \rangle \langle maximal\ S \rangle$
shows $\langle p \in S \longleftrightarrow \neg p \notin S \rangle$

$\langle proof \rangle$

corollary *MCS-not-both*:

assumes $\langle consistent\ S \rangle \langle maximal\ S \rangle$
shows $\langle p \notin S \vee \neg p \notin S \rangle$

$\langle proof \rangle$

corollary *MCS-not-neither*:

assumes $\langle consistent\ S \rangle \langle maximal\ S \rangle$
shows $\langle p \in S \vee \neg p \in S \rangle$

$\langle proof \rangle$

end

locale *Derivations-Con = Derivations-Cut-MCS consistent derive*

for *consistent* :: $\langle 'fm\ set \Rightarrow bool \rangle$
and *derive* :: $\langle 'fm\ list \Rightarrow 'fm \Rightarrow bool \rangle$ (**infix** $\langle \vdash \rangle$ 50) +
fixes *con* :: $\langle 'fm \Rightarrow 'fm \Rightarrow 'fm \rangle \langle \neg \wedge \neg \rangle$
assumes
conI: $\langle \bigwedge A\ p\ q.\ A \vdash p \Longrightarrow A \vdash q \Longrightarrow A \vdash (p \wedge q) \rangle$ **and**
conE: $\langle \bigwedge A\ p\ q\ r.\ A \vdash (p \wedge q) \Longrightarrow p \# q \# A \vdash r \Longrightarrow A \vdash r \rangle$

begin

corollary *MCS-conI [intro]*:

assumes $\langle consistent\ S \rangle \langle maximal\ S \rangle$
and $\langle p \in S \rangle \langle q \in S \rangle$

shows $\langle (p \wedge q) \in S \rangle$

$\langle proof \rangle$

corollary *MCS-conE [dest]*:

assumes $\langle consistent\ S \rangle \langle maximal\ S \rangle$
and $\langle (p \wedge q) \in S \rangle$

shows $\langle p \in S \wedge q \in S \rangle$

$\langle proof \rangle$

corollary *MCS-con*:

assumes $\langle consistent\ S \rangle \langle maximal\ S \rangle$

shows $\langle (p \wedge q) \in S \longleftrightarrow p \in S \wedge q \in S \rangle$

$\langle proof \rangle$

end

locale *Derivations-Dis = Derivations-Cut-MCS consistent derive*
for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$
and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50) +
fixes *dis* :: $\langle 'fm \Rightarrow 'fm \Rightarrow 'fm \rangle$ ($\langle \cdot \vee \cdot \rangle$)
assumes
disI1: $\langle \bigwedge A \ p \ q. A \vdash p \Longrightarrow A \vdash (p \vee q) \rangle$ **and**
disI2: $\langle \bigwedge A \ p \ q. A \vdash q \Longrightarrow A \vdash (p \vee q) \rangle$ **and**
disE: $\langle \bigwedge A \ p \ q \ r. A \vdash (p \vee q) \Longrightarrow p \# A \vdash r \Longrightarrow q \# A \vdash r \Longrightarrow A \vdash r \rangle$
begin

corollary *MCS-disI1 [intro]*:
assumes $\langle \text{consistent } S \rangle$ $\langle \text{maximal } S \rangle$
and $\langle p \in S \rangle$
shows $\langle (p \vee q) \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-disI2 [intro]*:
assumes $\langle \text{consistent } S \rangle$ $\langle \text{maximal } S \rangle$
and $\langle q \in S \rangle$
shows $\langle (p \vee q) \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-disE [elim]*:
assumes $\langle \text{consistent } S \rangle$ $\langle \text{maximal } S \rangle$
and $\langle (p \vee q) \in S \rangle$
shows $\langle p \in S \vee q \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-dis*:
assumes $\langle \text{consistent } S \rangle$ $\langle \text{maximal } S \rangle$
shows $\langle (p \vee q) \in S \longleftrightarrow p \in S \vee q \in S \rangle$
 $\langle \text{proof} \rangle$

end

locale *Derivations-Imp = Derivations-Cut-MCS consistent derive*
for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$
and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50) +
fixes *imp* :: $\langle 'fm \Rightarrow 'fm \Rightarrow 'fm \rangle$ ($\langle \cdot \rightarrow \cdot \rangle$)
assumes
impI: $\langle \bigwedge A \ p \ q. p \# A \vdash q \Longrightarrow A \vdash (p \rightarrow q) \rangle$ **and**
impE: $\langle \bigwedge A \ p \ q. A \vdash p \Longrightarrow A \vdash (p \rightarrow q) \Longrightarrow A \vdash q \rangle$
begin

corollary *MCS-impI [intro]*:
assumes $\langle \text{consistent } S \rangle$ $\langle \text{maximal } S \rangle$
and $\langle p \in S \longrightarrow q \in S \rangle$
shows $\langle (p \rightarrow q) \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-impE [dest]*:
assumes $\langle \text{consistent } S \rangle$ $\langle \text{maximal } S \rangle$
and $\langle (p \rightarrow q) \in S \rangle$ $\langle p \in S \rangle$
shows $\langle q \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-imp*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle (p \rightarrow q) \in S \longleftrightarrow (p \in S \longrightarrow q \in S) \rangle$
 $\langle \text{proof} \rangle$

end

locale *Derivations-Exi = MCS-Witness consistent witness params + Derivations-Cut-MCS consistent derive*

for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$
and *witness params*
and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50) +
fixes *exi* :: $\langle 'fm \Rightarrow 'fm \rangle$ ($\langle \exists \rangle$)
and *inst* :: $\langle 't \Rightarrow 'fm \Rightarrow 'fm \rangle$ ($\langle \langle - \rangle \rangle$)
assumes
exi-witness: $\langle \bigwedge S S' p. \text{MCS } S \Longrightarrow \text{witness } (\exists p) S' \subseteq S \Longrightarrow \exists t. \langle t \rangle p \in S \rangle$ **and**
exiI: $\langle \bigwedge A p t. A \vdash \langle t \rangle p \Longrightarrow A \vdash \exists p \rangle$

begin

corollary *MCS-exiI [intro]*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
and $\langle \langle t \rangle p \in S \rangle$
shows $\langle \exists p \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-exiE [dest]*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle \text{witnessed } S \rangle$
and $\langle \exists p \in S \rangle$
shows $\langle \exists t. \langle t \rangle p \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-exi*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle \text{witnessed } S \rangle$
shows $\langle \exists p \in S \longleftrightarrow (\exists t. \langle t \rangle p \in S) \rangle$
 $\langle \text{proof} \rangle$

end

locale *Derivations-Uni = MCS-Witness consistent witness params + Derivations-Not consistent derive bot not*

for *consistent* :: $\langle 'fm \text{ set} \Rightarrow \text{bool} \rangle$
and *witness params*
and *derive* :: $\langle 'fm \text{ list} \Rightarrow 'fm \Rightarrow \text{bool} \rangle$ (**infix** $\langle \vdash \rangle$ 50)
and *bot* :: $\langle 'fm \rangle$ ($\langle \perp \rangle$)
and *not* :: $\langle 'fm \Rightarrow 'fm \rangle$ ($\langle \neg \rangle$) +
fixes *uni* :: $\langle 'fm \Rightarrow 'fm \rangle$ ($\langle \forall \rangle$)
and *inst* :: $\langle 't \Rightarrow 'fm \Rightarrow 'fm \rangle$ ($\langle \langle - \rangle \rangle$)
assumes
uni-witness: $\langle \bigwedge S S' p. \text{MCS } S \Longrightarrow \text{witness } (\neg (\forall p)) S' \subseteq S \Longrightarrow \exists t. \neg (\langle t \rangle p) \in S \rangle$ **and**
uniE: $\langle \bigwedge A p t. A \vdash \forall p \Longrightarrow A \vdash \langle t \rangle p \rangle$

begin

corollary *MCS-uniE [dest]*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
and $\langle \forall p \in S \rangle$

shows $\langle \langle t \rangle p \in S \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-uniI* [*intro*]:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle \text{witnessed } S \rangle$

and $\langle \forall t. \langle t \rangle p \in S \rangle$

shows $\langle \forall p \in S \rangle$

$\langle \text{proof} \rangle$

corollary *MCS-uni*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle \text{witnessed } S \rangle$

shows $\langle \forall p \in S \longleftrightarrow (\forall t. \langle t \rangle p \in S) \rangle$

$\langle \text{proof} \rangle$

end

end

Chapter 3

Refutations

theory *Refutations* **imports** *Maximal-Consistent-Sets* **begin**

lemma *split-finite-sets*:

assumes $\langle \text{finite } A \rangle \langle \text{finite } B \rangle$

and $\langle A \subseteq B \cup S \rangle$

shows $\langle \exists B' C. \text{finite } C \wedge A = B' \cup C \wedge B' = A \cap B \wedge C \subseteq S \rangle$

$\langle \text{proof} \rangle$

lemma *split-list*:

assumes $\langle \text{set } A \subseteq \text{set } B \cup S \rangle$

shows $\langle \exists B' C. \text{set } (B' @ C) = \text{set } A \wedge \text{set } B' = \text{set } A \cap \text{set } B \wedge \text{set } C \subseteq S \rangle$

$\langle \text{proof} \rangle$

3.1 Rearranging Refutations

locale *Refutations* =

fixes *refute* :: $\langle \text{fm list} \Rightarrow \text{bool} \rangle$

assumes *refute-set*: $\langle \bigwedge A B. \text{refute } A \Longrightarrow \text{set } A = \text{set } B \Longrightarrow \text{refute } B \rangle$

begin

theorem *refute-split*:

assumes $\langle \text{set } A \subseteq \text{set } B \cup X \rangle \langle \text{refute } A \rangle$

shows $\langle \exists B' C. \text{set } B' = \text{set } A \cap \text{set } B \wedge \text{set } C \subseteq X \wedge \text{refute } (B' @ C) \rangle$

$\langle \text{proof} \rangle$

corollary *refute-split1*:

assumes $\langle \text{set } A \subseteq \{q\} \cup X \rangle \langle \text{refute } A \rangle \langle q \in \text{set } A \rangle$

shows $\langle \exists C. \text{set } C \subseteq X \wedge \text{refute } (q \# C) \rangle$

$\langle \text{proof} \rangle$

end

3.2 MCSs and Refutability

locale *Refutations-MCS* = *MCS-Base* + *Refutations* +

assumes *consistent-refute*: $\langle \bigwedge S. \text{consistent } S \longleftrightarrow (\forall A. \text{set } A \subseteq S \longrightarrow \neg \text{refute } A) \rangle$

begin

theorem *MCS-refute*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$

shows $\langle p \notin S \longleftrightarrow (\exists A. \text{set } A \subseteq S \wedge \text{refute } (p \# A)) \rangle$
<proof>

end

end

Chapter 4

Example: Propositional Tableau Calculus

theory *Example-Propositional-Tableau* **imports** *Refutations* **begin**

4.1 Syntax

```
datatype 'p fm
  = Pro 'p (<·>)
  | Neg <'p fm> (<¬ -> [70] 70)
  | Imp <'p fm> <'p fm> (infixr <⟶> 55)
```

4.2 Semantics

```
type-synonym 'p model = <'p ⇒ bool>
```

```
fun semantics :: <'p model ⇒ 'p fm ⇒ bool> (infix <|=T> 50) where
  <I |=T ·P ⟷ I P>
| <I |=T ¬ p ⟷ ¬ I |=T p>
| <I |=T p ⟶ q ⟷ I |=T p ⟶ I |=T q>
```

4.3 Calculus

```
inductive Calculus :: <'p fm list ⇒ bool> (<⊢T -> [50] 50) where
  Axiom [simp]: <⊢T ·P # ¬ ·P # A>
| NegI [intro]: <⊢T p # A ⟹ ⊢T ¬ ¬ p # A>
| ImpP [intro]: <⊢T ¬ p # A ⟹ ⊢T q # A ⟹ ⊢T (p ⟶ q) # A>
| ImpN [intro]: <⊢T p # ¬ q # A ⟹ ⊢T ¬ (p ⟶ q) # A>
| Weak: <⊢T A ⟹ set A ⊆ set B ⟹ ⊢T B>
```

lemma *Weak2*:

```
assumes <⊢T p # A> <⊢T q # B>
shows <⊢T p # A @ B ∧ ⊢T q # A @ B>
<proof>
```

4.4 Soundness

```
theorem soundness: <⊢T A ⟹ ∃ p ∈ set A. ¬ I |=T p>
<proof>
```

corollary *soundness'*: $\langle \vdash_T [\neg p] \implies I \models_T p \rangle$
 $\langle \text{proof} \rangle$

corollary $\langle \neg \vdash_T [] \rangle$
 $\langle \text{proof} \rangle$

4.5 Maximal Consistent Sets

definition *consistent* :: $\langle 'p \text{ fm set} \implies \text{bool} \rangle$ **where**
 $\langle \text{consistent } S \equiv \forall A. \text{ set } A \subseteq S \longrightarrow \neg \vdash_T A \rangle$

interpretation *MCS-No-Witness-UNIV consistent*
 $\langle \text{proof} \rangle$

interpretation *Refutations-MCS consistent Calculus*
 $\langle \text{proof} \rangle$

4.6 Truth Lemma

abbreviation (*input*) *canonical* :: $\langle 'p \text{ fm set} \implies 'p \text{ model} \rangle$ ($\langle \mathcal{M}_T \rangle$) **where**
 $\langle \mathcal{M}_T(S) \equiv \lambda P. \cdot P \in S \rangle$

locale *Hintikka* =
fixes $H :: \langle 'a \text{ fm set} \rangle$
assumes *AxiomH*: $\langle \bigwedge P. \cdot P \in H \implies \neg \cdot P \in H \implies \text{False} \rangle$
and *NegIH*: $\langle \bigwedge p. \neg \neg p \in H \implies p \in H \rangle$
and *ImpPH*: $\langle \bigwedge p q. p \longrightarrow q \in H \implies \neg p \in H \vee q \in H \rangle$
and *ImpNH*: $\langle \bigwedge p q. \neg (p \longrightarrow q) \in H \implies p \in H \wedge \neg q \in H \rangle$

lemma *Hintikka-model*:
assumes $\langle \text{Hintikka } H \rangle$
shows $\langle (p \in H \longrightarrow \mathcal{M}_T(H) \models_T p) \wedge (\neg p \in H \longrightarrow \neg \mathcal{M}_T(H) \models_T p) \rangle$
 $\langle \text{proof} \rangle$

lemma *MCS-Hintikka*:
assumes $\langle \text{MCS } H \rangle$
shows $\langle \text{Hintikka } H \rangle$
 $\langle \text{proof} \rangle$

lemma *truth-lemma*:
assumes $\langle \text{MCS } H \rangle \langle p \in H \rangle$
shows $\langle \mathcal{M}_T(H) \models_T p \rangle$
 $\langle \text{proof} \rangle$

4.7 Completeness

theorem *strong-completeness*:
assumes $\langle \forall M. (\forall q \in X. M \models_T q) \longrightarrow M \models_T p \rangle$
shows $\langle \exists A. \text{ set } A \subseteq X \wedge \vdash_T \neg p \# A \rangle$
 $\langle \text{proof} \rangle$

abbreviation *valid* :: $\langle 'p \text{ fm} \implies \text{bool} \rangle$ **where**
 $\langle \text{valid } p \equiv \forall M. M \models_T p \rangle$

theorem *completeness*:

assumes $\langle \text{valid } p \rangle$

shows $\langle \vdash_T [\neg p] \rangle$

$\langle \text{proof} \rangle$

theorem *main*: $\langle \text{valid } p \iff \vdash_T [\neg p] \rangle$

$\langle \text{proof} \rangle$

end

Chapter 5

Example: Propositional Sequent Calculus

theory *Example-Propositional-SC* **imports** *Derivations* **begin**

5.1 Syntax

```
datatype 'p fm
  = Fls (<⊥>)
  | Pro 'p (<⋅>)
  | Imp 'p fm <'p fm> (infixr <⟶> 55)
```

abbreviation *Neg* (<¬ → [70] 70) **where** <¬ p ≡ p ⟶ ⊥>

5.2 Semantics

type-synonym 'p model = <'p ⇒ bool>

```
fun semantics :: '<'p model ⇒ 'p fm ⇒ bool> (infix <|=S> 50) where
  <⊥ |=S ⊥ ⟷ False>
  | <I |=S ⋅P ⟷ I P>
  | <I |=S p ⟶ q ⟷ I |=S p ⟶ I |=S q>
```

5.3 Calculus

inductive *Calculus* :: '<'p fm list ⇒ 'p fm list ⇒ bool> (**infix** <⊢_S> 50) **where**

```
  Axiom [simp]: <p # A ⊢S p # B>
  | FlsL [simp]: <⊥ # A ⊢S B>
  | FlsR [elim]: <A ⊢S ⊥ # B ⟹ A ⊢S B>
  | ImpL [intro]: <A ⊢S p # B ⟹ q # A ⊢S B ⟹ (p ⟶ q) # A ⊢S B>
  | ImpR [intro]: <p # A ⊢S q # B ⟹ A ⊢S (p ⟶ q) # B>
  | Cut: <A ⊢S [p] ⟹ p # A ⊢S B ⟹ A ⊢S B>
  | WeakL: <A ⊢S B ⟹ set A ⊆ set A' ⟹ A' ⊢S B>
  | WeakR: <A ⊢S B ⟹ set B ⊆ set B' ⟹ A ⊢S B'>
```

lemma *Boole*: <¬ p # A ⊢_S [] ⟹ A ⊢_S [p]>
<*proof*>

5.4 Soundness

theorem *soundness*: $\langle A \vdash_S B \implies \forall q \in \text{set } A. I \models_S q \implies \exists p \in \text{set } B. I \models_S p \rangle$
 $\langle \text{proof} \rangle$

corollary *soundness'*: $\langle [] \vdash_S [p] \implies I \models_S p \rangle$
 $\langle \text{proof} \rangle$

corollary $\langle \neg [] \vdash_S [] \rangle$
 $\langle \text{proof} \rangle$

5.5 Maximal Consistent Sets

definition *consistent* :: $\langle 'p \text{ fm set} \implies \text{bool} \rangle$ **where**
 $\langle \text{consistent } S \equiv \forall A. \text{set } A \subseteq S \longrightarrow \neg A \vdash_S [\perp] \rangle$

interpretation *MCS-No-Witness-UNIV consistent*
 $\langle \text{proof} \rangle$

interpretation *Derivations-Cut-MCS consistent* $\langle \lambda A p. A \vdash_S [p] \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Bot consistent* $\langle \lambda A p. A \vdash_S [p] \rangle \langle \perp \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Imp consistent* $\langle \lambda A p. A \vdash_S [p] \rangle \langle \lambda p q. p \longrightarrow q \rangle$
 $\langle \text{proof} \rangle$

5.6 Truth Lemma

abbreviation *canonical* :: $\langle 'p \text{ fm set} \implies 'p \text{ model} \rangle (\langle \mathcal{M}_S \rangle)$ **where**
 $\langle \mathcal{M}_S(S) \equiv \lambda P. \cdot P \in S \rangle$

fun *semics* :: $\langle 'p \text{ model} \implies ('p \text{ model} \implies 'p \text{ fm} \implies \text{bool}) \implies 'p \text{ fm} \implies \text{bool} \rangle$
 $(\langle \llbracket - \rrbracket_S \rightarrow [55, 0, 55] 55 \rangle)$ **where**
 $\langle \llbracket - \rrbracket_S \perp \longleftrightarrow \text{False} \rangle$
 $| \langle I \llbracket - \rrbracket_S \cdot P \longleftrightarrow I P \rangle$
 $| \langle I \llbracket \mathcal{R} \rrbracket_S p \longrightarrow q \longleftrightarrow \mathcal{R} I p \longrightarrow \mathcal{R} I q \rangle$

fun *rel* :: $\langle 'p \text{ fm set} \implies 'p \text{ model} \implies 'p \text{ fm} \implies \text{bool} \rangle (\langle \mathcal{R}_S \rangle)$ **where**
 $\langle \mathcal{R}_S(S) - p \longleftrightarrow p \in S \rangle$

theorem *saturated-model*:

assumes $\langle \bigwedge p. \forall M \in \{\mathcal{M}_S(S)\}. M \llbracket \mathcal{R}_S(S) \rrbracket_S p = \mathcal{R}_S(S) M p \rangle \langle M \in \{\mathcal{M}_S(S)\} \rangle$
shows $\langle \mathcal{R}_S(S) M p \longleftrightarrow M \models_S p \rangle$
 $\langle \text{proof} \rangle$

theorem *saturated-MCS*:

assumes $\langle \text{MCS } S \rangle \langle M \in \{\mathcal{M}_S(S)\} \rangle$
shows $\langle M \llbracket \mathcal{R}_S(S) \rrbracket_S p \longleftrightarrow \mathcal{R}_S(S) M p \rangle$
 $\langle \text{proof} \rangle$

interpretation *Truth-No-Witness semics semantics* $\langle \lambda S. \{\mathcal{M}_S(S)\} \rangle$ *rel consistent*
 $\langle \text{proof} \rangle$

5.7 Completeness

theorem *strong-completeness*:

assumes $\langle \forall M. (\forall q \in X. M \models_S q) \longrightarrow M \models_S p \rangle$

shows $\langle \exists A. \text{set } A \subseteq X \wedge A \vdash_S [p] \rangle$

$\langle \text{proof} \rangle$

abbreviation *valid* :: $\langle 'p \text{ fm} \Rightarrow \text{bool} \rangle$ **where**

$\langle \text{valid } p \equiv \forall M. M \models_S p \rangle$

theorem *completeness*:

assumes $\langle \text{valid } p \rangle$

shows $\langle [] \vdash_S [p] \rangle$

$\langle \text{proof} \rangle$

theorem *main*: $\langle \text{valid } p \longleftrightarrow [] \vdash_S [p] \rangle$

$\langle \text{proof} \rangle$

end

Chapter 6

Example: Modal Logic

theory *Example-Modal-Logic* imports *Derivations* begin

6.1 Syntax

```
datatype ('i, 'p) fm
  = Fls (<⊥>)
  | Pro 'p (<⋅>)
  | Imp <('i, 'p) fm> <('i, 'p) fm> (infixr <⟶> 55)
  | Box 'i <('i, 'p) fm> (<□>)
```

abbreviation *Neg* (<¬ → [70] 70) where
 <¬ p ≡ p ⟶ ⊥>

6.2 Semantics

```
datatype ('i, 'p, 'w) model =
  Model (W: <'w set>) (R: <'i ⇒ 'w ⇒ 'w set>) (V: <'w ⇒ 'p ⇒ bool>)
```

type-synonym ('i, 'p, 'w) ctx = <('i, 'p, 'w) model × 'w>

```
fun semantics :: <('i, 'p, 'w) ctx ⇒ ('i, 'p) fm ⇒ bool> (infix <⊨□> 50) where
  <⊥ ⊨□ False>
  | <(M, w) ⊨□ ⋅P ⟷ V M w P>
  | <(M, w) ⊨□ p ⟶ q ⟷ (M, w) ⊨□ p ⟶ (M, w) ⊨□ q>
  | <(M, w) ⊨□ □ i p ⟷ (∀ v ∈ W M ∩ R M i w. (M, v) ⊨□ p)>
```

6.3 Calculus

```
primrec eval :: <('p ⇒ bool) ⇒ (('i, 'p) fm ⇒ bool) ⇒ ('i, 'p) fm ⇒ bool> where
  <eval - - ⊥ = False>
  | <eval g - (⋅P) = g P>
  | <eval g h (p ⟶ q) = (eval g h p ⟶ eval g h q)>
  | <eval - h (□ i p) = h (□ i p)>
```

abbreviation <tautology p ≡ ∀ g h. eval g h p>

```
inductive Calculus :: <('i, 'p) fm ⇒ bool> (<⊢□ → [50] 50) where
  A1: <tautology p ⟹ ⊢□ p>
  | A2: <⊢□ □ i (p ⟶ q) ⟶ □ i p ⟶ □ i q>
```

| *R1*: $\langle \vdash_{\square} p \implies \vdash_{\square} p \longrightarrow q \implies \vdash_{\square} q \rangle$
 | *R2*: $\langle \vdash_{\square} p \implies \vdash_{\square} \square i p \rangle$

primrec *imply* :: $\langle ('i, 'p) \text{ fm list} \Rightarrow ('i, 'p) \text{ fm} \Rightarrow ('i, 'p) \text{ fm} \rangle$ (**infixr** $\langle \rightsquigarrow \rangle$ 56) **where**
 $\langle (\square \rightsquigarrow p) = p \rangle$
 | $\langle (q \# A \rightsquigarrow p) = (q \longrightarrow A \rightsquigarrow p) \rangle$

abbreviation *Calculus-assms* (**infix** $\langle \vdash_{\square} \rangle$ 50) **where**
 $\langle A \vdash_{\square} p \equiv \vdash_{\square} A \rightsquigarrow p \rangle$

6.4 Soundness

lemma *eval-antics*: $\langle \text{eval } (g w) (\lambda q. (\text{Model } Ws r g, w) \models_{\square} q) p = ((\text{Model } Ws r g, w) \models_{\square} p) \rangle$
 $\langle \text{proof} \rangle$

lemma *tautology*:
assumes $\langle \text{tautology } p \rangle$
shows $\langle (M, w) \models_{\square} p \rangle$
 $\langle \text{proof} \rangle$

theorem *soundness*: $\langle \vdash_{\square} p \implies w \in W M \implies (M, w) \models_{\square} p \rangle$
 $\langle \text{proof} \rangle$

6.5 Admissible rules

lemma *K-imply-head*: $\langle p \# A \vdash_{\square} p \rangle$
 $\langle \text{proof} \rangle$

lemma *K-imply-Cons*:
assumes $\langle A \vdash_{\square} q \rangle$
shows $\langle p \# A \vdash_{\square} q \rangle$
 $\langle \text{proof} \rangle$

lemma *K-right-mp*:
assumes $\langle A \vdash_{\square} p \rangle \langle A \vdash_{\square} p \longrightarrow q \rangle$
shows $\langle A \vdash_{\square} q \rangle$
 $\langle \text{proof} \rangle$

lemma *deduct1*: $\langle A \vdash_{\square} p \longrightarrow q \implies p \# A \vdash_{\square} q \rangle$
 $\langle \text{proof} \rangle$

lemma *imply-append [iff]*: $\langle (A @ B \rightsquigarrow r) = (A \rightsquigarrow B \rightsquigarrow r) \rangle$
 $\langle \text{proof} \rangle$

lemma *imply-swap-append*: $\langle A @ B \vdash_{\square} r \implies B @ A \vdash_{\square} r \rangle$
 $\langle \text{proof} \rangle$

lemma *K-ImpI*: $\langle p \# A \vdash_{\square} q \implies A \vdash_{\square} p \longrightarrow q \rangle$
 $\langle \text{proof} \rangle$

lemma *imply-mem [simp]*: $\langle p \in \text{set } A \implies A \vdash_{\square} p \rangle$
 $\langle \text{proof} \rangle$

lemma *add-imply [simp]*: $\langle \vdash_{\square} q \implies A \vdash_{\square} q \rangle$
 $\langle \text{proof} \rangle$

lemma *K-imply-weaken*: $\langle A \vdash_{\square} q \implies \text{set } A \subseteq \text{set } A' \implies A' \vdash_{\square} q \rangle$
 $\langle \text{proof} \rangle$

lemma *K-Boole*:
assumes $\langle (\neg p) \# A \vdash_{\square} \perp \rangle$
shows $\langle A \vdash_{\square} p \rangle$
 $\langle \text{proof} \rangle$

lemma *K-distrib-K-imp*:
assumes $\langle \vdash_{\square} \square i (A \rightsquigarrow q) \rangle$
shows $\langle \text{map } (\square i) A \vdash_{\square} \square i q \rangle$
 $\langle \text{proof} \rangle$

6.6 Maximal Consistent Sets

definition *consistent* :: $\langle ('i, 'p) \text{ fm set} \Rightarrow \text{bool} \rangle$ **where**
 $\langle \text{consistent } S \equiv \forall A. \text{ set } A \subseteq S \longrightarrow \neg A \vdash_{\square} \perp \rangle$

interpretation *MCS-No-Witness-UNIV consistent*
 $\langle \text{proof} \rangle$

interpretation *Derivations-Cut-MCS consistent Calculus-assms*
 $\langle \text{proof} \rangle$

interpretation *Derivations-Bot consistent Calculus-assms* $\langle \perp \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Imp consistent Calculus-assms* $\langle \lambda p q. p \longrightarrow q \rangle$
 $\langle \text{proof} \rangle$

theorem *deriv-in-maximal*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle \vdash_{\square} p \rangle$
shows $\langle p \in S \rangle$
 $\langle \text{proof} \rangle$

6.7 Truth Lemma

abbreviation *known* :: $\langle ('i, 'p) \text{ fm set} \Rightarrow 'i \Rightarrow ('i, 'p) \text{ fm set} \rangle$ **where**
 $\langle \text{known } S i \equiv \{p. \square i p \in S\} \rangle$

abbreviation *reach* :: $\langle 'i \Rightarrow ('i, 'p) \text{ fm set} \Rightarrow ('i, 'p) \text{ fm set set} \rangle$ **where**
 $\langle \text{reach } i S \equiv \{S'. \text{ known } S i \subseteq S' \wedge \text{MCS } S'\} \rangle$

abbreviation *canonical* :: $\langle ('i, 'p) \text{ fm set} \Rightarrow ('i, 'p, ('i, 'p) \text{ fm set}) \text{ ctx} \rangle$ $\langle \mathcal{M}_{\square} \rangle$ **where**
 $\langle \mathcal{M}_{\square}(S) \equiv (\text{Model } \{S. \text{MCS } S\} \text{ reach } (\lambda S P. \cdot P \in S), S) \rangle$

fun *semics* ::
 $\langle ('i, 'p, 'w) \text{ ctx} \Rightarrow (('i, 'p, 'w) \text{ ctx} \Rightarrow ('i, 'p) \text{ fm} \Rightarrow \text{bool}) \Rightarrow ('i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$
 $\langle \langle \text{[-]}_{\square} \text{-} \rangle [55, 0, 55] 55 \rangle$ **where**
 $\langle \text{[-]}_{\square} \perp \longleftrightarrow \text{False} \rangle$
 $| \langle (M, w) \llbracket \text{-} \rrbracket_{\square} \cdot P \longleftrightarrow V M w P \rangle$
 $| \langle (M, w) \llbracket \mathcal{R} \rrbracket_{\square} p \longrightarrow q \longleftrightarrow \mathcal{R} (M, w) p \longrightarrow \mathcal{R} (M, w) q \rangle$
 $| \langle (M, w) \llbracket \mathcal{R} \rrbracket_{\square} \square i p \longleftrightarrow (\forall v \in W M \cap R M i w. \mathcal{R} (M, v) p) \rangle$

fun $rel :: \langle ('i, 'p) \text{ fm set} \Rightarrow ('i, 'p, ('i, 'p) \text{ fm set}) \text{ ctx} \Rightarrow ('i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle (\langle \mathcal{R}_\square \rangle)$ **where**
 $\langle \mathcal{R}_\square(-) (-, w) p \longleftrightarrow p \in w \rangle$

theorem *saturated-model:*

fixes $S :: \langle ('i, 'p) \text{ fm set} \rangle$

assumes $\langle \bigwedge (S :: ('i, 'p) \text{ fm set}) p. \text{MCS } S \Longrightarrow \mathcal{M}_\square(S) \llbracket \mathcal{R}_\square(S') \rrbracket_\square p \longleftrightarrow p \in S \rangle$

shows $\langle \text{MCS } S \Longrightarrow \mathcal{M}_\square(S) \models_\square p \longleftrightarrow p \in S \rangle$

$\langle \text{proof} \rangle$

theorem *saturated-MCS:*

assumes $\langle \text{MCS } S \rangle$

shows $\langle \mathcal{M}_\square(S) \llbracket \mathcal{R}_\square(S') \rrbracket_\square p \longleftrightarrow \mathcal{R}_\square(S') (\mathcal{M}_\square(S)) p \rangle$

$\langle \text{proof} \rangle$

interpretation *Truth-No-Witness semics semantics* $\langle \lambda-. \{ \mathcal{M}_\square(S) \mid S. \text{MCS } S \} \rangle$ *rel consistent*

$\langle \text{proof} \rangle$

lemma *Truth-lemma:*

assumes $\langle \text{MCS } S \rangle$

shows $\langle \mathcal{M}_\square(S) \models_\square p \longleftrightarrow p \in S \rangle$

$\langle \text{proof} \rangle$

6.8 Completeness

theorem *strong-completeness:*

assumes $\langle \forall M :: ('i, 'p, ('i, 'p) \text{ fm set}) \text{ model}. \forall w \in W M.$

$(\forall q \in X. (M, w) \models_\square q) \longrightarrow (M, w) \models_\square p \rangle$

shows $\langle \exists A. \text{set } A \subseteq X \wedge A \vdash_\square p \rangle$

$\langle \text{proof} \rangle$

abbreviation *valid* $:: \langle ('i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$ **where**

$\langle \text{valid } p \equiv \forall (M :: ('i, 'p, ('i, 'p) \text{ fm set}) \text{ model}). \forall w \in W M. (M, w) \models_\square p \rangle$

corollary *completeness:* $\langle \text{valid } p \Longrightarrow \vdash_\square p \rangle$

$\langle \text{proof} \rangle$

theorem *main:* $\langle \text{valid } p \longleftrightarrow \vdash_\square p \rangle$

$\langle \text{proof} \rangle$

end

Chapter 7

Example: Hybrid Logic

theory *Example-Hybrid-Logic* imports *Derivations* begin

7.1 Syntax

datatype (*nominals-fm*: 'i, 'p) *fm*
= *Fls* ($\langle \perp \rangle$)
| *Pro* 'p ($\langle \cdot \rangle$)
| *Nom* 'i ($\langle \cdot \rangle$)
| *Imp* $\langle ('i, 'p) \text{ fm} \rangle \langle ('i, 'p) \text{ fm} \rangle$ (**infix** $\langle \longrightarrow \rangle$ 55)
| *Dia* $\langle ('i, 'p) \text{ fm} \rangle \langle \langle \diamond \rangle \rangle$
| *Sat* 'i $\langle ('i, 'p) \text{ fm} \rangle \langle \langle @ \rangle \rangle$
| *All* $\langle ('i, 'p) \text{ fm} \rangle \langle \langle \mathbf{A} \rangle \rangle$

abbreviation *Neg* ($\langle \neg \rightarrow \rangle$ [70] 70) **where** $\langle \neg p \equiv p \longrightarrow \perp \rangle$

abbreviation *Con* (**infix** $\langle \wedge \rangle$ 35) **where** $\langle p \wedge q \equiv \neg (p \longrightarrow \neg q) \rangle$

type-synonym ('i, 'p) *lbd* = $\langle 'i \times ('i, 'p) \text{ fm} \rangle$

primrec *nominals-lbd* :: $\langle ('i, 'p) \text{ lbd set} \Rightarrow 'i \text{ set} \rangle$ **where**
 $\langle \text{nominals-lbd } (i, p) = \{i\} \cup \text{nominals-fm } p \rangle$

abbreviation *nominals* :: $\langle ('i, 'p) \text{ lbd set} \Rightarrow 'i \text{ set} \rangle$ **where**
 $\langle \text{nominals } S \equiv \bigcup ip \in S. \text{nominals-lbd } ip \rangle$

lemma *finite-nominals-fm* [*simp*]: $\langle \text{finite } (\text{nominals-fm } p) \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-nominals-lbd*: $\langle \text{finite } (\text{nominals-lbd } p) \rangle$
 $\langle \text{proof} \rangle$

7.2 Semantics

datatype ('w, 'p) *model* =
Model (*W*: $\langle 'w \text{ set} \rangle$) (*R*: $\langle 'w \Rightarrow 'w \text{ set} \rangle$) (*V*: $\langle 'w \Rightarrow 'p \Rightarrow \text{bool} \rangle$)

type-synonym ('i, 'p, 'w) *ctx* = $\langle ('w, 'p) \text{ model} \times ('i \Rightarrow 'w) \times 'w \rangle$

fun *semantics* :: $\langle ('i, 'p, 'w) \text{ ctx} \Rightarrow ('i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$ (**infix** $\langle \models_{@} \rangle$ 50) **where**
 $\langle (M, g, w) \models_{@} \perp \longleftrightarrow \text{False} \rangle$

$\langle (M, -, w) \models_{\mathbb{Q}} \cdot P \longleftrightarrow V M w P \rangle$
 $\langle (-, g, w) \models_{\mathbb{Q}} \cdot i \longleftrightarrow w = g i \rangle$
 $\langle (M, g, w) \models_{\mathbb{Q}} p \longrightarrow q \longleftrightarrow (M, g, w) \models_{\mathbb{Q}} p \longrightarrow (M, g, w) \models_{\mathbb{Q}} q \rangle$
 $\langle (M, g, w) \models_{\mathbb{Q}} \diamond p \longleftrightarrow (\exists v \in W M \cap R M w. (M, g, v) \models_{\mathbb{Q}} p) \rangle$
 $\langle (M, g, -) \models_{\mathbb{Q}} @i p \longleftrightarrow (M, g, g i) \models_{\mathbb{Q}} p \rangle$
 $\langle (M, g, -) \models_{\mathbb{Q}} \mathbf{A} p \longleftrightarrow (\forall v \in W M. (M, g, v) \models_{\mathbb{Q}} p) \rangle$

lemma semantics-fresh: $\langle i \notin \text{nominals-fm } p \implies (M, g, w) \models_{\mathbb{Q}} p \longleftrightarrow (M, g(i := v), w) \models_{\mathbb{Q}} p \rangle$
 $\langle \text{proof} \rangle$

lemma semantics-fresh-lbd:

$\langle k \notin \text{nominals-lbd } (i, p) \implies (M, g, w) \models_{\mathbb{Q}} p \longleftrightarrow (M, g(k := v), w) \models_{\mathbb{Q}} p \rangle$
 $\langle \text{proof} \rangle$

7.3 Calculus

inductive Calculus :: $\langle ('i, 'p) \text{ lbd list} \implies ('i, 'p) \text{ lbd} \implies \text{bool} \rangle$ (**infix** $\langle \vdash_{\mathbb{Q}} \rangle$ 50) **where**

Assm [simp]: $\langle (i, p) \in \text{set } A \implies A \vdash_{\mathbb{Q}} (i, p) \rangle$
Ref [simp]: $\langle A \vdash_{\mathbb{Q}} (i, \cdot i) \rangle$
Nom [dest]: $\langle A \vdash_{\mathbb{Q}} (i, \cdot k) \implies A \vdash_{\mathbb{Q}} (i, p) \implies A \vdash_{\mathbb{Q}} (k, p) \rangle$
FlsE [elim]: $\langle A \vdash_{\mathbb{Q}} (i, \perp) \implies A \vdash_{\mathbb{Q}} (k, p) \rangle$
ImpI [intro]: $\langle (i, p) \# A \vdash_{\mathbb{Q}} (i, q) \implies A \vdash_{\mathbb{Q}} (i, p \longrightarrow q) \rangle$
ImpE [dest]: $\langle A \vdash_{\mathbb{Q}} (i, p \longrightarrow q) \implies A \vdash_{\mathbb{Q}} (i, p) \implies A \vdash_{\mathbb{Q}} (i, q) \rangle$
SatI [intro]: $\langle A \vdash_{\mathbb{Q}} (i, p) \implies A \vdash_{\mathbb{Q}} (k, @i p) \rangle$
SatE [dest]: $\langle A \vdash_{\mathbb{Q}} (i, @k p) \implies A \vdash_{\mathbb{Q}} (k, p) \rangle$
DiaI [intro]: $\langle A \vdash_{\mathbb{Q}} (i, \diamond (\cdot k)) \implies A \vdash_{\mathbb{Q}} (k, p) \implies A \vdash_{\mathbb{Q}} (i, \diamond p) \rangle$
DiaE [elim]: $\langle A \vdash_{\mathbb{Q}} (i, \diamond p) \implies k \notin \text{nominals } (\{(i, p), (j, q)\} \cup \text{set } A) \implies (k, p) \# (i, \diamond (\cdot k)) \# A \vdash_{\mathbb{Q}} (j, q) \implies A \vdash_{\mathbb{Q}} (j, q) \rangle$
AllI [intro]: $\langle A \vdash_{\mathbb{Q}} (k, p) \implies k \notin \text{nominals } (\{(i, p)\} \cup \text{set } A) \implies A \vdash_{\mathbb{Q}} (i, \mathbf{A} p) \rangle$
Alle [dest]: $\langle A \vdash_{\mathbb{Q}} (i, \mathbf{A} p) \implies A \vdash_{\mathbb{Q}} (k, p) \rangle$
Clas: $\langle (i, p \longrightarrow q) \# A \vdash_{\mathbb{Q}} (i, p) \implies A \vdash_{\mathbb{Q}} (i, p) \rangle$
Cut: $\langle A \vdash_{\mathbb{Q}} (k, q) \implies (k, q) \# B \vdash_{\mathbb{Q}} (i, p) \implies A @ B \vdash_{\mathbb{Q}} (i, p) \rangle$

7.4 Soundness

theorem soundness: $\langle A \vdash_{\mathbb{Q}} (i, p) \implies \text{list-all } (\lambda(i, p). (M, g, g i) \models_{\mathbb{Q}} p) A \implies \text{range } g \subseteq W M \implies (M, g, g i) \models_{\mathbb{Q}} p \rangle$
 $\langle \text{proof} \rangle$

corollary soundness':

assumes $\langle [] \vdash_{\mathbb{Q}} (i, p) \rangle \langle i \notin \text{nominals-fm } p \rangle$
and $\langle \text{range } g \subseteq W M \rangle \langle w \in W M \rangle$
shows $\langle (M, g, w) \models_{\mathbb{Q}} p \rangle$
 $\langle \text{proof} \rangle$

corollary $\langle \neg ([] \vdash_{\mathbb{Q}} (i, \perp)) \rangle$
 $\langle \text{proof} \rangle$

7.5 Admissible Rules

lemma Assm-head [simp]: $\langle (p, i) \# A \vdash_{\mathbb{Q}} (p, i) \rangle$
 $\langle \text{proof} \rangle$

lemma SatE':

assumes $\langle (k, q) \# A \vdash_{\text{@}} (i, p) \rangle$
shows $\langle (j, @k q) \# A \vdash_{\text{@}} (i, p) \rangle$
 $\langle \text{proof} \rangle$

lemma *ImpI'*:

assumes $\langle (k, q) \# A \vdash_{\text{@}} (i, p) \rangle$
shows $\langle A \vdash_{\text{@}} (i, (@k q) \longrightarrow p) \rangle$
 $\langle \text{proof} \rangle$

lemma *Weak'*: $\langle A \vdash_{\text{@}} (i, p) \implies A @ B \vdash_{\text{@}} (i, p) \rangle$
 $\langle \text{proof} \rangle$

lemma *Weaken*: $\langle A \vdash_{\text{@}} (i, p) \implies \text{set } A \subseteq \text{set } B \implies B \vdash_{\text{@}} (i, p) \rangle$
 $\langle \text{proof} \rangle$

lemma *Weak*: $\langle A \vdash_{\text{@}} (i, p) \implies (k, q) \# A \vdash_{\text{@}} (i, p) \rangle$
 $\langle \text{proof} \rangle$

lemma *deduct1*: $\langle A \vdash_{\text{@}} (i, p \longrightarrow q) \implies (i, p) \# A \vdash_{\text{@}} (i, q) \rangle$
 $\langle \text{proof} \rangle$

lemma *Boole*: $\langle (i, \neg p) \# A \vdash_{\text{@}} (i, \perp) \implies A \vdash_{\text{@}} (i, p) \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations Calculus*

$\langle \text{proof} \rangle$

7.6 Maximal Consistent Sets

definition *consistent* :: $\langle ('i, 'p) \text{ lbd set} \Rightarrow \text{bool} \rangle$ **where**
 $\langle \text{consistent } S \equiv \forall A \text{ a. set } A \subseteq S \longrightarrow \neg A \vdash_{\text{@}} (a, \perp) \rangle$

lemma *consistent-add-diamond-witness*:

assumes $\langle \text{consistent } S \rangle \langle (i, \diamond p) \in S \rangle \langle k \notin \text{nominals } S \rangle$
shows $\langle \text{consistent } (\{(k, p), (i, \diamond (\cdot k))\} \cup S) \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-add-global-witness*:

assumes $\langle \text{consistent } S \rangle \langle (i, \neg \mathbf{A} p) \in S \rangle \langle k \notin \text{nominals } S \rangle$
shows $\langle \text{consistent } (\{(k, \neg p)\} \cup S) \rangle$
 $\langle \text{proof} \rangle$

fun *witness* :: $\langle ('i, 'p) \text{ lbd} \Rightarrow ('i, 'p) \text{ lbd set} \Rightarrow ('i, 'p) \text{ lbd set} \rangle$ **where**

$\langle \text{witness } (i, \diamond p) S = (\text{let } k = \text{SOME } k. k \notin \text{nominals } (\{(i, p)\} \cup S) \text{ in } \{(k, p), (i, \diamond (\cdot k))\}) \rangle$
 $| \langle \text{witness } (i, \neg \mathbf{A} p) S = (\text{let } k = \text{SOME } k. k \notin \text{nominals } (\{(i, p)\} \cup S) \text{ in } \{(k, \neg p)\}) \rangle$
 $| \langle \text{witness } (-, -) - = \{\} \rangle$

lemma *consistent-witness'*:

assumes $\langle \text{consistent } (\{(i, p)\} \cup S) \rangle \langle \text{infinite } (\text{UNIV} - \text{nominals } S) \rangle$
shows $\langle \text{consistent } (\text{witness } (i, p) S \cup \{(i, p)\} \cup S) \rangle$
 $\langle \text{proof} \rangle$

interpretation *MCS-Witness-UNIV consistent witness nominals-lbd*

$\langle \text{proof} \rangle$

lemma *witnessed-diamond*: $\langle \text{witnessed } S \implies (i, \diamond p) \in S \implies \exists k. (i, \diamond (\cdot k)) \in S \wedge (k, p) \in S \rangle$
 $\langle \text{proof} \rangle$

lemma *witnessed-global*: $\langle \text{witnessed } S \implies (i, \neg \mathbf{A} p) \in S \implies \exists k. (k, \neg p) \in S \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Cut-MCS consistent Calculus*
 $\langle \text{proof} \rangle$

interpretation *Derivations-Bot consistent Calculus* $\langle (i, \perp) \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Not consistent Calculus* $\langle (i, \perp) \rangle \langle \lambda(i, p). (i, \neg p) \rangle$
 $\langle \text{proof} \rangle$

lemma *MCS-impE'*: $\langle \text{consistent } S \implies \text{maximal } S \implies (i, p \longrightarrow q) \in S \implies (i, p) \in S \longrightarrow (i, q) \in S \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Uni consistent witness nominals-lbd Calculus* $\langle (i, \perp) \rangle \langle \lambda(i, p). (i, \neg p) \rangle$
 $\langle \lambda(i, p). (i, \mathbf{A} p) \rangle \langle \lambda k (i, p). (k, p) \rangle$
 $\langle \text{proof} \rangle$

lemma *conE1* [*elim*]: $\langle A \vdash_{\@} (i, p \wedge q) \implies A \vdash_{\@} (i, p) \rangle$
 $\langle \text{proof} \rangle$

lemma *conE2* [*elim*]: $\langle A \vdash_{\@} (i, p \wedge q) \implies A \vdash_{\@} (i, q) \rangle$
 $\langle \text{proof} \rangle$

lemma *conI* [*intro*]: $\langle A \vdash_{\@} (i, p) \implies A \vdash_{\@} (i, q) \implies A \vdash_{\@} (i, p \wedge q) \rangle$
 $\langle \text{proof} \rangle$

lemma *MCS-con*:
assumes $\langle \text{MCS } S \rangle$
shows $\langle (i, p \wedge q) \in S \longleftrightarrow (i, p) \in S \wedge (i, q) \in S \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Exi consistent witness nominals-lbd Calculus*
 $\langle \lambda(i, p). (i, \diamond p) \rangle \langle \lambda k (i, p). (i, @k p \wedge \diamond (\cdot k)) \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-uni'*:
assumes $\langle \text{MCS } S \rangle \langle \text{witnessed } S \rangle$
shows $\langle (i, \mathbf{A} p) \in S \longleftrightarrow (\forall k. (k, p) \in S) \rangle$
 $\langle \text{proof} \rangle$

corollary *MCS-exi'*:
assumes $\langle \text{MCS } S \rangle \langle \text{witnessed } S \rangle$
shows $\langle (i, \diamond p) \in S \longleftrightarrow (\exists k. (i, @k p \wedge \diamond (\cdot k)) \in S) \rangle$
 $\langle \text{proof} \rangle$

7.7 Nominals

lemma *MCS-Nom-refl*:
assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle (i, \cdot i) \in S \rangle$

$\langle \text{proof} \rangle$

lemma *MCS-Nom-sym*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle (i, \cdot k) \in S \rangle$

shows $\langle (k, \cdot i) \in S \rangle$

$\langle \text{proof} \rangle$

lemma *MCS-Nom-trans*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle (i, \cdot j) \in S \rangle \langle (j, \cdot k) \in S \rangle$

shows $\langle (i, \cdot k) \in S \rangle$

$\langle \text{proof} \rangle$

7.8 Truth Lemma

fun *semics* :: $\langle (i, 'p, 'w) \text{ ctx} \Rightarrow ((i, 'p, 'w) \text{ ctx} \Rightarrow (i, 'p) \text{ fm} \Rightarrow \text{bool}) \Rightarrow (i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$

$\langle \langle (- \llbracket - \rrbracket_{\text{at}} -) \rangle [55, 0, 55] 55 \rangle$ **where**

$\langle - \llbracket - \rrbracket_{\text{at}} \perp \longleftrightarrow \text{False} \rangle$

| $\langle (M, -, w) \llbracket - \rrbracket_{\text{at}} \cdot P \longleftrightarrow V M w P \rangle$

| $\langle (-, g, w) \llbracket - \rrbracket_{\text{at}} \cdot i \longleftrightarrow w = g i \rangle$

| $\langle (M, g, w) \llbracket \mathcal{R} \rrbracket_{\text{at}}(p) \longrightarrow q \longleftrightarrow \mathcal{R} (M, g, w) p \longrightarrow \mathcal{R} (M, g, w) q \rangle$

| $\langle (M, g, w) \llbracket \mathcal{R} \rrbracket_{\text{at}} \diamond p \longleftrightarrow (\exists v \in W M \cap R M w. \mathcal{R} (M, g, v) p) \rangle$

| $\langle (M, g, -) \llbracket \mathcal{R} \rrbracket_{\text{at}} @i p \longleftrightarrow \mathcal{R} (M, g, g i) p \rangle$

| $\langle (M, g, -) \llbracket \mathcal{R} \rrbracket_{\text{at}} \mathbf{A} p \longleftrightarrow (\forall v \in W M. \mathcal{R} (M, g, v) p) \rangle$

fun *rel* :: $\langle (i, 'p) \text{ lbd set} \Rightarrow (i, 'p, 'i) \text{ ctx} \Rightarrow (i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle \langle \mathcal{R}_{\text{at}} \rangle$ **where**

$\langle \mathcal{R}_{\text{at}}(S) (-, -, i) p \longleftrightarrow (i, p) \in S \rangle$

definition *equiv-nom* :: $\langle (i, 'p) \text{ lbd set} \Rightarrow 'i \Rightarrow 'i \Rightarrow \text{bool} \rangle$ **where**

$\langle \text{equiv-nom } S i k \equiv (i, \cdot k) \in S \rangle$

lemma *equiv-nom-reflp*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$

shows $\langle \text{reflp (equiv-nom } S) \rangle$

$\langle \text{proof} \rangle$

lemma *equiv-nom-symp*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$

shows $\langle \text{symp (equiv-nom } S) \rangle$

$\langle \text{proof} \rangle$

lemma *equiv-nom-transp*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$

shows $\langle \text{transp (equiv-nom } S) \rangle$

$\langle \text{proof} \rangle$

lemma *equiv-nom-equivp*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$

shows $\langle \text{equivp (equiv-nom } S) \rangle$

$\langle \text{proof} \rangle$

definition *assign* :: $\langle 'i \Rightarrow (i, 'p) \text{ lbd set} \Rightarrow 'i \rangle \langle \llbracket - \rrbracket \rangle [0, 100] 100$ **where**

$\langle [i]_S \equiv \text{minim} (|UNIV|) \{k. \text{equiv-nom } S i k \} \rangle$

lemma *equiv-nom-ne*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$

shows $\langle \{k. \text{equiv-nom } S \ i \ k\} \neq \{\}\rangle$
 $\langle \text{proof} \rangle$

lemma *equiv-nom-assign*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle$
shows $\langle \text{equiv-nom } S \ i \ ([i]_S) \rangle$
 $\langle \text{proof} \rangle$

lemma *equiv-nom-Nom*:

assumes $\langle \text{consistent } S \rangle \langle \text{maximal } S \rangle \langle \text{equiv-nom } S \ i \ k \rangle \langle (i, p) \in S \rangle$
shows $\langle (k, p) \in S \rangle$
 $\langle \text{proof} \rangle$

definition *reach* :: $\langle ('i, 'p) \text{ lbd set} \Rightarrow 'i \Rightarrow 'i \text{ set} \rangle$ **where**

$\langle \text{reach } S \ i \equiv \{[k]_S \mid k. (i, \diamond (\cdot k)) \in S\} \rangle$

primrec *canonical* :: $\langle ('i, 'p) \text{ lbd set} \times 'i \Rightarrow ('i, 'p, 'i) \text{ ctx} \rangle$ $\langle \mathcal{M}_{@} \rangle$ **where**

$\langle \mathcal{M}_{@}(S, i) = (\text{Model } \{[k]_S \mid k. \text{True}\} (\text{reach } S)) (\lambda i P. (i, \cdot P) \in S), \lambda i. [i]_S, [i]_S) \rangle$

theorem *saturated-model*:

assumes $\langle \bigwedge p \ i. \mathcal{M}_{@}(S, i) \llbracket \mathcal{R}_{@}(S) \rrbracket_{@}(p) \longleftrightarrow \mathcal{R}_{@}(S) (\mathcal{M}_{@}(S, i)) \ p \rangle \langle M \in \{\mathcal{M}_{@}(S, i) \mid i. \text{True}\} \rangle$
shows $\langle \mathcal{R}_{@}(S) (\mathcal{M}_{@}(S, i)) \ p \longleftrightarrow \mathcal{M}_{@}(S, i) \models_{@} p \rangle$
 $\langle \text{proof} \rangle$

lemma *reach-assign*: $\langle \text{reach } S \ ([i]_S) \subseteq \{[k]_S \mid k. \text{True}\} \rangle$

$\langle \text{proof} \rangle$

theorem *saturated-MCS*:

assumes $\langle \text{MCS } S \rangle$
shows $\langle \mathcal{M}_{@}(S, i) \llbracket \mathcal{R}_{@}(S) \rrbracket_{@}(p) \longleftrightarrow \mathcal{R}_{@}(S) (\mathcal{M}_{@}(S, i)) \ p \rangle$
 $\langle \text{proof} \rangle$

interpretation *Truth-Witness semics semantics* $\langle \lambda S. \{\mathcal{M}_{@}(S, i) \mid i. \text{True}\} \rangle$ *rel consistent witness nominals-lbd*

$\langle \text{proof} \rangle$

lemma *Truth-lemma*:

assumes $\langle \text{MCS } S \rangle$
shows $\langle \mathcal{M}_{@}(S, i) \models_{@} p \longleftrightarrow (i, p) \in S \rangle$
 $\langle \text{proof} \rangle$

7.9 Cardinalities

datatype *marker* = *FlsM* | *ImpM* | *DiaM* | *SatM* | *AllM*

type-synonym $\langle ('i, 'p) \text{ enc} = \langle ('i + 'p) + \text{marker} \times \text{nat} \rangle$

abbreviation $\langle \text{NOM } i \equiv \text{Inl } (\text{Inl } i) \rangle$

abbreviation $\langle \text{PRO } x \equiv \text{Inl } (\text{Inr } x) \rangle$

abbreviation $\langle \text{FLS} \equiv \text{Inr } (\text{FlsM}, 0) \rangle$

abbreviation $\langle \text{IMP } n \equiv \text{Inr } (\text{FlsM}, n) \rangle$

abbreviation $\langle \text{DIA} \equiv \text{Inr } (\text{DiaM}, 0) \rangle$

abbreviation $\langle \text{SAT} \equiv \text{Inr } (\text{SatM}, 0) \rangle$

abbreviation $\langle \text{GLO} \equiv \text{Inr } (\text{AllM}, 0) \rangle$

primrec *encode* :: $\langle ('i, 'p) \text{ fm} \Rightarrow ('i, 'p) \text{ enc list} \rangle$ **where**
 $\langle \text{encode } \perp = [\text{FLS}] \rangle$
 $\langle \text{encode } (\cdot P) = [\text{PRO } P] \rangle$
 $\langle \text{encode } (\cdot i) = [\text{NOM } i] \rangle$
 $\langle \text{encode } (p \longrightarrow q) = \text{IMP } (\text{length } (\text{encode } p)) \# \text{encode } p @ \text{encode } q \rangle$
 $\langle \text{encode } (\diamond p) = \text{DIA} \# \text{encode } p \rangle$
 $\langle \text{encode } (@ i p) = \text{SAT} \# \text{NOM } i \# \text{encode } p \rangle$
 $\langle \text{encode } (\mathbf{A} p) = \text{GLO} \# \text{encode } p \rangle$

lemma *encode-ne* [*simp*]: $\langle \text{encode } p \neq [] \rangle$
 $\langle \text{proof} \rangle$

lemma *inj-encode'*: $\langle \text{encode } p = \text{encode } q \Longrightarrow p = q \rangle$
 $\langle \text{proof} \rangle$

primrec *encode-lbd* :: $\langle ('i, 'p) \text{ lbd} \Rightarrow ('i, 'p) \text{ enc list} \rangle$ **where**
 $\langle \text{encode-lbd } (i, p) = \text{NOM } i \# \text{encode } p \rangle$

lemma *inj-encode-lbd'*: $\langle \text{encode-lbd } (i, p) = \text{encode-lbd } (k, q) \Longrightarrow i = k \wedge p = q \rangle$
 $\langle \text{proof} \rangle$

lemma *inj-encode-lbd*: $\langle \text{inj encode-lbd} \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-marker*: $\langle \text{finite } (\text{UNIV} :: \text{marker set}) \rangle$
 $\langle \text{proof} \rangle$

lemma *card-of-lbd*:
assumes $\langle \text{infinite } (\text{UNIV} :: 'i \text{ set}) \rangle$
shows $\langle |\text{UNIV} :: ('i, 'p) \text{ lbd set}| \leq o |\text{UNIV} :: 'i \text{ set}| + c |\text{UNIV} :: 'p \text{ set}| \rangle$
 $\langle \text{proof} \rangle$

7.10 Completeness

theorem *strong-completeness*:

fixes $p :: \langle ('i, 'p) \text{ fm} \rangle$
assumes $\langle \forall M :: ('i, 'p) \text{ model. } \forall g. \forall w \in W M. \text{range } g \subseteq W M \longrightarrow$
 $(\forall (k, q) \in X. (M, g, g k) \models_{@} q) \longrightarrow (M, g, w) \models_{@} p \rangle$
 $\langle \text{infinite } (\text{UNIV} :: 'i \text{ set}) \rangle$
 $\langle |\text{UNIV} :: 'i \text{ set}| + c |\text{UNIV} :: 'p \text{ set}| \leq o |\text{UNIV} - \text{nominals } X| \rangle$
shows $\langle \exists A. \text{set } A \subseteq X \wedge A \vdash_{@} (i, p) \rangle$
 $\langle \text{proof} \rangle$

abbreviation *valid* :: $\langle ('i, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$ **where**

$\langle \text{valid } p \equiv \forall (M :: ('i, 'p) \text{ model}) g. \forall w \in W M. \text{range } g \subseteq W M \longrightarrow (M, g, w) \models_{@} p \rangle$

theorem *completeness*:

fixes $p :: \langle ('i, 'p) \text{ fm} \rangle$
assumes $\langle \text{valid } p \rangle \langle \text{infinite } (\text{UNIV} :: 'i \text{ set}) \rangle \langle |\text{UNIV} :: 'p \text{ set}| \leq o |\text{UNIV} :: 'i \text{ set}| \rangle$
shows $\langle [] \vdash_{@} (i, p) \rangle$
 $\langle \text{proof} \rangle$

corollary *completeness'*:

fixes $p :: \langle ('i, 'i) \text{ fm} \rangle$
assumes $\langle \text{valid } p \rangle \langle \text{infinite } (\text{UNIV} :: 'i \text{ set}) \rangle$

shows $\langle \Box \vdash_{@} (i, p) \rangle$
 $\langle \text{proof} \rangle$

theorem *main*:

fixes $p :: \langle ('i, 'p) \text{ fm} \rangle$

assumes $\langle i \notin \text{nominals-fm } p \rangle \langle \text{infinite } (UNIV :: 'i \text{ set}) \rangle \langle |UNIV :: 'p \text{ set}| \leq o |UNIV :: 'i \text{ set}| \rangle$

shows $\langle \text{valid } p \longleftrightarrow \Box \vdash_{@} (i, p) \rangle$

$\langle \text{proof} \rangle$

corollary *main'*:

fixes $p :: \langle ('i, 'i) \text{ fm} \rangle$

assumes $\langle i \notin \text{nominals-fm } p \rangle \langle \text{infinite } (UNIV :: 'i \text{ set}) \rangle$

shows $\langle \text{valid } p \longleftrightarrow \Box \vdash_{@} (i, p) \rangle$

$\langle \text{proof} \rangle$

end

Chapter 8

Example: First-Order Logic

theory *Example-First-Order-Logic* imports *Derivations* begin

8.1 Syntax

datatype (*params-tm*: 'f) *tm*
= *Var nat* (<#>)
| *Fun 'f* <'f *tm list*> (<·>)

abbreviation *Const* (<★>) **where** <★*a* ≡ ·*a* []>

datatype (*params-fm*: 'f, 'p) *fm*
= *Fls* (<⊥>)
| *Pre 'p* <'f *tm list*> (<·>)
| *Imp* <'f, 'p> *fm*> <'f, 'p> *fm*> (**infixr** <→> 55)
| *Exi* <'f, 'p> *fm*> (<∃>)

abbreviation *Neg* (<¬ -> [70] 70) **where** <¬ *p* ≡ *p* → ⊥>

8.2 Semantics

type-synonym ('a, 'f, 'p) *model* = <(nat ⇒ 'a) × ('f ⇒ 'a list ⇒ 'a) × ('p ⇒ 'a list ⇒ bool)>

fun *semantics-tm* :: <(nat ⇒ 'a) × ('f ⇒ 'a list ⇒ 'a) ⇒ 'f *tm* ⇒ 'a> (<[-]>) **where**
| <[(*E*, -)] (#*n*) = *E n*>
| <[(*E*, *F*)] (·*f ts*) = *F f* (map [(*E*, *F*)] *ts*)>

primrec *add-env* :: <'a ⇒ (nat ⇒ 'a) ⇒ nat ⇒ 'a> (**infix** <§> 0) **where**
| <(t § *s*) 0 = *t*>
| <(t § *s*) (*Suc n*) = *s n*>

fun *semantics-fm* :: <'a, 'f, 'p> *model* ⇒ ('f, 'p) *fm* ⇒ bool (**infix** <|=₃> 50) **where**
| <- |=₃ ⊥ ↔ *False*>
| <(*E*, *F*, *G*) |=₃ ·*P ts* ↔ *G P* (map [(*E*, *F*)] *ts*)>
| <(*E*, *F*, *G*) |=₃ *p* → *q* ↔ (*E*, *F*, *G*) |=₃ *p* → (*E*, *F*, *G*) |=₃ *q*>
| <(*E*, *F*, *G*) |=₃ ∃*p* ↔ (∃*x*. (*x* § *E*, *F*, *G*) |=₃ *p*)>

8.3 Operations

primrec *lift-tm* :: <'f *tm* ⇒ 'f *tm*> **where**

$\langle \text{lift-tm } (\#n) = \#(n+1) \rangle$
 $\mid \langle \text{lift-tm } (\cdot f \text{ ts}) = \cdot f (\text{map lift-tm ts}) \rangle$

primrec *sub-tm* :: $\langle (\text{nat} \Rightarrow 'f \text{ tm}) \Rightarrow 'f \text{ tm} \Rightarrow 'f \text{ tm} \rangle$ **where**
 $\langle \text{sub-tm } s (\#n) = s \ n \rangle$
 $\mid \langle \text{sub-tm } s (\cdot f \text{ ts}) = \cdot f (\text{map } (\text{sub-tm } s) \text{ ts}) \rangle$

primrec *sub-fm* :: $\langle (\text{nat} \Rightarrow 'f \text{ tm}) \Rightarrow ('f, 'p) \text{ fm} \Rightarrow ('f, 'p) \text{ fm} \rangle$ **where**
 $\langle \text{sub-fm } - \perp = \perp \rangle$
 $\mid \langle \text{sub-fm } s (\cdot P \text{ ts}) = \cdot P (\text{map } (\text{sub-tm } s) \text{ ts}) \rangle$
 $\mid \langle \text{sub-fm } s (p \longrightarrow q) = \text{sub-fm } s \ p \longrightarrow \text{sub-fm } s \ q \rangle$
 $\mid \langle \text{sub-fm } s (\exists p) = \exists (\text{sub-fm } (\#0 \circ \lambda n. \text{lift-tm } (s \ n)) \ p) \rangle$

abbreviation *inst-single* :: $\langle 'f \text{ tm} \Rightarrow ('f, 'p) \text{ fm} \Rightarrow ('f, 'p) \text{ fm} \rangle$ ($\langle \langle - \rangle \rangle$) **where**
 $\langle \langle t \rangle \equiv \text{sub-fm } (t \circ \#) \rangle$

abbreviation $\langle \text{params } S \equiv \bigcup p \in S. \text{params-fm } p \rangle$

abbreviation $\langle \text{params}' l \equiv \text{params } (\text{set } l) \rangle$

lemma *upd-params-tm [simp]*: $\langle f \notin \text{params-tm } t \Longrightarrow \llbracket (E, F(f := x)) \rrbracket t = \llbracket (E, F) \rrbracket t \rangle$
 $\langle \text{proof} \rangle$

lemma *upd-params-fm [simp]*: $\langle f \notin \text{params-fm } p \Longrightarrow (E, F(f := x), G) \models_{\exists} p \longleftrightarrow (E, F, G) \models_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-params-tm [simp]*: $\langle \text{finite } (\text{params-tm } t) \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-params-fm [simp]*: $\langle \text{finite } (\text{params-fm } p) \rangle$
 $\langle \text{proof} \rangle$

lemma *env [simp]*: $\langle P ((x \circ E) \ n) = (P \ x \circ \lambda n. P (E \ n)) \ n \rangle$
 $\langle \text{proof} \rangle$

lemma *lift-lemma*: $\langle \llbracket (x \circ E, F) \rrbracket (\text{lift-tm } t) = \llbracket (E, F) \rrbracket t \rangle$
 $\langle \text{proof} \rangle$

lemma *sub-tm-semantics*: $\langle \llbracket (E, F) \rrbracket (\text{sub-tm } s \ t) = \llbracket (\lambda n. \llbracket (E, F) \rrbracket (s \ n), F) \rrbracket t \rangle$
 $\langle \text{proof} \rangle$

lemma *sub-fm-semantics [simp]*: $\langle (E, F, G) \models_{\exists} \text{sub-fm } s \ p \longleftrightarrow (\lambda n. \llbracket (E, F) \rrbracket (s \ n), F, G) \models_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

lemma *sub-tm-Var [simp]*: $\langle \text{sub-tm } \# \ t = t \rangle$
 $\langle \text{proof} \rangle$

lemma *reduce-Var [simp]*: $\langle (\# \ 0 \circ \lambda n. \# (Suc \ n)) = \# \rangle$
 $\langle \text{proof} \rangle$

lemma *sub-fm-Var [simp]*:
fixes $p :: \langle ('f, 'p) \text{ fm} \rangle$
shows $\langle \text{sub-fm } \# \ p = p \rangle$
 $\langle \text{proof} \rangle$

lemma *semantics-tm-id [simp]*: $\langle \llbracket (\#, \cdot) \rrbracket t = t \rangle$

⟨proof⟩

lemma *semantics-tm-id-map* [simp]: ⟨map ((#, ·)) ts = ts⟩
⟨proof⟩

The built-in *size* is not invariant under substitution.

primrec *size-fm* :: ⟨('f, 'p) fm ⇒ nat⟩ **where**
| *size-fm* ⊥ = 1
| *size-fm* (·- ·) = 1
| *size-fm* (p → q) = 1 + *size-fm* p + *size-fm* q
| *size-fm* (∃ p) = 1 + *size-fm* p

lemma *size-sub-fm* [simp]: ⟨*size-fm* (sub-fm s p) = *size-fm* p⟩
⟨proof⟩

8.4 Calculus

inductive *Calculus* :: ⟨('f, 'p) fm list ⇒ ('f, 'p) fm ⇒ bool⟩ (**infix** ⟨⊢_∃⟩ 50) **where**
| *Assm* [simp]: ⟨p ∈ set A ⇒ A ⊢_∃ p⟩
| *FlsE* [elim]: ⟨A ⊢_∃ ⊥ ⇒ A ⊢_∃ p⟩
| *ImpI* [intro]: ⟨p # A ⊢_∃ q ⇒ A ⊢_∃ p → q⟩
| *ImpE* [dest]: ⟨A ⊢_∃ p → q ⇒ A ⊢_∃ p ⇒ A ⊢_∃ q⟩
| *ExiI* [intro]: ⟨A ⊢_∃ ⟨t⟩p ⇒ A ⊢_∃ ∃ p⟩
| *ExiE* [elim]: ⟨A ⊢_∃ ∃ p ⇒ a ∉ params (set (p # q # A)) ⇒ ⟨★a⟩p # A ⊢_∃ q ⇒ A ⊢_∃ q⟩
| *Clas*: ⟨(p → q) # A ⊢_∃ p ⇒ A ⊢_∃ p⟩

8.4.1 Weakening

abbreviation ⟨*psub* f ≡ map-fm f id⟩

lemma *map-tm-sub-tm* [simp]: ⟨map-tm f (sub-tm g t) = sub-tm (map-tm f o g) (map-tm f t)⟩
⟨proof⟩

lemma *map-tm-lift-tm* [simp]: ⟨map-tm f (lift-tm t) = lift-tm (map-tm f t)⟩
⟨proof⟩

lemma *psub-sub-fm*: ⟨*psub* f (sub-fm g p) = sub-fm (map-tm f o g) (*psub* f p)⟩
⟨proof⟩

lemma *map-tm-inst-single*: ⟨(map-tm f o (u ∩ #)) t = (map-tm f u ∩ #) t⟩
⟨proof⟩

lemma *psub-inst-single* [simp]: ⟨*psub* f (⟨t⟩p) = ⟨map-tm f t⟩(*psub* f p)⟩
⟨proof⟩

lemma *map-tm-upd* [simp]: ⟨a ∉ params-tm t ⇒ map-tm (f(a := b)) t = map-tm f t⟩
⟨proof⟩

lemma *psub-upd* [simp]: ⟨a ∉ params-fm p ⇒ *psub* (f(a := b)) p = *psub* f p⟩
⟨proof⟩

class *inf-univ* =
| **fixes** *itself* :: ⟨'a itself⟩
| **assumes** *infinite-UNIV*: ⟨infinite (UNIV :: 'a set)⟩

lemma *Calculus-psub*:

fixes $f :: \langle 'f \Rightarrow 'g :: \text{inf-univ} \rangle$

shows $\langle A \vdash_{\exists} p \Longrightarrow \text{map } (\text{psub } f) A \vdash_{\exists} \text{psub } f p \rangle$

$\langle \text{proof} \rangle$

lemma *Weaken*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$

shows $\langle A \vdash_{\exists} p \Longrightarrow \text{set } A \subseteq \text{set } B \Longrightarrow B \vdash_{\exists} p \rangle$

$\langle \text{proof} \rangle$

8.5 Soundness

theorem *soundness*: $\langle A \vdash_{\exists} p \Longrightarrow \forall q \in \text{set } A. (E, F, G) \models_{\exists} q \Longrightarrow (E, F, G) \models_{\exists} p \rangle$

$\langle \text{proof} \rangle$

corollary *soundness'*: $\langle [] \vdash_{\exists} p \Longrightarrow M \models_{\exists} p \rangle$

$\langle \text{proof} \rangle$

corollary $\langle \neg ([] \vdash_{\exists} \perp) \rangle$

$\langle \text{proof} \rangle$

8.6 Admissible Rules

lemma *Assm-head*: $\langle p \# A \vdash_{\exists} p \rangle$

$\langle \text{proof} \rangle$

lemma *Boole*: $\langle (\neg p) \# A \vdash_{\exists} \perp \Longrightarrow A \vdash_{\exists} p \rangle$

$\langle \text{proof} \rangle$

corollary *Weak*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$

shows $\langle A \vdash_{\exists} p \Longrightarrow q \# A \vdash_{\exists} p \rangle$

$\langle \text{proof} \rangle$

lemma *deduct1*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$

shows $\langle A \vdash_{\exists} p \longrightarrow q \Longrightarrow p \# A \vdash_{\exists} q \rangle$

$\langle \text{proof} \rangle$

lemma *Weak'*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$

shows $\langle A \vdash_{\exists} p \Longrightarrow B @ A \vdash_{\exists} p \rangle$

$\langle \text{proof} \rangle$

interpretation *Derivations* $\langle \text{Calculus} :: ('f :: \text{inf-univ}, 'p) \text{ fm list} \Rightarrow \rightarrow \rangle$

$\langle \text{proof} \rangle$

8.7 Maximal Consistent Sets

definition *consistent* :: $\langle ('f, 'p) \text{ fm set} \Rightarrow \text{bool} \rangle$ **where**

$\langle \text{consistent } S \equiv \forall A. \text{ set } A \subseteq S \longrightarrow \neg A \vdash_{\exists} \perp \rangle$

fun *witness* :: $\langle ('f, 'p) \text{ fm} \Rightarrow ('f, 'p) \text{ fm set} \Rightarrow ('f, 'p) \text{ fm set} \rangle$ **where**

$\langle \text{witness } (\exists p) S = (\text{let } a = \text{SOME } a. a \notin \text{params } (\{p\} \cup S) \text{ in } \{\langle \star a \rangle p\}) \rangle$

| $\langle \text{witness } - = \{ \} \rangle$

lemma *consistent-add-witness*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
assumes $\langle \text{consistent } S \rangle \langle \exists p \in S \rangle \langle a \notin \text{params } S \rangle$
shows $\langle \text{consistent } (\{ \star a \} \cup S) \rangle$
 $\langle \text{proof} \rangle$

lemma *consistent-witness'*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
assumes $\langle \text{consistent } (\{ p \} \cup S) \rangle \langle \text{infinite } (\text{UNIV} - \text{params } S) \rangle$
shows $\langle \text{consistent } (\text{witness } p \ S \cup \{ p \} \cup S) \rangle$
 $\langle \text{proof} \rangle$

interpretation *MCS-Witness-UNIV consistent witness* $\langle \text{params-fm} :: ('f :: \text{inf-univ}, 'p) \text{ fm} \Rightarrow - \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Cut-MCS consistent* $\langle \text{Calculus} :: ('f :: \text{inf-univ}, 'p) \text{ fm list} \Rightarrow - \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Bot consistent Calculus* $\langle \perp :: ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Imp consistent Calculus* $\langle \lambda p \ q. p \longrightarrow q :: ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
 $\langle \text{proof} \rangle$

interpretation *Derivations-Exi consistent witness params-fm Calculus* $\langle \exists \rangle \langle \lambda t \ p. \langle t \rangle p :: ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
 $\langle \text{proof} \rangle$

8.8 Truth Lemma

abbreviation *canonical* $:: \langle ('f, 'p) \text{ fm set} \Rightarrow ('f \text{ tm}, 'f, 'p) \text{ model} \rangle \langle \mathcal{M}_\exists \rangle$ **where**
 $\langle \mathcal{M}_\exists(S) \equiv (\#, \cdot, \lambda P \text{ ts. } \cdot P \text{ ts} \in S) \rangle$

fun *semics* $::$

$\langle ('a, 'f, 'p) \text{ model} \Rightarrow (('a, 'f, 'p) \text{ model} \Rightarrow ('f, 'p) \text{ fm} \Rightarrow \text{bool}) \Rightarrow ('f, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$
 $\langle \langle - \llbracket - \rrbracket_\exists - \rangle [55, 0, 55] 55 \rangle$ **where**
 $\langle - \llbracket - \rrbracket_\exists \perp \longleftrightarrow \text{False} \rangle$
 $| \langle (E, F, G) \llbracket - \rrbracket_\exists \cdot P \text{ ts} \longleftrightarrow G \ P \ (\text{map } \llbracket (E, F) \rrbracket \text{ ts}) \rangle$
 $| \langle (E, F, G) \llbracket \mathcal{R} \rrbracket_\exists p \longrightarrow q \longleftrightarrow \mathcal{R} \ (E, F, G) \ p \longrightarrow \mathcal{R} \ (E, F, G) \ q \rangle$
 $| \langle (E, F, G) \llbracket \mathcal{R} \rrbracket_\exists \exists p \longleftrightarrow (\exists x. \mathcal{R} \ (x \circ E, F, G) \ p) \rangle$

fun *rel* $:: \langle ('f, 'p) \text{ fm set} \Rightarrow ('f \text{ tm}, 'f, 'p) \text{ model} \Rightarrow ('f, 'p) \text{ fm} \Rightarrow \text{bool} \rangle \langle \mathcal{R}_\exists \rangle$ **where**
 $\langle \mathcal{R}_\exists(S) \ (E, -, -) \ p \longleftrightarrow \text{sub-fm } E \ p \in S \rangle$

theorem *saturated-model*:

assumes $\langle \bigwedge p. \forall M \in \{ \mathcal{M}_\exists(S) \}. M \llbracket \mathcal{R}_\exists(S) \rrbracket_\exists p \longleftrightarrow \mathcal{R}_\exists(S) \ M \ p \rangle \langle M \in \{ \mathcal{M}_\exists(S) \} \rangle$
shows $\langle \mathcal{R}_\exists(S) \ M \ p \longleftrightarrow M \models_\exists p \rangle$
 $\langle \text{proof} \rangle$

theorem *saturated-MCS*:

fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
assumes $\langle \text{MCS } S \rangle$
shows $\langle \mathcal{R}_\exists(S) \ (\mathcal{M}_\exists(S)) \ p \longleftrightarrow \mathcal{M}_\exists(S) \llbracket \mathcal{R}_\exists(S) \rrbracket_\exists p \rangle$

⟨proof⟩

interpretation *Truth-Witness semics semantics-fm* $\langle \lambda S. \{\mathcal{M}_\exists(S)\} \rangle$ *rel consistent witness*

⟨params-fm :: ('f :: inf-univ, 'p) fm \Rightarrow -⟩

⟨proof⟩

8.9 Cardinalities

datatype *marker* = *VarM* | *FunM* | *TmM* | *FlsM* | *PreM* | *ImpM* | *ExiM*

type-synonym ('f, 'p) *enc* = $\langle ('f + 'p) + \text{marker} \times \text{nat} \rangle$

abbreviation $\langle \text{FUNS } f \equiv \text{Inl } (\text{Inl } f) \rangle$

abbreviation $\langle \text{PRES } p \equiv \text{Inl } (\text{Inr } p) \rangle$

abbreviation $\langle \text{VAR } n \equiv \text{Inr } (\text{VarM}, n) \rangle$

abbreviation $\langle \text{FUN } n \equiv \text{Inr } (\text{FunM}, n) \rangle$

abbreviation $\langle \text{TM } n \equiv \text{Inr } (\text{TmM}, n) \rangle$

abbreviation $\langle \text{PRE } n \equiv \text{Inr } (\text{PreM}, n) \rangle$

abbreviation $\langle \text{FLS} \equiv \text{Inr } (\text{FlsM}, 0) \rangle$

abbreviation $\langle \text{IMP } n \equiv \text{Inr } (\text{FlsM}, n) \rangle$

abbreviation $\langle \text{EXI} \equiv \text{Inr } (\text{ExiM}, 0) \rangle$

primrec

encode-tm :: $\langle 'f \text{ tm} \Rightarrow ('f, 'p) \text{ enc list} \rangle$ **and**

encode-tms :: $\langle 'f \text{ tm list} \Rightarrow ('f, 'p) \text{ enc list} \rangle$ **where**

⟨*encode-tm* (#*n*) = [VAR *n*]⟩

| ⟨*encode-tm* (\cdot *ts*) = FUN (length *ts*) # FUNS *f* # *encode-tms ts*⟩

| ⟨*encode-tms* [] = []⟩

| ⟨*encode-tms* (*t* # *ts*) = TM (length (encode-tm *t*)) # *encode-tm t* @ *encode-tms ts*⟩

lemma *encode-tm-ne* [*simp*]: $\langle \text{encode-tm } t \neq [] \rangle$

⟨proof⟩

lemma *inj-encode-tm'*:

⟨(encode-tm *t* :: ('f, 'p) enc list) = encode-tm *s* $\implies t = s$ ⟩

⟨(encode-tms *ts* :: ('f, 'p) enc list) = encode-tms *ss* $\implies ts = ss$ ⟩

⟨proof⟩

lemma *inj-encode-tm*: $\langle \text{inj } \text{encode-tm} \rangle$

⟨proof⟩

primrec *encode-fm* :: $\langle ('f, 'p) \text{ fm} \Rightarrow ('f, 'p) \text{ enc list} \rangle$ **where**

⟨*encode-fm* \perp = [FLS]⟩

| ⟨*encode-fm* (\cdot *P ts*) = PRE (length *ts*) # PRES *P* # *encode-tms ts*⟩

| ⟨*encode-fm* (*p* \longrightarrow *q*) = IMP (length (encode-fm *p*)) # *encode-fm p* @ *encode-fm q*⟩

| ⟨*encode-fm* (\exists *p*) = EXI # *encode-fm p*⟩

lemma *encode-fm-ne* [*simp*]: $\langle \text{encode-fm } p \neq [] \rangle$

⟨proof⟩

lemma *inj-encode-fm'*: $\langle \text{encode-fm } p = \text{encode-fm } q \implies p = q \rangle$

⟨proof⟩

lemma *inj-encode-fm*: $\langle \text{inj encode-fm} \rangle$
 $\langle \text{proof} \rangle$

lemma *finite-marker*: $\langle \text{finite (UNIV :: marker set)} \rangle$
 $\langle \text{proof} \rangle$

lemma *card-of-fm*:
 $\langle |UNIV :: ('f :: \text{inf-univ}, 'p) \text{ fm set}| \leq o |UNIV :: 'f \text{ set}| + c |UNIV :: 'p \text{ set}| \rangle$
 $\langle \text{proof} \rangle$

8.10 Completeness

theorem *strong-completeness*:
assumes $\langle \forall M :: ('f \text{ tm}, 'f :: \text{inf-univ}, 'p) \text{ model. } (\forall q \in X. M \models_{\exists} q) \longrightarrow M \models_{\exists} p \rangle$
 $\langle |UNIV :: 'f \text{ set}| + c |UNIV :: 'p \text{ set}| \leq o |UNIV - \text{params } X| \rangle$
shows $\langle \exists A. \text{set } A \subseteq X \wedge A \vdash_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

abbreviation *valid* :: $\langle ('f, 'p) \text{ fm} \Rightarrow \text{bool} \rangle$ **where**
 $\langle \text{valid } p \equiv \forall M :: ('f \text{ tm}, -, -) \text{ model. } M \models_{\exists} p \rangle$

theorem *completeness*:
fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
assumes $\langle \text{valid } p \rangle$ $\langle |UNIV :: 'p \text{ set}| \leq o |UNIV :: 'f \text{ set}| \rangle$
shows $\langle [] \vdash_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

corollary *completeness'*:
fixes $p :: \langle ('f :: \text{inf-univ}, 'f) \text{ fm} \rangle$
assumes $\langle \text{valid } p \rangle$
shows $\langle [] \vdash_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

theorem *main*:
fixes $p :: \langle ('f :: \text{inf-univ}, 'p) \text{ fm} \rangle$
assumes $\langle |UNIV :: 'p \text{ set}| \leq o |UNIV :: 'f \text{ set}| \rangle$
shows $\langle \text{valid } p \longleftrightarrow [] \vdash_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

corollary *main'*:
fixes $p :: \langle ('f :: \text{inf-univ}, 'f) \text{ fm} \rangle$
shows $\langle \text{valid } p \longleftrightarrow [] \vdash_{\exists} p \rangle$
 $\langle \text{proof} \rangle$

end

Bibliography

- [1] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
- [2] J. C. Blanchette, A. Popescu, and D. Traytel. Cardinals in Isabelle/HOL. In G. Klein and R. Gamboa, editors, *Interactive Theorem Proving - 5th International Conference, ITP 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8558 of *Lecture Notes in Computer Science*, pages 111–127. Springer, 2014.
- [3] T. Braüner. *Hybrid Logic and its Proof-Theory*. Springer Dordrecht, first edition, 2011.
- [4] C. C. Chang and H. J. Keisler. *Model theory, Third Edition*, volume 73 of *Studies in logic and the foundations of mathematics*. North-Holland, 1992.
- [5] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
- [6] R. M. Smullyan. *First-order logic*. Dover Publications, 1995.