Mechanised Modal Model Theory

Yiming Xu[1,3] and Michael Norrish[2,1] [0000−0003−1163−8467]

[1] Australian National University
[2] michael.norrish@data61.csiro.au
Data61, CSIRO
[3] u5943321@163.com

Abstract. In this paper, we discuss the mechanisation of some fundamental propositional modal model theory. The focus on models is novel: previous work in mechanisations of modal logic have centered on proof systems and applications in model-checking. We have mechanised a number of fundamental results from the first two chapters of a standard textbook (Blackburn *et al.* [1]). Among others, one important result, the Van Benthem characterisation theorem, characterises the connection between modal logic and first order logic. This latter captures the desired saturation property of ultraproduct models on countably incomplete ultrafilters.

1 Introduction

The theory of modal logic has long been a fruitful area when it comes to mechanisation. The proof systems are appealing, and the applications in model-checking are of clear real-world interest. It helps also that the subject domain (proof calculi and automata) are well-suited to "standard" theorem-proving technology (rule inductions and interesting data types).

There has been much less work on the model theory behind modal logic; indeed even in first order logic, most developments concern themselves only with model theory inasmuch as it is required to show completeness of an accompanying proof system. As our experience demonstrates, it is also clear that modern theorem-proving systems are not necessarily so well-suited to the mathematics behind model theory. Harrison [5] complained in 1998 that the very notion of validity is awkward to capture in HOL, and our own work shows up further failings in simple type theory.

Nonetheless, there is much interesting mathematics to be found even in the early chapters of a standard text such as Blackburn *et al.* [1]. The fact that mechanising only as far as [1, Chapter 2] requires what we believe to be the first mechanisation of the notion of ultraproduct (ultimately leading to Łoś's theorems and other results), is a strong suggestion that we are exploring novel mathematical ground for interactive theorem-proving systems.

Contributions This paper presents the first mechanised proofs of a number of basic results from the first two chapters of Blackburn *et al.* [1] (e.g., bounded

morphisms, bisimulations and the finite model property *via* selection), as well as

- Two versions of Łoś's theorem on the saturation of ultraproduct models;
- modal equivalence as bisimilarity between ultrafilter extensions; and
- a close approximation of Van Benthem's Characterisation Theorem.

We also discuss where HOL's simple type theory lets us down: some standard results (including the best possible statement of Van Benthem's Characterisation Theorem) seem impossible to prove in our setting.

HOL4 Notation All of our theorems have been pretty-printed to LaTeX from the HOL theory files. We hope that most of the basic syntax is easy to follow. In a few places we use CHOICE s to denote the arbitrary choice of an element from set s (appealing to the Axiom of Choice). The power-set of a set s is written $\mathcal{P}\ s$. In a number of places, we use HOL's "itself" type to allow us to explicitly mention a type *via* a term. The type α `itself` has just one value inhabiting it for any given choice of α; that value is written $(:\alpha)$.

Source Availability Our HOL4 sources are available from GitHub at

https://github.com/u5943321/Modal-Logic

The sources build under HOL4 commit with SHA `03829d8986f`.

2 Syntax, Semantics and the Standard Translation

In our mechanisation, we consider the basic modal language, in which the only primitive modal operator is the '$\Diamond$'. A modal formula is either of form $\mathsf{V_m}\ p$, where p is of type `num`, enumerating all the possible variable symbols, a disjunction DISJ $\phi\ \psi$ (pretty-printed to $\phi \vee_{\mathsf{m}} \psi$ in most places), the falsity $\bot_{\mathsf{m}}$, a negation $\neg_{\mathsf{m}}\ \phi$, or, finally, of the form $\Diamond\phi$. We define a data type called `form` to represent the formulas of this modal language.

Definition 1. [1, Definition 1.9]

$$form\ =\ \mathsf{V_m}\ num \mid \mathsf{DISJ}\ form\ form \mid \bot_{\mathsf{m}} \mid \neg_{\mathsf{m}}\ form \mid \Diamond\ form$$

If we wanted to consider modal operators with any arity, we should change the last constructor of modal formulas so it takes two parameters: a natural number indexing the modal operator, and a list of modal formulas. This would in turn require a well-formedness predicate to be defined over formulas to make sure that modalities were applied to the right number of arguments.

A model where these formulas can be interpreted consists of a *frame* and a *valuation*, where a β `frame` is a β-set with a relation on it, and a model adds valuations for the variables present at each world:

Definition 2. [1, Definition 1.19]

$$\beta\ \mathit{frame} = \langle\!\langle\ \mathsf{world} : \beta \to \mathit{bool};\ \mathsf{rel} : \beta \to \beta \to \mathit{bool}\ \rangle\!\rangle$$
$$\beta\ \mathit{model} = \langle\!\langle\ \mathsf{frame} : \beta\ \mathit{frame};\ \mathsf{valt} : \mathit{num} \to \beta \to \mathit{bool}\ \rangle\!\rangle$$

In the rest of the paper, the field $M.\mathsf{valt}$ of a model M will be called the *valuation*, and M^W, M^R and M^V are used to denote the world set, the relation, and the valuation of M respectively. The interpretation of modal formulas on a model is given by the predicate *satisfaction*. We read '$M, w \Vdash \phi$' as 'ϕ is satisfied at the world w in M'.

Definition 3. [1, Definition 1.20]

$$\begin{aligned}
M, w \Vdash \mathsf{V_m}\ p &\stackrel{\text{def}}{=} w \in M^W \wedge w \in M^V\ p \\
M, w \Vdash \bot_{\mathsf{m}} &\stackrel{\text{def}}{=} \mathsf{F} \\
M, w \Vdash \neg_{\mathsf{m}}\ \phi &\stackrel{\text{def}}{=} w \in M^W \wedge M, w \nVdash \phi \\
M, w \Vdash (\phi_1 \vee_{\mathsf{m}} \phi_2) &\stackrel{\text{def}}{=} M, w \Vdash \phi_1 \vee M, w \Vdash \phi_2 \\
M, w \Vdash \Diamond\phi &\stackrel{\text{def}}{=} w \in M^W \wedge \exists\, v.\ M^R\ w\ v \wedge v \in M^W \wedge M, v \Vdash \phi
\end{aligned}$$

By requiring $w \in M^W$ in various clauses above, we ensure that models' world sets must be non-empty if they are to satisfy any formulas.

Two worlds $w_1 \in M_1^W$ and $w_2 \in M_2^W$ are *modal equivalent* (written $M_1, w_1 \leftrightsquigarrow M_2, w_2$) if they satisfy the same set of modal formulas. If ϕ_1, ϕ_2 are modal formulas, we say they are *equivalent* over β models (written $\phi_1 \equiv_{(:\beta)} \phi_2$) if they are satisfied in the same worlds in every model:

Definition 4 (Notions of equivalence).

$$M, w \leftrightsquigarrow M', w' \stackrel{\text{def}}{=} \forall\, \phi.\ M, w \Vdash \phi \iff M', w' \Vdash \phi$$

$$\begin{aligned}
&(\phi_1 : \mathit{form}) \equiv_{(:\beta)} (\phi_2 : \mathit{form}) \stackrel{\text{def}}{=} \\
&\quad \forall\, (M : \beta\ \mathit{model})\ (w : \beta).\ M, w \Vdash \phi_1 \iff M, w \Vdash \phi_2
\end{aligned}$$

We cannot omit the type parameter $(:\beta)$ in the definition, as there would otherwise be a type, namely the type of the underlying set of the models we are talking about, that only appears on the right-hand side but not on the left-hand side of the definition. HOL forbids such definitions for soundness reasons. Also, HOL does not permit quantification over types, so it is impossible to write $\forall\, \mu.\ \phi_1 \equiv_\mu \phi_2$, with μ a type. Therefore, this definition is not exactly encoding the equivalence in the usual sense: when we mention equivalence of formulas in usual mathematical language, we are implicitly referring to the class of all models, but the constraint here bans us from talking about all models of all possible types at once.

A modal formula can be translated into a first-order formula via the *standard translation*. To mechanise this translation, we build on Harrison's construction of first-order logic [5]. The first-order connectives are decorated with an f. A first order model M is a set $M.\mathsf{Dom}$ with interpretation of function symbols $M.\mathsf{Fun}$

and predicate symbols $M.\mathsf{Pred}$. A valuation σ of M is a function that maps all the natural numbers into the domain of M. If a first-order formula ϕ is satisfied in a first-order model M with σ a valuation assigning free variables of ϕ elements in the domain of M, we write $M, \sigma \vDash \phi$.

For a modal formula ϕ, $\mathsf{ST}_x\ \phi$ is the standard translation of ϕ using x as the only free variable that may occur:

Definition 5. [1, Definition 2.45 (Standard Translation)]

$$\begin{aligned}
\mathsf{ST}_x\ (\mathsf{V_m}\ p) &\stackrel{\mathsf{def}}{=} \mathsf{P_f}\ p\ (\mathsf{V_f}\ x) \\
\mathsf{ST}_x\ \bot_{\mathsf{m}} &\stackrel{\mathsf{def}}{=} \bot_{\mathsf{f}} \\
\mathsf{ST}_x\ (\neg_{\mathsf{m}}\ \phi) &\stackrel{\mathsf{def}}{=} \neg_{\mathsf{f}}\ (\mathsf{ST}_x\ \phi) \\
\mathsf{ST}_x\ (\phi \vee_{\mathsf{m}} \psi) &\stackrel{\mathsf{def}}{=} \mathsf{ST}_x\ \phi \vee_{\mathsf{f}} \mathsf{ST}_x\ \psi \\
\mathsf{ST}_x\ (\Diamond \phi) &\stackrel{\mathsf{def}}{=} \exists_{\mathsf{f}}\ (x\ +\ 1)\ (\mathsf{R_f}\ (\mathsf{V_f}\ x)\ (\mathsf{V_f}\ (x\ +\ 1)) \wedge_{\mathsf{f}} \mathsf{ST}_{x\,+\,1}\ \phi)
\end{aligned}$$

As one would expect, we translate $\Diamond\phi$ into an existential formula. To ensure we use a fresh variable, we use $x+1$ as our new variable symbol in this clause. The standard translation gives a first-order reformulation of satisfaction of modal formulas:

Proposition 1. [1, Theorem 2.47 (i)]

$$\vdash M, w \Vdash \phi \iff \mathsf{mm2folm}\ M, (\lambda\, n.\ w) \vDash \mathsf{ST}_x\ \phi$$

Here mm2folm is the function that turns a modal model into a first-order model, defined as:

$$\begin{array}{l}
\mathsf{mm2folm}\ M \stackrel{\mathsf{def}}{=} \\
\ \langle\!\langle \mathsf{Dom}\ :=\ M^W;\ \mathsf{Fun}\ :=\ (\lambda\, n\ l.\ \mathsf{CHOICE}\ M^W); \\
\ \ \mathsf{Pred}\ := \\
\ \ \ (\lambda\, p\ zs. \\
\ \ \ \ \mathtt{case}\ zs\ \mathtt{of} \\
\ \ \ \ \ [\,] \Rightarrow \mathsf{F} \\
\ \ \ \ |\ [w_1] \Rightarrow w_1 \in M^W \wedge M^V\ p\ w_1 \\
\ \ \ \ |\ [w_1;\ w_2] \Rightarrow p = 0 \wedge M^R\ w_1\ w_2 \wedge w_1 \in M^W \wedge w_2 \in M^W \\
\ \ \ \ |\ w_1 :: w_2 :: w_3 :: ws \Rightarrow \mathsf{F})\rangle\!\rangle
\end{array}$$

That is, the model obtained by converting a modal model M has domain M^W, maps every term $\mathsf{Fn_f}\ f\ l$ into an arbitary world, maps each propositional letter to distinct predicates on worlds, and uses one binary predicate (the "0th predicate") to encode the frame relation.

3 Basic Results

We discuss some highlights of mechanised results from Blackburn *at al.* [1, §2.1–§2.3] below.

3.1 Tree-like property

A tree-like model is a model whose underlying frame is a tree. If Tr, a frame, is also a tree with root r, we write $\mathsf{tree}\ Tr\ r$:

Definition 6. [1, Definition 1.7]

$$\begin{array}{l}\mathsf{tree}\ Tr\ r \stackrel{\mathsf{def}}{=} \\ \quad r \in Tr.\mathsf{world} \wedge (\forall w.\ w \in Tr.\mathsf{world} \Rightarrow Tr.\mathsf{rel}\,|_{Tr.\mathsf{world}}{}^{*}\ r\ w) \wedge \\ \quad (\forall w.\ w \in Tr.\mathsf{world} \Rightarrow \neg Tr.\mathsf{rel}\ w\ r) \wedge \\ \quad \forall w.\ w \in Tr.\mathsf{world} \wedge w \neq r \Rightarrow \exists! w_0.\ w_0 \in Tr.\mathsf{world} \wedge Tr.\mathsf{rel}\ w_0\ w\end{array}$$

The tree-like property says an satisfiable modal formula can be satisfied in a tree-like model:

Proposition 2. [1, Proposition 2.15]

$$\begin{array}{l}\vdash (M : \beta\ \mathit{model}), (w : \beta) \Vdash\ (\phi : \mathit{form}) \Rightarrow \\ \quad \exists (M' : \beta\ \mathit{list}\ \mathit{model})\ (r : \beta\ \mathit{list}).\ \mathsf{tree}\ M'.\mathsf{frame}\ r \wedge M', r \Vdash \phi\end{array}$$

The world set of the tree-like model constructed from M is a set of lists of worlds in M (such lists are effectively paths from the root to various positions within the tree). Thus, passing to a tree-like model does not preserve the model type. The tree-like lemma is used to prove the finite model property via selection afterwards.

3.2 Bisimulation

Though apparently verbose, the definition of bisimulation in HOL is straightforward.

Definition 7. [1, Definition 2.16 (Bisimulations)]

$$\begin{array}{l}M_1 \overset{Z}{\leftrightarrows} M_2 \stackrel{\mathsf{def}}{=} \\ \quad \forall w_1\ w_2. \\ \quad\quad w_1 \in M_1^W \wedge w_2 \in M_2^W \wedge Z\ w_1\ w_2 \Rightarrow \\ \quad\quad (\forall p.\ M_1, w_1 \Vdash \mathsf{V_m}\ p \iff M_2, w_2 \Vdash \mathsf{V_m}\ p) \wedge \\ \quad\quad (\forall v_1. \\ \quad\quad\quad v_1 \in M_1^W \wedge M_1^R\ w_1\ v_1 \Rightarrow \\ \quad\quad\quad\quad \exists v_2.\ v_2 \in M_2^W \wedge Z\ v_1\ v_2 \wedge M_2^R\ w_2\ v_2) \wedge \\ \quad\quad \forall v_2. \\ \quad\quad\quad v_2 \in M_2^W \wedge M_2^R\ w_2\ v_2 \Rightarrow \\ \quad\quad\quad\quad \exists v_1.\ v_1 \in M_1^W \wedge Z\ v_1\ v_2 \wedge M_1^R\ w_1\ v_1\end{array}$$

$$M, w \leftrightarrows M', w' \stackrel{\mathsf{def}}{=} \exists Z.\ M \overset{Z}{\leftrightarrows} M' \wedge w \in M^W \wedge w' \in M'^W \wedge Z\ w\ w'$$

It is trivial to prove by induction that bisimilar worlds are modal equivalent. As the most significant theorem on the basic theory of bisimulations, we proved the Hennessy-Milner theorem, which states that modal equivalence and bisimulation on *image finite* models are the same thing. An image-finite model is a model where every world can only be related to finitely many worlds. In HOL, we get:

Theorem 1. [1, Theorem 2.24 (Hennessy-Milner Theorem)]

$$\vdash \mathsf{image\text{-}finite}\ M_1\ \wedge\ \mathsf{image\text{-}finite}\ M_2\ \wedge\ w_1\ \in\ M_1^W\ \wedge\ w_2\ \in\ M_2^W\ \Rightarrow \\ (M_1, w_1 \leftrightsquigarrow M_2, w_2\ \iff\ M_1, w_1 \leftrightarrows M_2, w_2)$$

Bisimulation is an interesting topic in modal logic. Three other significant theorems on bisimulations (including an approximation of Van Benthem Characterisation theorem) are discussed later.

3.3 Finite model property

There are two classical approaches to constructing finite models using model theory, namely via selection and via filtration. The complicated one is the former: Given $M_1, w_1 \Vdash \phi$, where ϕ has degree k, we can construct M_2, M_3, M_4 and M_5 comsecutively, such that M_5 is the finite model we want, where:

- M_2 is the tree-like model obtained from Proposition 2 with root w_2 such that $M_2, w_2 \Vdash \phi$.
- M_3 is the restriction of M_2 to height k.
- M_4 is obtained from M_3 by modifying the valuation into $\lambda\, p\ v.\ \mathtt{if}\ p\ \in\ \mathsf{prop\text{-}letters}\ \phi\ \mathtt{then}\ {M_3}^V\ p\ v\ \mathtt{else}\ \mathsf{F}$, where $\mathsf{prop\text{-}letters}\ \phi$ is the set of all propositional letters used by ϕ.

The construction of M_5 requires a lemma:

Lemma 1. [1, Proposition 2.29]

$$\vdash \mathsf{FINITE}\ (\Phi : \mathit{num} \to \mathit{bool})\ \wedge\ \mathsf{INFINITE}\ \mathcal{U}(:\beta)\ \Rightarrow \\ \forall\,(n : \mathit{num}).\ \mathsf{FINITE}\ \{\ \phi \mid \mathsf{DEG}\ \phi\ \leq\ n\ \wedge\ \mathsf{prop\text{-}letters}\ \phi\ \subseteq\ \Phi\ \}\,/{\equiv_{(:\beta)}}$$

The proof of Lemma 1 further relies on the following fact: Given a set A of modal-formulas that is finite up to equivalence, if we combine the elements of A using only connectives other than $\Diamond$, then we get only finitely many non-equivalent formulas. To show this, we prove that there is an injection from the set of equivalence classes of such combinations to a finite set. For the antecedent of Lemma 1, we require the assumption that the universe of β is infinite since we rely on the fact that two modal formulas $\Diamond\phi_1$ and $\Diamond\phi_2$ are equivalent if and only if ϕ_1 and ϕ_2 are equivalent. This would be easy to prove in set theory. However, in simple type theory, the proof of $\phi_1 \equiv_{(:\beta)} \phi_2$ iff $\Diamond\phi_1 \equiv_{(:\beta)} \Diamond\phi_2$ requires us (in the left-to-right direction) to be able to construct a model with a new world inserted, something only sure to be possible if the β universe is infinite. As the construction used Proposition 2, we change the type of the model by passing to a finite model via selection:

Theorem 2. [1, Theorem 2.34 (Finite model property, via selection)]

$$\vdash (M_1 : \beta\ \mathit{model}), (w_1 : \beta) \Vdash\ (\phi : \mathit{form})\ \Rightarrow \\ \exists\,(M : \beta\ \mathit{list}\ \mathit{model})\ (v : \beta\ \mathit{list}).\ \mathsf{FINITE}\ M^W\ \wedge\ v\ \in\ M^W\ \wedge\ M, v \Vdash\ \phi$$

We also mechanised the filtration approach, but omit the details for lack of space. The advantage of filtration is that the resulting finite model is over worlds of the same type as in the starting model.

All the results proved above can be captured using `num` models everywhere. If one takes β to be `num` (or any infinite type) in Theorem 2, one can also exploit the fact that numbers and lists of numbers have the same cardinality to derive a finite model result that preserves the "input type".

4 Mechanising Ultrafilters and Ultraproducts

A number of results in Blackburn *et al.* [1, §2.5–§2.7] rely on theorems about ultrafilters and ultraproducts.

4.1 Ultrafilters

Given a non-empty set J, a set $L \subseteq \mathcal{P}\ J$ is a *filter* if it contains J itself, is closed under binary intersection, and is closed upward.

Definition 8. [1, Definition A.12 (Filters)]

$$\begin{array}{l}\mathsf{filter}\ L\ J \stackrel{\mathsf{def}}{=} \\ \quad J \neq \emptyset \wedge L \subseteq \mathcal{P}\ J \wedge J \in L \wedge \\ \quad (\forall X\ Y.\ X \in L \wedge Y \in L \Rightarrow X \cap Y \in L) \wedge \\ \quad \forall X\ Z.\ X \in L \wedge X \subseteq Z \wedge Z \subseteq J \Rightarrow Z \in L\end{array}$$

We call L a *proper filter* if L is not the whole power set. An *ultrafilter* is a filter U such that for every $X \subseteq J$, exactly one of X or $J \setminus X$ is in U. Intuitively, subsets $X \subseteq J$ in an ultrafilter U are considered as 'large' subsets of J.

The ultrafilter theorem states that every proper filter is contained in an ultrafilter:

Theorem 3. [1, Fact A.14, first half]

$$\vdash \mathsf{proper\text{-}filter}\ L\ J \Rightarrow \exists U.\ \mathsf{ultrafilter}\ U\ J \wedge L \subseteq U$$

(The proof uses Zorn's Lemma.)

A subset A of the power set on J has *finite intersection property* if once we take the intersection of a finite, nonempty family in A, the resultant set is nonempty.

Definition 9. [1, Definition A.13 (Finite Intersection Property)]

$$\begin{array}{l}\vdash \mathsf{FIP}\ A\ J \iff \\ \quad A \subseteq \mathcal{P}\ J \wedge \forall B.\ B \subseteq A \wedge \mathsf{FINITE}\ B \wedge B \neq \emptyset \Rightarrow \bigcap B \neq \emptyset\end{array}$$

As a corollary of ultrafilter theorem, a set with finite intersection property is contained in an ultrafilter.

4.2 Ultraproducts

The notion of ultraproducts is defined for sets, modal models, and first-order models.

Ultraproduct of sets A family of sets indexed by J is a function A^{s} in HOL. For $j \in J$, $A^{\mathsf{s}}\ j$ is the set indexed by j. Given a family A^{s} indexed by a non-empty set J such that each $A^{\mathsf{s}}\ j$ is non-empty, the ultraproduct of A^{s} is defined as a quotient of the cartesian product of the family.

Definition 10. [1, Page 495 (Cartesian product)]

$$\mathsf{Cart\text{-}prod}\ J\ A^{\mathsf{s}} \stackrel{\mathsf{def}}{=} \{\ f \mid \forall j.\ j \in J \Rightarrow f\ j \in A^{\mathsf{s}}\ j\ \}$$

If U is an ultrafilter on J, for two functions f, g in the Cartesian product $\mathsf{Cart\text{-}prod}\ J\ A^{\mathsf{s}}$, we say f and g are U-equivalent (notation: $f \sim_U^{A^{\mathsf{s}}} g$) if the set $\{\ j \mid j \in J \wedge f\ j = g\ j\ \}$ (where the values of f and g agree) is in U. The ultraproduct of A^{s} modulo U is the quotient of $\mathsf{Cart\text{-}prod}\ J\ A^{\mathsf{s}}$ by $\sim_U^{A^{\mathsf{s}}}$.

Definition 11. [1, Definition 2.69 (Ultraproduct of Sets)]

$$\mathsf{ultraproduct}\ U\ J\ A^{\mathsf{s}} \stackrel{\mathsf{def}}{=} \mathsf{Cart\text{-}prod}\ J\ A^{\mathsf{s}} / \sim_U^{A^{\mathsf{s}}}$$

We write f_U to denote the equivalence class that f belongs to. In the case where $A^{\mathsf{s}}\ j = A$ for all $j \in J$, the ultraproduct is called the ultrapower of A modulo U.

Ultraproduct for modal models Given a family M^{s} of modal models indexed by J and an ultrafilter U on J, the ultraproduct model of M^{s} modulo U (notation: $\Pi_U\ M^{\mathsf{s}}$) is described as follows:

- The world set is the ultraproduct of world sets of M^{s} modulo U.
- Two equivalence classes f_U, g_U of functions are related in $\Pi_U\ M^{\mathsf{s}}$ iff there exist $f_0 \in f_U, g_0 \in g_U$, such that $\{\ j \in J \mid (M^{\mathsf{s}}\ j)^R\ (f_0\ j)\ (g_0\ j)\ \}$ is in U.
- A propositional letter p is satisfied at f_U in $\Pi_U\ M^{\mathsf{s}}$ iff there exists $f_0 \in f_U$ such that $\{\ j \mid j \in J \wedge f_0\ j \in (M^{\mathsf{s}}\ j)^V\ p\ \}$ is in U.

Definition 12. [1, Definition 2.70 (Ultraproduct of Modal Models)]

$$\begin{array}{l}
\Pi_U\ M^{\mathsf{s}} \stackrel{\mathsf{def}}{=} \\
\quad \langle\!\langle \mathsf{frame} := \\
\qquad \langle\!\langle \mathsf{world} := \mathsf{ultraproduct}\ U\ J\ (\lambda j.\ (M^{\mathsf{s}}\ j)^W); \\
\qquad\ \mathsf{rel} := \\
\qquad\quad (\lambda f_U\ g_U. \\
\qquad\qquad \exists f_0\ g_0. \\
\qquad\qquad\quad f_0 \in f_U \wedge g_0 \in g_U \wedge \\
\qquad\qquad\quad \{\ j \mid j \in J \wedge (M^{\mathsf{s}}\ j)^R\ (f_0\ j)\ (g_0\ j)\ \} \in U)\rangle\!\rangle; \\
\qquad \mathsf{valt} := \\
\qquad\ (\lambda p\ f_U.\ \exists f_0.\ f_0 \in f_U \wedge \{\ j \mid j \in J \wedge f_0\ j \in (M^{\mathsf{s}}\ j)^V\ p\ \} \in U)\rangle\!\rangle
\end{array}$$

As $\sim_U^A$ is an equivalence relation, if one element in an equivalence class satisfies the required condition, then all the elements in the equivalence class will satisfy the condition. Therefore, if we replace all the existential quantifiers with universal quantifiers in the above definition, the construction is still valid, and will give the same model as the current definition.

The critical result we need about ultraproducts of modal models is a modal version of the fundamental theorem of ultraproducts, also known as Łoś's theorem.

Theorem 4 (Łoś's theorem, Modal version).

$$\begin{array}{l}\vdash \mathsf{ultrafilter}\ U\ J\ \wedge\ f_U\ \in\ (\prod_U\ M^{\mathsf{s}})^W\ \Rightarrow\\ \quad (\prod_U\ M^{\mathsf{s}}, f_U \Vdash\ \phi\ \Longleftrightarrow\\ \quad\quad \exists f_0.\ f_0\ \in\ f_U\ \wedge\ \{\ j \mid j\ \in\ J\ \wedge\ M^{\mathsf{s}}\ j, f_0\ j \Vdash\ \phi\ \}\ \in\ U)\end{array}$$

According to our intuition about ultrafilters, we can gloss this theorem to mean that the ultraproduct of a family of modal models satisfies a modal formula if and only if 'most of' the models in the family satisfy the formula. Though it is possible to derive this result from Łoś's theorem of first-order models using the standard translation, our proof is direct, by structural induction on ϕ.

Ultraproducts for first-order models Given a family M^{s} of first-order models indexed by J and an ultrafilter U on J, the ultraproduct model of M^{s} modulo U (notation: ${}^{\mathsf{f}}\prod_U\ M^{\mathsf{s}}$) is given by:

- The domain is the ultraproduct of the domains of M^{s} over U on J.
- A function named by symbol (natural number) n sends a list zs of equivalence classes to the equivalence class of a function that sends $j \in J$ to $(M^{\mathsf{s}}\ j).\mathsf{Fun}\ n\ l$, where the k-th member of the list l is a representative of the k-th member (which is an equivalence class) of zs.
- A predicate named by symbol p will hold for a list zs of equivalence classes if and only if when we have a list zr, where k-th member is a representative of the k-th member of zs, the set of elements $j\ \in\ J$ such that $(M^{\mathsf{s}}\ j).\mathsf{Pred}\ p\ zr$ is in U.

Definition 13. [1, Definition A.18 (Ultraproduct of First-Order Models)]

$$\begin{array}{l}{}^{\mathsf{f}}\prod_U\ M^{\mathsf{s}}\ \stackrel{\mathsf{def}}{=}\\ \quad \langle\!\langle \mathsf{Dom}\ :=\ \mathsf{ultraproduct}\ U\ J\ (\lambda j.\ (M^{\mathsf{s}}\ j).\mathsf{Dom});\\ \quad\ \mathsf{Fun}\ :=\\ \quad\quad (\lambda\, n\ zs.\\ \quad\quad\quad \{\ y \mid\\ \quad\quad\quad\ (\forall j.\ j\ \in\ J\ \Rightarrow\ y\ j\ \in\ (M^{\mathsf{s}}\ j).\mathsf{Dom})\ \wedge\\ \quad\quad\quad\ \{\ j \mid j\ \in\ J\ \wedge\ y\ j\ =\ \mathsf{Fun\text{-}component}\ M^{\mathsf{s}}\ n\ zs\ j\ \}\ \in\ U\ \});\\ \quad\ \mathsf{Pred}\ :=\ (\lambda\, p\ zs.\ \{\ j \mid j\ \in\ J\ \wedge\ \mathsf{Pred\text{-}component}\ M^{\mathsf{s}}\ p\ zs\ j\ \}\ \in\ U)\rangle\!\rangle\end{array}$$

Here we fix the representative of each equivalence class f_U to be $\mathsf{CHOICE}\ f_U$. Therefore, as described above, the functions Fun-component and Pred-component are:

$$\mathsf{Fun\text{-}component}\ M^{\mathsf{s}}\ n\ fs\ i \stackrel{\mathsf{def}}{=} (M^{\mathsf{s}}\ i).\mathsf{Fun}\ n\ (\mathsf{MAP}\ (\lambda f_U.\ \mathsf{CHOICE}\ f_U\ i)\ fs)$$

$$\mathsf{Pred\text{-}component}\ M^{\mathsf{s}}\ p\ zs\ i \stackrel{\mathsf{def}}{=} (M^{\mathsf{s}}\ i).\mathsf{Pred}\ p\ (\mathsf{MAP}\ (\lambda f_U.\ \mathsf{CHOICE}\ f_U\ i)\ zs)$$

The semantic behavior of ultraproduct models is characterised by Łoś's theorem: for the ultraproduct of a family M^{s} of first-order models over an ultrafilter U on J, a formula ϕ is satisfied under a valuation σ if and only if the set indexing the models $M^{\mathsf{s}}\ j$ in the family where ϕ is true under the valuation $\lambda v.\ \mathsf{CHOICE}\ (\sigma\ v)\ j$ is in the ultrafilter U.

Theorem 5. [1, Theorem A.19 (Łoś's theorem)]

$$\begin{array}{l}\vdash \mathsf{ultrafilter}\ U\ J\ \wedge\ \mathsf{valuation}\ ({}^{\mathsf{f}}\prod_U M^{\mathsf{s}})\ \sigma\ \wedge \\ \quad (\forall j.\ j\ \in\ J\ \Rightarrow\ \mathsf{wffm}\ (M^{\mathsf{s}}\ j))\ \Rightarrow \\ \quad\quad ({}^{\mathsf{f}}\prod_U M^{\mathsf{s}}, \sigma \vDash\ \phi\ \Longleftrightarrow \\ \quad\quad\quad \{\ j \mid j\ \in\ J\ \wedge\ M^{\mathsf{s}}\ j, (\lambda v.\ \mathsf{CHOICE}\ (\sigma\ v)\ j) \vDash\ \phi\ \}\ \in\ U)\end{array}$$

where $\mathsf{wffm}\ M$ means the functions of M never map a list out of the domain of M.

5 Ultrafilter Extensions

The first application of the theory of ultrafilters above is to construct the ultrafilter extension of a model, which has the nice property of being *modally saturated* (m-saturated hereafter). To define m-saturation, we give the following three definitions (the first two are called *finitely satisfiable*, *satisfiable*) consecutively:

Definition 14. [1, Definition 2.53]

$$\begin{array}{l}\mathsf{satisfiable\text{-}in}\ \Sigma\ X\ M \stackrel{\mathsf{def}}{=} \\ \quad X\ \subseteq\ M^W\ \wedge\ \exists w.\ w\ \in\ X\ \wedge\ \forall \phi.\ \phi\ \in\ \Sigma\ \Rightarrow\ M, w \Vdash\ \phi\end{array}$$

$$\mathsf{fin\text{-}satisfiable\text{-}in}\ \Sigma\ X\ M \stackrel{\mathsf{def}}{=} \forall S.\ S\ \subseteq\ \Sigma\ \wedge\ \mathsf{FINITE}\ S\ \Rightarrow\ \mathsf{satisfiable\text{-}in}\ S\ X\ M$$

$$\begin{array}{l}\mathsf{m\text{-}sat}\ M \stackrel{\mathsf{def}}{=} \\ \quad \forall w\ \Sigma. \\ \quad\quad w\ \in\ M^W\ \wedge\ \mathsf{fin\text{-}satisfiable\text{-}in}\ \Sigma\ \{\ v \mid v\ \in\ M^W\ \wedge\ M^R\ w\ v\ \}\ M\ \Rightarrow \\ \quad\quad\quad \mathsf{satisfiable\text{-}in}\ \Sigma\ \{\ v \mid v\ \in\ M^W\ \wedge\ M^R\ w\ v\ \}\ M\end{array}$$

For m-saturated models, bisimulation and modal equivalence coincide:

Proposition 3. [1, Proposition 2.54]

$$\begin{array}{l}\vdash \mathsf{m\text{-}sat}\ M_1\ \wedge\ \mathsf{m\text{-}sat}\ M_2\ \wedge\ w_1\ \in\ M_1^W\ \wedge\ w_2\ \in\ M_2^W\ \Rightarrow \\ \quad (M_1, w_1 \leftrightsquigarrow M_2, w_2\ \Longleftrightarrow\ M_1, w_1 \leftrightarrows M_2, w_2)\end{array}$$

Given a model M and a set X of worlds of M, the set of worlds that 'can see' X (notation: $M_\Diamond(X)$) is the set of worlds w of M such that there exists some $v \in X$ such that $M^R\ w\ v$. We define the ultrafilter extension ${}^{ue}M$ of M as:

- The world set is the set of all ultrafilters on M^W.
- Two ultrafilters u, v on M are related in the ultrafilter extension of M if for every $X \in v$, the set of worlds that can see X is in u.
- A propositional letter p to be satisfied at an ultrafilter v if and only if the set of worlds in M which satisfies p is in v.

In HOL:

Definition 15. [1, Definition 2.57 (Ultrafilter Extension)]

$$
\begin{array}{l}
{}^{ue}M \stackrel{\text{def}}{=} \\
\quad \langle\!\langle \mathsf{frame} := \\
\quad\quad \langle\!\langle \mathsf{world} := \{\ u \mid \mathsf{ultrafilter}\ u\ M^W\ \}; \\
\quad\quad\ \ \mathsf{rel} := \\
\quad\quad\quad (\lambda\, u\ v. \\
\quad\quad\quad\quad \mathsf{ultrafilter}\ u\ M^W \wedge \mathsf{ultrafilter}\ v\ M^W \wedge \\
\quad\quad\quad\quad \forall X.\ X \in v \Rightarrow M_\Diamond(X) \in u) \rangle\!\rangle; \\
\quad\ \ \mathsf{valt} := (\lambda\, p\ v.\ \mathsf{ultrafilter}\ v\ M^W \wedge \{\ w \mid w \in M^W \wedge M^V\ p\ w\ \} \in v) \rangle\!\rangle
\end{array}
$$

Using the ultrafilter theorem and some basic properties about ultrafilters, we derive:

Proposition 4. [1, Proposition 2.59 (i)]

$$
\begin{array}{l}
\vdash \mathsf{ultrafilter}\ u\ M^W \Rightarrow \\
\quad (\{\ w \mid w \in M^W \wedge M, w \Vdash \phi\ \} \in u \iff {}^{ue}M, u \Vdash \phi)
\end{array}
$$

In particular, every world $w \in M^W$ is embedded as the principal filter $\pi_w^{M^W}$ on M^W generated by w in the ultrafilter extension or M. Also, the above leads to the proof of the fact that the ultrafilter extension of every model is m-saturated. The m-saturatedness of ultrafilter extensions together with Proposition 3 immediately gives the central result about ultrafilter extension: bisimilarity of worlds in a model M can be characterised as bisimilarity in ${}^{ue}M$.

Theorem 6. [1, Proposition 2.62]

$$
\begin{array}{l}
\vdash w_1 \in M_1^W \wedge w_2 \in M_2^W \Rightarrow \\
\quad (M_1, w_1 \leftrightsquigarrow M_2, w_2 \iff {}^{ue}M_1, \pi_{w_1}^{M_1^W} \leftrightarrows {}^{ue}M_2, \pi_{w_2}^{M_2^W})
\end{array}
$$

6 Countable Saturatedness of Ultrapower Models

Given a first-order model M with no information about interpretation of its function symbols, we can expand the model M by adding an interpretation of some function symbols. For our purpose, we are only interested in adding the

interpretation of finitely many nullary function symbols, also called *constants*. We write $\mathsf{expand}\ M\ A\ f$ to denote the model that is the result of adding each element in A to M as a new constant. Further, the function f is a bijection between $\{0, \cdots, n-1\}$ and A, which is assumed to be finite, so that each nullary function symbol c will be interpreted as $f\ c$ in M'.

Definition 16. [1, Definition A.9 (Expansion)]

$$\begin{array}{l}\mathsf{expand}\ M\ A\ f \stackrel{\mathsf{def}}{=} \\ \ \langle\!\langle \mathsf{Dom} := M.\mathsf{Dom}; \\ \ \ \mathsf{Fun} := \\ \ \ \ (\lambda\, c\ l.\ \mathtt{if}\ c < \mathsf{CARD}\ A \wedge l = [\,]\ \mathtt{then}\ f\ c\ \mathtt{else}\ \mathsf{CHOICE}\ M.\mathsf{Dom}); \\ \ \ \mathsf{Pred} := M.\mathsf{Pred} \rangle\!\rangle\end{array}$$

As is apparent from the definition, the only difference between a model and its expansion is the interpretation of function symbols.

A set Σ of first-order formulas is called *consistent* with a model M if for every finite subset $\Sigma_0 \subseteq \Sigma$, there exists a valuation of M such that all elements of Σ_0 are satisfied, in this case, we write $\mathsf{consistent}\ M\ \Sigma$. A set Γ of first-order formula is an *x-type* if for each formula in Γ, the only free variable that may contain is x. In this case, we write '$\mathsf{ftype}\ x\ \Gamma$' in HOL. If Γ is an x-type, when evaluating formulas in Γ, the valuations will only control where the only free variable x goes to. We say Γ is *realised* in M if there is an element w in the domain of M such that $M, (\lambda\, v.\ w) \vDash \phi$ for all $\phi \in \Gamma$. In this case, we write '$\mathsf{frealises}\ M\ x\ \Gamma$' in HOL. Let M be a model and n be a natural number. If for every $A \subseteq M.\mathsf{Dom}$ with $|A| < n$ and every $f : \mathbb{N} \to M.\mathsf{Dom}$, the model $\mathsf{expand}\ M\ A\ f$ realises every x-type Γ that is consistent with $\mathsf{expand}\ M\ A\ f$, then we say M is n-saturated. In HOL:

Definition 17. [1, Definition 2.63 (n-Saturated)]

$$\begin{array}{l}\mathsf{n\text{-}saturated}\ M\ n \stackrel{\mathsf{def}}{=} \\ \ \forall\, A\ \Gamma\ x\ f. \\ \ \ \ \mathsf{IMAGE}\ f\ \mathcal{U}(:\mathit{num}) \subseteq M.\mathsf{Dom} \wedge \mathsf{FINITE}\ A \wedge \mathsf{CARD}\ A \leq n \wedge \\ \ \ \ A \subseteq M.\mathsf{Dom} \wedge \mathsf{BIJ}\ f\ (\mathsf{count}\ (\mathsf{CARD}\ A))\ A \wedge \\ \ \ \ (\forall\, \phi.\ \phi \in \Gamma \Rightarrow \mathsf{form\text{-}functions}\ \phi \subseteq \{\, (c, 0) \mid c < \mathsf{CARD}\ A \,\}) \wedge \\ \ \ \ \mathsf{ftype}\ x\ \Gamma \wedge \mathsf{consistent}\ (\mathsf{expand}\ M\ A\ f)\ \Gamma \Rightarrow \\ \ \ \ \ \mathsf{frealises}\ (\mathsf{expand}\ M\ A\ f)\ x\ \Gamma\end{array}$$

We say M is countably saturated if M is n-saturated for every natural number n. The ultimate goal is to prove a lemma to be used in the proof of Van Benthem characterisation theorem: For a family of non-empty models, their ultraproduct on a countably incomplete ultrafilter is *countably saturated*.

Lemma 2. [1, Lemma 2.73]

$$\begin{array}{l}\vdash (\forall\, j.\ j \in J \Rightarrow (M^{\mathsf{s}}\ j)^W \neq \emptyset) \wedge \mathsf{countably\text{-}incomplete}\ U\ J \Rightarrow \\ \ \ \ \mathsf{countably\text{-}saturated}\ (\mathsf{mm2folm}\ (\Pi_U\ M^{\mathsf{s}}))\end{array}$$

Here a countably incomplete ultrafilter is an ultrafilter that contains a countably infinite family that intersects to the empty set. We prove in HOL that such ultrafilters do exist using Theorem 3. The above theorem is not simply a direct consequence of Łoś's theorem: that result is about ultraproducts of first-order models, and it says nothing about expansion. But to prove Lemma 2, we must prove a statement for an expanded first-order model, and this first order model is itself obtained by converting a ultraproduct of modal models.

To deal with this issue, the key observation is that constants are nothing more than forcing some symbols to be sent to some points in a model under every valuation, hence rather than use nullary function symbols, we fix a set of variable letters, each corresponding to a function symbol, and only consider the valuations that send these variable letters to certain fixed points. With this idea, we can remove all the constants in a formula, and hence change our scope from an expanded model back to the unexpanded model. To get rid of the constants $\{0, \cdots, n-1\}$, we replace every $\mathsf{V_f}\ m$ with $\mathsf{V_f}\ (m\ +\ n)$, and replace every constant $\mathsf{Fn_f}\ c\ [\,]$ by $\mathsf{V_f}\ c$. This operation is done by the function shift-form which takes a natural number (the number of constants we want to remove), and a first-order formula (where the only function symbols may appear are the constants $0, \cdots, n-1$). Since $0, \cdots, n-1$ in a shifted formula are now designed to be sent to fixed places $f\ 0, \cdots, f\ (n-1)$, it does not make sense to assign these variable symbols anywhere else. Therefore, to talk about evaluation of shifted formula, the first thing is to make sure that the valuations we are considering send the variables which actually denote constants to the right place. Hence we shift the valuations accordingly, and then prove that a formula is satisfied on an expanded model is satisfied under a valuation if and only if the shifted formula is satisfied under the shifted valuation. Also, we prove that 'taking the ultraproduct first-order model commutes with the convertion from modal to a first-order model on certain formulas', in the sense that the resulting models satisfies the same first-order formulas without function symbols. By putting these two results together, we prove Lemma 2 using the proof in Chang and Keisler [3].

7 Van Benthem's Characterisation Theorem

Note that the standard translation of any modal formula can only contain unary predicate symbols which correspond to propositional letters, one binary predicate symbol which corresponds to the relation, and no function symbols. A first-order formula which only uses these symbols is called an $\mathcal{L}^1_\tau$-formula. An $\mathcal{L}^1_\tau$-formula which contains only one free variable is called *invariant under bisimulation* if for all models M and N with $w\ \in\ M^W$ and $v\ \in\ N^W$, if there exists a bisimulation relation between M and N relating w and v, then ϕ holds at w if and only if it holds at v when both M and N are viewed as first-order models.

Definition 18. [1, Definition 2.67 (Invariant for Bisimulations)]

$$\begin{array}{l}\mathsf{invar4bisim}\ (x : \mathit{num})\ (:\alpha)\ (:\beta)\ (\phi : \mathit{folform}) \stackrel{\mathsf{def}}{=} \\ \quad \mathsf{FV}\ \phi\ \subseteq\ \{\ x\ \}\ \wedge\ \mathcal{L}^1_\tau\ \phi\ \wedge \\ \quad \forall\,(M : \alpha\ \mathit{model})\ (N : \beta\ \mathit{model})\ (v : \beta)\ (w : \alpha). \\ \qquad M, w \leftrightarrows N, v\ \Rightarrow \\ \qquad\quad (\mathsf{mm2folm}\ M, (\lambda\,(x : \mathit{num}).\ w) \vDash\ \phi\ \iff \\ \qquad\qquad \mathsf{mm2folm}\ N, (\lambda\,(x : \mathit{num}).\ v) \vDash\ \phi)\end{array}$$

Because of the same problem we met when defining equivalence of formulas, the type parameters are necessary here. However, although it is possible to prove theorems for different types α and β in the above definition, in the theorems to come, we will only consider the case where α and β are the same.

The Van Benthem characterisation theorem says an $\mathcal{L}^1_\tau$ formula with at most one free variable x is invariant under bisimulation precisely when it is equivalent to the standard translation of some modal formula at x. It is immediate from Proposition 1 that every such formula which is equivalent to a standard translation is invariant for bisimulation. We cannot prove it as an 'if and only if' statement, since according to the proofs in [1], we can only prove the two directions separately as:

Proposition 5. [1, Theorem 2.68, as two separate directions]

$$\vdash \mathsf{FV}\ \delta\ \subseteq\ \{\ x\ \}\ \wedge\ \mathcal{L}^1_\tau\ \delta\ \wedge\ \delta\ {}^{\mathsf{f}}\!\equiv_{(:\alpha)}\ \mathsf{ST}_x\ \phi\ \Rightarrow\ \mathsf{invar4bisim}\ x\ (:\alpha)\ (:\alpha)\ \delta$$

$$\begin{array}{l}\vdash \mathsf{INFINITE}\ \mathcal{U}(:\alpha)\ \wedge \\ \quad \mathsf{invar4bisim}\ x\ (:(\mathit{num} \to \alpha) \to \mathit{bool})\ (:(\mathit{num} \to \alpha) \to \mathit{bool})\ \delta\ \Rightarrow \\ \qquad \exists\,\phi.\ \delta\ {}^{\mathsf{f}}\!\equiv_{(:\alpha)}\ \mathsf{ST}_x\ \phi\end{array}$$

which cannot be put together into a double implication. To see the reason: given an $\mathcal{L}^1_\tau$-formula ϕ with no more then one free variable, by the second theorem above, if ϕ is invariant under bisimulation for models with $(\mathtt{num} \to \alpha) \to \mathtt{bool}$-worlds, then ϕ is equivalent to a standard translation on a model with α-worlds. However, if we want to prove the converse of this statement, we need to start with the assumption that ϕ is equivalent to a standard translation on models with α-worlds, and prove that ϕ is invariant for bisimulation for models with $(\mathtt{num} \to \alpha) \to \mathtt{bool}$-worlds. But by the first theorem above, we can only conclude ϕ is invariant for bisimulation for models of type α. The point is that it is not the fact that all our desired operations can be taken within a type. In particular, we cannot take ultraproducts of models and preserve cardinalities. The cardinality of the type universe of $(\mathtt{num} \to \alpha) \to \mathtt{bool}$ is too large to be embedded into α, so we cannot just fix the 'base type' to be α and get an 'if and only if' statement-we cannot derive ϕ is invariant for bisimulation for models with $(\mathtt{num} \to \alpha) \to \mathtt{bool}$-worlds from the fact that ϕ is invariant for bisimulation for models with α-worlds. If we could quantify over types (as we could in a theorem prover based on dependent type theory), then we can could define 'invariant under bisimulation for models of every type', and hence prove the original statement of Van Benthem characterisation theorem.

For the proof of the two theorems above, the first one is immediate from Proposition 1, and the second one requires another critical lemma saying 'modal equivalence between two worlds implies bisimilarity of the two worlds when embedded in some other models'. More precisely, if two worlds $w \in M^W$ and $v \in N^W$ are modal equivalent, then we can find an ultrafilter U on J such that in ultrapower models of M and N on U respectively, there is a bisimulation between the worlds corresponding to w and v.

Theorem 7. [1, Theorem 2.74, one direction]

$$\begin{array}{l} \vdash w \in M^W \wedge v \in N^W \wedge (\forall \phi.\, M, w \Vdash \phi \iff N, v \Vdash \phi) \Rightarrow \\ \quad \exists U\ J. \\ \qquad \mathsf{ultrafilter}\ U\ J \wedge \\ \qquad \Pi_U\ (\lambda j.\, M), \{\, f \mid (\lambda j.\, w) \sim_U^{\mathsf{worlds}\ (\lambda j.\, M)} f\ \} \leftrightarrows \Pi_U\ (\lambda j.\, N), \{\, g \mid \\ \qquad (\lambda j.\, v) \sim_U^{\mathsf{worlds}\ (\lambda j.\, N)} g\ \} \end{array}$$

The proof of the above relies on Lemma 2.

8 Conclusion

To summarise, we have mechanised all of the results (appearing as propositions, lemmas and theorems) in the first two chapters in Blackburn *at al.* [1] that can be captured by the HOL logic, and which are about the basic modal language. The exceptions are:

- The result in Section 2.6 about 'definability', which requires a definition of the 'models closed under taking ultraproducts'. Simple type theory cannot capture such large sets.
- The result about 'safety' in Section 2.7 is a result about the PDL language, which has infinitely many modal operators. For the moment, we have restricted our attention to the basic modal language, with only $\Diamond$ (and the derived $\Box$).

The two characterisation theorems from Blackburn *et al.* [1], namely Theorem 2.68 (Van Benthem's Characterisation Theorem) and Theorem 2.78, are the only two mechanised theorem such that translating the 'if and only if' statements from set theory into simple theory theory does not yield an 'if and only if' statement. Blackburn *et al.*'s proof of Theorem 2.78 has the same pattern as Van Benthem's Characterisation Theorem (discussed earlier), and is less complicated.

For each of the mechanised definitions and results, we write the statement in HOL to be as close as possible to the original statement in [1]. We believe that this makes it as easy as possible for people who are interested in mechanising other results in [1] to continue with our work as a starting point. The work on ultraproducts up to Łoś's theorem is independent of our work on modal model theory, and should be generally useful in other model-theoretic applications.

8.1 Related Work

We believe that we are the first to mechanise the bulk of the results in this paper. Of course, much work has been done in this and similar areas. For example, de Wind's thesis [7] is a notable early mechanisation of modal logic, mainly focusing on proving the validity of modal formulas via natural deduction. Of similar vintage is Harrison's mechanisation of foundational results about first order model theory [5], in particular compactness. We used this mechanisation directly in our own work. A great deal of work has also been done in the mechanisation of first order *proof* theory, such as the recent pearl by Blanchette *et al.* [2], showing completeness in elegant fashion.

The connections between modal logic and process algebra are well-understood and there has been a great deal of mechanised work on the operational theory of such (co-)algebraic systems, starting at least as far back as Nesi [6]. Our proof of the Hennessy-Milner theorem (Theorem 1) is a gesture in this direction, but Van Benthem's theorem is much deeper and uses bisimulations as a tool to understanding the connection between modal and first order logics, rather than as a connection to process algebras.

Mechanised work with ultrafilters began with Fleuriot's use of them to mechanise non-standard analysis [4]. We are unaware of any previous mechanised use of ultraproducts or ultrapowers.

References

1. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
2. Blanchette, J.C., Popescu, A., Traytel, D.: Unified classical logic completeness: A coinductive pearl. In: IJCAR 2014. pp. 46–60. No. 8562 in Lecture Notes in Computer Science, Springer (2014)
3. Chang, C.C., Keisler, H.J.: Model Theory. North Holland (1990)
4. Fleuriot, J.: A Combination of Geometry Theorem Proving and Nonstandard Analysis with Application to Newton's Principia. Springer (2001). https://doi.org/10.1007/978-0-85729-329-9
5. Harrison, J.: Formalizing basic first order model theory. In: Theorem Proving in Higher Order Logics, 11th International Conference. pp. 153–170. No. 1479 in Lecture Notes in Computer Science, Springer (1998)
6. Nesi, M.: Mechanising a modal logic for value-passing agents in HOL. Electr. Notes Theor. Comput. Sci. **5**, 31–46 (1996). https://doi.org/10.1016/S1571-0661(05)80682-6
7. de Wind, P.: Modal Logic in Coq. Master's thesis, Vrije Universiteit (2001)