Modal Logic Mechanisation in HOL4

Yiming Xu

15/6/2018

0 Introduction

There is four questions to anwser in order to make sense of our title:

0.1 What is modal logic?

In our formalisation, we restrict our scope into propositional modal logic. Propositional modal logic is a type of formal logic extending propositional logic by adding modalities that express the mode of truth value, and we also restrict to the simplest modalites 'possibly', and 'necessarily', which is written as '$\Box$' and '$\Diamond$'. Given a propositional symbol 'p', '$\Diamond p$' reads 'it is possible that p', and '$\Box p$' reads 'it is necessarily p'.

We follow the textbook 'Modal Logic' by Blackburn et al, the authers give us three nice slogans:

> Slogan 1: Modal languages are simple yet expressive languages for talking about relational structures, where a relational structure is simply a set together with a relation defined on it.
> Slogan 2: Modal languages provide an internal, local perspective on relational structures.
> Slogan 3: Modal languages are not isolated formal systems.

We will see the evidences of those slogans consecutively in the body of the article. In chapter 1 and 2, we investigate what can our modal formulas express about relational structures, and what are the relations between relational structures and satisfication of modal formulas on relational sturctures with certain relation. In chapter 3, we prove the modal satisfication is intrinsically local. And after connecting our world of modal logic to the wide logical word, in chapter 5 we borrow tools from set theory, first order logic and model theory to prove some characterisation theorems about our modal formulas.

0.2 What is the HOL?

HOL is a proof assistant written in standard meta language. A proof assistant is a machine which can allow us give definitions, state theorems and prove them in the language that the machine can understand. Once we prove some theorem in the HOL, we can store them for using them later, the files we write in the HOL are called script files, and the definitions and proofs we have built can be stored as a file called _Theory.sml.

The underlying system of the HOL is plain type theory, it does not support dependent type theory, which means that we cannot quantify over types. So it can never let us say something that 'for all types α, define

.. as ...'. This gives arise to some inconvenience, but fortunately does not really trigger actual problems in our formalisation. As we will see, sometimes it cause the demand of extra assumption on the universe of types. But it has its own advantage: HOL is much faster then many theorem provers which supports dependent type theory, for instance, lean and Coq. Another advantage is that since the plain type theory is much simpler than dependent type theory, it makes theorems looks simpler than the ones in theorem prover with dependent type theory, and make the way of stating theorems in the machine less arguable.

0.3 Why we need modal logic?

Modal logic is a formal system taking about relational structures, and relational structure can be found just about everywhere. It means that we have chance to apply modal logic everywhere. For example, labeled transation systems in theoretical computer since gives a model of modal logic, where the relational sturcture given by using the states as the set, and the relation on the set is given by transations. Moreover, examples of applications of modal logics can be found in many other displines, including but not limited to linguistics, formal semantics, ecnomics, mathematics, and philsophy.

0.4 Why we need the HOL?

Since modal logic are such useful, we will frequently want to apply it to actuall problems in various place. There are several reasons that make it valuable to formalise:

- We want a fully formal discription of our definitions and theorems. And we can hardly find a way rather than checking by a machine that can make us confident to say that 'our definitions and results are formal enough'.
- Although the proofs written on paper is convincing enough for humans, it may comes out that something subtle can prevent us from using the theorems which proved by humans. And there may hidden assumptions about, for instance, well-formedness issues. We want to be fully clear that in which situation can we apply our theorems, and if we are not in 'nice enough' situations, what can we do in order to become able to apply our results. Some examples can be found in our formalisation: A finite type universe is not nice enough for some result to be applied, and we can solve this problem by injecting into an infinite universe.

We hope that the above is enough to convince the reader that we are doing something useful and worthwhile, so let us start looking at what have we done.

(Add the list of functions as tookit in introduction or explain them along the way?)

1 Getting Start

In our formalisation, we only consider the most standard modal language, called the basic modal language. The basic modal language is defined using a set Φ of propositional formulas, and only one modal operator $\Diamond$. A formula in the basic modal language is either a propositional symbol p where $p \in \Phi$, or a disjunction

$\psi \vee \phi$ of two modal formulas ψ and ϕ, or the negation $\neg\phi$ of a modal formula ϕ, or else in the form $\Diamond\phi$ obtained by putting a diamond before a modal formula. In the HOL, we create a datatype called 'form' of the formulas of this modal language.

Definition 1 (Modal formulas as a type)**.**

$$\begin{aligned}
&\alpha\ \mathit{chap1\$form} = \\
&\quad \mathsf{VAR}\ \alpha \\
&\ \mid \mathsf{DISJ}\ (\alpha\ \mathit{chap1\$form})\ (\alpha\ \mathit{chap1\$form}) \\
&\ \mid \bot \\
&\ \mid (\neg)\ (\alpha\ \mathit{chap1\$form}) \\
&\ \mid \Diamond\ (\alpha\ \mathit{chap1\$form})
\end{aligned}$$

When we say an 'α form', we mean a formula in the basic modal language defined on a set Φ with elements of type α, we will just call such a set α-set, an α-set is of type $\alpha \to \mathit{bool}$. For the non-primitive connectives: the conjunction '$\wedge$', the implication $\to$, and the truth '$\top$', they are defined in a standard way:

Definition 2 (Non-primitive connectives)**.**

$$\begin{aligned}
&\mathsf{AND}\ f_1\ f_2 \overset{\mathsf{def}}{=} \neg\mathsf{DISJ}\ (\neg f_1)\ (\neg f_2) \\
&f_1\ \to\ f_2 \overset{\mathsf{def}}{=} \mathsf{DISJ}\ (\neg f_1)\ f_2 \\
&\mathsf{TRUE} \overset{\mathsf{def}}{=} \neg\bot
\end{aligned}$$

As an analogue of the duality between the universal quantifier and the existential quantifier, in the sense that $\exists$ is defined to be $\neg\forall\neg$, we have a modal operator that is dual to the diamond, that is the box $\Box\phi := \neg\Diamond\neg\phi$.

Definition 3 (Box)**.**

$$\Box\ \phi \overset{\mathsf{def}}{=} \neg\Diamond\ (\neg\phi)$$

A modal formula cannot contain any extra ingredient rather then a diamond compared to a propositional formula, so a modal formula without diamond is a propositional formula:

Definition 4 (Propositional formulas)**.**

$$\begin{aligned}
&\mathsf{propform}\ (\mathsf{VAR}\ p) \overset{\mathsf{def}}{=} \mathsf{T} \\
&\mathsf{propform}\ (\mathsf{DISJ}\ \phi_1\ \phi_2) \overset{\mathsf{def}}{=} \mathsf{propform}\ \phi_1\ \wedge\ \mathsf{propform}\ \phi_2 \\
&\mathsf{propform}\ (\neg f) \overset{\mathsf{def}}{=} \mathsf{propform}\ f \\
&\mathsf{propform}\ \bot \overset{\mathsf{def}}{=} \mathsf{T} \\
&\mathsf{propform}\ (\Diamond\ f) \overset{\mathsf{def}}{=} \mathsf{F}
\end{aligned}$$

Knowing how does the modal formulas we are interested in look like, now we want to know the meaning of them. If the modal language is defined using propositional symbols in an α-set Φ. For a propositional formula, it is no more than a combination of propositional formulas using $\vee$ and $\neg$, so its truth value depends entirely on the truth value of the propositional letters in Φ. It follows that we can define the function that evaluate a propositional formula as takes an assignment of truth values of propositional symbols and a modal formula, and gives us the truth value of the formula under the current assignment. In the HOL, to define a function from an α-set, where α is any type, we are not allowed to only assign values to the elements of the set, instead, we must define the function on for all the terms that has type α. Hence our evaluating function peval takes a function $\alpha \to \mathit{bool}$ and an α-form:

Definition 5 (Propositional evaluation)**.**

$$\begin{aligned}
&\mathsf{peval}\ \sigma\ (\mathsf{VAR}\ p) \overset{\text{def}}{=} \sigma\ p\\
&\mathsf{peval}\ \sigma\ (\mathsf{DISJ}\ f_1\ f_2) \overset{\text{def}}{=} \mathsf{peval}\ \sigma\ f_1\ \lor\ \mathsf{peval}\ \sigma\ f_2\\
&\mathsf{peval}\ \sigma\ \bot \overset{\text{def}}{=} \mathsf{F}\\
&\mathsf{peval}\ \sigma\ (\neg f) \overset{\text{def}}{=} \neg\mathsf{peval}\ \sigma\ f\\
&\mathsf{peval}\ \sigma\ (\lozenge\ f) \overset{\text{def}}{=} \mathsf{F}
\end{aligned}$$

For a modal formula that involves diamonds, an assignment of truth value is not enough, since we will need a relational structure to talk about the $\lozenge$. Such a relational structure is called a frame. As pointed out in the introduction, a relational structure consists of a set and a binary relation defined on elements in the set. A relational structure together with an assignment of truth value of propositional letters at each point of the set gives us a model of modal logic. We will call a frame with underlying set is a β-set a β-frame, and a model with β-frame which is used to interpret α-formulas an (α, β)-model.

Definition 6 (Modal model as a type)**.**

$$(\alpha,\ \beta)\ \mathit{chap1\$model} = \texttt{<|}\ \mathit{frame}\ :\ \beta\ \mathit{frame};\ \mathit{valt}\ :\ \alpha\ \to\ \beta\ \to\ \mathit{bool}\ \texttt{|>}$$

When we say an (α, β)-model, we mean a model for α-forms with a β-set as its underlying set.

Now we can interpret modal formulas in the basic modal language on a model by defining satisfication.

Definition 7 (Satisfication)**.**

$$\begin{aligned}
&\mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{VAR}\ p) \overset{\text{def}}{=} w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ w\ \in\ \mathfrak{M}.\mathit{valt}\ p\\
&\mathsf{satis}\ \mathfrak{M}\ w\ \bot \overset{\text{def}}{=} w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ \mathsf{F}\\
&\mathsf{satis}\ \mathfrak{M}\ w\ (\neg\phi) \overset{\text{def}}{=} w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ \neg\mathsf{satis}\ \mathfrak{M}\ w\ \phi\\
&\mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{DISJ}\ \phi_1\ \phi_2) \overset{\text{def}}{=} \mathsf{satis}\ \mathfrak{M}\ w\ \phi_1\ \lor\ \mathsf{satis}\ \mathfrak{M}\ w\ \phi_2\\
&\mathsf{satis}\ \mathfrak{M}\ w\ (\lozenge\ \phi) \overset{\text{def}}{=}\\
&\quad w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ \exists\, v.\ \mathfrak{M}.\mathit{frame.rel}\ w\ v\ \land\ v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ v\ \phi
\end{aligned}$$

(insert an example of model and satisfication here)

Just as in first order logic and propositional logic, we can define the notion of logical equivalence of modal formulas. This gives an equivalence relation between modal formulas of the same type. It does not make sense to talk about logic equivalence of modal formulas in different types.

Definition 8 (Equivalence)**.**

$$\mathsf{equiv0}\ \mathit{tyi}\ f_1\ f_2 \overset{\text{def}}{=} \forall\,\mathfrak{M}\ w.\ \mathsf{satis}\ \mathfrak{M}\ w\ f_1\ \iff\ \mathsf{satis}\ \mathfrak{M}\ w\ f_2$$

Proposition 1 (`equiv0_equiv_on`).

$$\vdash \mathsf{equiv0}\ \mu\ \mathsf{equiv\_on}\ s$$

A notable thing is that $\mathsf{equiv0}$ need to take a type itself, denoted as μ, and if μ denotes the type α itself and f_1, f_2 are β formulas, $\mathsf{equiv0}\ \mu\ f_1\ f_2$ means for any (α, β)-model $\mathfrak{M}$ and any world w in it, we have $\mathsf{satis}\ \mathfrak{M}\ w\ f_1 \iff \mathsf{satis}\ \mathfrak{M}\ w\ f_2$. We are not allowed to omit the μ in the definition, since then there will be a type, namely the type of models $\mathfrak{M}$, that only appears on the right hand side but not on the left hand side of the definition, which is not allowed in the HOL. Also we are not allowed to quantify over the μ, and something like 'equiv f1 f2 $\Leftrightarrow$!μ. equiv0 μ f1 f2' is also not valid. Hence this definition is not encoding the equivalence in mathematical sense precisely, since when we mention equivalence of formulas in usual mathematical language, we are implicitly talking about the class of all models, but the constrain here bans us from talking about all models at once.

We can immediately prove that for μ denotes any type, if $\mathsf{equiv0}\ \mu\ f\ g$ then $\mathsf{equiv0}\ \mu\ (\Diamond\ f)\ (\Diamond\ g)$. The converse holds in the usual sense, but requires extra assumptions to prove in the HOL. For a proof using the mathematical notion of equivalence: If two diamond formulas $\Diamond\ f$ and $\Diamond\ g$ are equivalent, then for a contradiction, suppose that f and g are not equivalent, then there exists a model $\mathfrak{M}$ and a world w such that w satisfies f but not g. We can add a world v to the world set of $\mathfrak{M}$ that is only linked to w, then v will be a witness of the fact that $\Diamond\ f$ and $\Diamond\ g$ are not equivalent. But under our definition in the HOL, if the μ denotes a finite type, the proof is blocked: since we cannot make sure that we can come up with a world v which is not already being used by $\mathfrak{M}$, and add a fresh world to add to $\mathfrak{M}$ which is only linked to w. For a finite type μ, it is possible that we have $\mathsf{equiv0}\ \mu\ (\Diamond\ f)\ (\Diamond\ g)$ but $\neg\mathsf{equiv0}\ \mu\ f\ g$. For example, consider the type that has only two inhabitants a, b, then in the model below, we have $\mathsf{equiv0}\ \mu\ (\Diamond\ (\mathsf{VAR}\ p))\ (\Diamond\ (\mathsf{VAR}\ q))$ but $\neg\mathsf{equiv0}\ \mu\ (\mathsf{VAR}\ p)\ (\mathsf{VAR}\ q)$.

(add the picture of example, how to draw it?)

As the situition above is only because of our special definition, such case is uninteresting. For any model, regardless its world set is of a finite type or not, we can always create a copy of the model in an infinite type. So it is harmless to only play with equivalence of formulas for models whose underlying set is of an infinite type. When μ denotes an infinite type, as we can always come up with a fresh world that allows the proof of equivalence between f and g from the equivalence of $\Diamond f$ and $\Diamond g$ as in the usual sense to go through, we do have a double implication:

Proposition 2 (`equiv0_DIAM`).

$$\vdash \mathsf{INFINITE}\,\mathcal{U}(:\beta)\ \Rightarrow\ (\mathsf{equiv0}\ \mu\ (\Diamond\ f)\ (\Diamond\ g) \iff \mathsf{equiv0}\ \mu\ f\ g)$$

Given a modal formula, we can extract the set of propositional letters appear in it, as:

Definition 9 (Propositional letters).

$$\begin{aligned}
&\mathsf{prop\_letters}\ (\mathsf{VAR}\ p) \overset{\text{def}}{=} \{\ p\ \} \\
&\mathsf{prop\_letters}\ \bot \overset{\text{def}}{=} \emptyset \\
&\mathsf{prop\_letters}\ (\mathsf{DISJ}\ f_1\ f_2) \overset{\text{def}}{=} \mathsf{prop\_letters}\ f_1 \cup \mathsf{prop\_letters}\ f_2 \\
&\mathsf{prop\_letters}\ (\neg f) \overset{\text{def}}{=} \mathsf{prop\_letters}\ f \\
&\mathsf{prop\_letters}\ (\Diamond\ f) \overset{\text{def}}{=} \mathsf{prop\_letters}\ f
\end{aligned}$$

Then we can show when evaluating a formula ϕ on a model, the only relevant information in the valuation are the values on the propositional letters actually occurring in ϕ:

Proposition 3 (`exercise_1_3_1`).

$$\begin{aligned}
&\vdash \mathfrak{M}_1.\mathit{frame} = \mathfrak{M}_2.\mathit{frame} \Rightarrow \\
&\quad (\forall\, p.\ p \in \mathsf{prop\_letters}\ \mathit{phi} \Rightarrow \mathfrak{M}_1.\mathit{valt}\ p = \mathfrak{M}_2.\mathit{valt}\ p) \Rightarrow \\
&\quad \forall\, w.\ w \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}_1\ w\ \mathit{phi} \iff \mathsf{satis}\ \mathfrak{M}_2\ w\ \mathit{phi})
\end{aligned}$$

In particular, if we are interpreting a propositional formula, then we do not need the relation on the model, as we can prove from definition:

Proposition 4 (`peval_satis`).

$$\begin{aligned}
&\vdash \mathsf{propform}\ f \land w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \\
&\quad (\mathsf{satis}\ \mathfrak{M}\ w\ f \iff \mathsf{peval}\ (\lambda\, a.\ w \in \mathfrak{M}.\mathit{valt}\ a)\ f)
\end{aligned}$$

Moreover, we only need the truth values of the propositional letters that appears in the formula:

Proposition 5 (`peval_satis_strengthen'`).

$$\begin{aligned}
&\vdash \mathsf{propform}\ f \land \mathsf{prop\_letters}\ f \subseteq s \land w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \\
&\quad (\mathsf{satis}\ \mathfrak{M}\ w\ f \iff \mathsf{peval}\ ((\lambda\, a.\ w \in \mathfrak{M}.\mathit{valt}\ a) \cap s)\ f)
\end{aligned}$$

One may confused why are we allowed to write $(\lambda\, a.\ w \in \mathfrak{M}.\mathsf{valt}\ a) \cap s$ here, for $\lambda\, a.\ w \in \mathfrak{M}.\mathsf{valt}\ a$ is a function but s is a set. This is because both the α-set s and the function $\lambda\ a.(w \in M.valt)$ has type $\alpha \to \mathit{bool}$, so they can be feeded to the function $\cap$, which has type $(\alpha \to \mathit{bool}) \to (\alpha \to \mathit{bool}) \to (\alpha \to \mathit{bool})$. A important thing to note is that for the set s, we can either view it as a collection of elements of type α, or a function that assigns each term of type α a truth value, where the truth value assigned to $a : \alpha$ is T if and only if $a \in s$. This viewpoint will become important in Interlude I.

(change subforms to propsyms later?)

2 Invariant results and bisimulations

The key concept we are interested in this chapter is called 'modal equivalent'. For a $w \in \mathfrak{M}.\mathsf{frame.world}$, the τ-theory of w is the set of modal formulas satisfied at w. Two worlds $w \in \mathfrak{M}.\mathsf{frame.world}$ and $w' \in \mathfrak{M}'.\mathsf{frame.world}$ are modal equivalent if they have the same τ-theory. We will just refer to 'satisfies the same modal formulas' rather than use the term τ-theory everywhere as a 'working definition':

Definition 10 (τ-theory)**.**

$$\mathsf{tau\_theory}\ \mathfrak{M}\ w \ \overset{\text{def}}{=}\ \{\ \phi\ /\ \mathsf{satis}\ \mathfrak{M}\ w\ \phi\ \}$$

Definition 11 (Modal equivalence)**.**

$$\mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w' \ \overset{\text{def}}{=}\ \mathsf{tau\_theory}\ \mathfrak{M}\ w\ =\ \mathsf{tau\_theory}\ \mathfrak{M}'\ w'$$

Proposition 6 (`modal_eq_tau`)**.**

$$\vdash \mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w' \iff \forall\,\phi.\ \mathsf{satis}\ \mathfrak{M}\ w\ \phi \iff \mathsf{satis}\ \mathfrak{M}'\ w'\ \phi$$

In this chapter, we investigate when can we get modal equivalence. In the first section, we talk about how can we get new models from old models without affecting modal satisfication. Then in the second section, we introduce morphisms between models and find out under which condition we can map a point of a model to another model preserving satisfication. In the last section, we unify all the modal operations and morphisms we have mentioned which gives modal equivalence using a relation between modals, called bisimulation, by proving all these constructions are cases of bisimulation.

2.1 Operations on models

Here we introduce two operations on models, called disjoint union and generated submodels respectively.

The operation on models that leads to invarience of formulas most straightforwardly is the disjoint union:

Definition 12 (Disjoint union)**.**

```
$\mathsf{DU}\ (f, dom)\ \overset{\text{def}}{=}$
  $\texttt{<|}\mathit{frame}\ :=$
    $\texttt{<|}\mathit{world}\ :=\ \{\ w\ /\ \mathsf{FST}\ w\ \in\ dom\ \wedge\ \mathsf{SND}\ w\ \in\ (f\ (\mathsf{FST}\ w)).\mathit{frame.world}\ \}\ ;$
     $\mathit{rel}\ :=$
      $(\lambda\, w_1\ w_2.$
          $\mathsf{FST}\ w_1\ =\ \mathsf{FST}\ w_2\ \wedge\ \mathsf{FST}\ w_1\ \in\ dom\ \wedge$
          $(f\ (\mathsf{FST}\ w_1)).\mathit{frame.rel}\ (\mathsf{SND}\ w_1)\ (\mathsf{SND}\ w_2))\ \texttt{|>};$
  $\mathit{valt}\ :=\ (\lambda\, v\ w.\ (f\ (\mathsf{FST}\ w)).\mathit{valt}\ v\ (\mathsf{SND}\ w))\ \texttt{|>}$
```

Disjoint union is denoted as DU, it is a function that takes a pair (f, dom). As we do not allow dependent type theory, we can only take disjoint union of modals of the same type. If we are taking disjoint union of (α, β)-models indexed by a γ-set dom, then $f : \gamma \to (\alpha, \beta)$-model will assign each index an (α, β)-model. The f must be defined on all the terms of type γ, but all we need is its value on dom. The world set of the disjoint union of such a family has elements of form (i, w) where $i \in dom$ and $w\ \in\ (f\ i).\mathsf{frame.world}$. The valuation on $\mathsf{DU}\ (f, dom)$ is given by $(\mathsf{DU}\ (f, dom)).\mathsf{valt}\ p\ (i, w)$ iff $(f\ i).\mathsf{valt}\ p\ w$, for any propositonal letter p and any world (i, w). Relation is defined by $(\mathsf{DU}\ (f, dom)).\mathsf{frame.rel}\ (i, w)\ (i', w')$ iff $i\ =\ i'$ and $(f\ i).\mathsf{frame.rel}\ w\ w'$, means that (i, w) and (i', w') come from the same model and are related by the relation in that model.

The disjoint union does nothing except for collecting a set of models together, so it is unsurprising that we can prove by induction that a modal formulad is satisfied at $w \in \mathfrak{M}.\mathsf{frame.world}$ iff it is satisfied in the copy of the $\mathfrak{M}$ at the same point when $\mathfrak{M}$ is embeded into a disjoint union:

Proposition 7 (`prop_2_3`)**.**

$$\vdash \mathsf{FST}\ w \in dom \Rightarrow (\mathsf{satis}\ (f\ (\mathsf{FST}\ w))\ (\mathsf{SND}\ w)\ phi \iff \mathsf{satis}\ (\mathsf{DU}\ (f, dom))\ w\ phi)$$

Disjoint union is the operation we use when we want to expand out scope from a smaller model to a large model. Accordingly, we have an operation that allows us to restrict our scope to a smaller model, this is called 'generated submodel' construction.

When we say '$\mathfrak{M}_1$ is a submodel of $\mathfrak{M}_2$', we mean all the information of $\mathfrak{M}_1$ is inherited from that of $\mathfrak{M}_2$:

Definition 13 (Submodel)**.**

$$\begin{aligned}
&\mathsf{SUBMODEL}\ \mathfrak{M}_1\ \mathfrak{M}_2 \overset{\mathsf{def}}{=} \\
&\ \mathfrak{M}_1.\mathit{frame.world} \subseteq \mathfrak{M}_2.\mathit{frame.world} \land \\
&\ \forall w_1. \\
&\quad w_1 \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow \\
&\quad (\forall v.\ \mathfrak{M}_1.\mathit{valt}\ v\ w_1 \iff \mathfrak{M}_2.\mathit{valt}\ v\ w_1) \land \\
&\quad \forall w_2.\ w_2 \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow (\mathfrak{M}_1.\mathit{frame.rel}\ w_1\ w_2 \iff \mathfrak{M}_2.\mathit{frame.rel}\ w_1\ w_2)
\end{aligned}$$

It is not necessary that submodel construction preserves modal satisfication. Although the clause about relation says that for any worlds w_1, w_2 in $\mathfrak{M}_1$, they are related in $\mathfrak{M}_1$ iff they are related in $\mathfrak{M}_2$, for linked worlds w_1, w_2 in $\mathfrak{M}_2$, we are allowed to include w_1 into the world set of $\mathfrak{M}_1$ and discard w_2. Hence the satisfication of formula with diamonds is not preserved. If we want to preserve the diamond formulas, we can simply add an extra constrain to ensure that we can never have this situation. A submodel which satisfies the extra condition is called a generated submodel:

Definition 14 (Generated submodel)**.**

$$\begin{aligned}
&\vdash \mathsf{GENSUBMODEL}\ \mathfrak{M}_1\ \mathfrak{M}_2 \iff \\
&\quad \mathsf{SUBMODEL}\ \mathfrak{M}_1\ \mathfrak{M}_2 \land \\
&\quad\ \forall w_1. \\
&\qquad w_1 \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow \\
&\qquad \forall w_2.\ w_2 \in \mathfrak{M}_2.\mathit{frame.world} \land \mathfrak{M}_2.\mathit{frame.rel}\ w_1\ w_2 \Rightarrow w_2 \in \mathfrak{M}_1.\mathit{frame.world}
\end{aligned}$$

Note that for a model $\mathfrak{M}_1$ to be a generated submodel of $\mathfrak{M}_2$, we only require all the worlds w' which have a link from a world w in M_1 to be also included in the world set of M_1, and if w_1, w_2 are in $\mathfrak{M}_2$ and $w_2 \in \mathfrak{M}_1.\mathsf{frame.world}$ we are allowed not to include w_1 into $\mathfrak{M}_1$. This is because from its definition, the diamond in modal formulas cannot 'look back', in the sense that adding an extra connection or discard connections to a world w does not change the satisfication of diamond formulas at w.

A rooted model is a submodel generated by a point. We will talk rooted models more frequently later, so they deserve a separate definition. As a special kind of generated model, a rooted model needs to be sitting in an ambient model. This make it reasonable to define rooted model as a predicate that takes three parameters: The model itself, the point that generates it, and the ambient model that it is sitting in:

Definition 15 (Rooted model)**.**

$$\begin{aligned}
&\mathsf{rooted\_model}\ \mathfrak{M}\ x\ \mathfrak{M}' \overset{\text{def}}{=} \\
&\quad x \in \mathfrak{M}'.\mathit{frame.world} \land \\
&\quad (\forall a. \\
&\qquad a \in \mathfrak{M}.\mathit{frame.world} \iff \\
&\qquad\quad a \in \mathfrak{M}'.\mathit{frame.world} \land (\mathsf{RESTRICT}\ \mathfrak{M}'.\mathit{frame.rel}\ \mathfrak{M}'.\mathit{frame.world})^*\ x\ a) \land \\
&\quad (\forall n_1\ n_2. \\
&\qquad n_1 \in \mathfrak{M}.\mathit{frame.world} \land n_2 \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \\
&\qquad (\mathfrak{M}.\mathit{frame.rel}\ n_1\ n_2 \iff \mathsf{RESTRICT}\ \mathfrak{M}'.\mathit{frame.rel}\ \mathfrak{M}'.\mathit{frame.world}\ n_1\ n_2)) \land \\
&\quad \forall v\ n.\ \mathfrak{M}.\mathit{valt}\ v\ n \iff \mathfrak{M}'.\mathit{valt}\ v\ n
\end{aligned}$$

Generated submodels do preserve modal satisfication:

Proposition 8 (`prop_2_6`)**.**

$$\vdash \mathsf{GENSUBMODEL}\ \mathfrak{M}_1\ \mathfrak{M}_2 \land n \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}_1\ n\ \mathit{phi} \iff \mathsf{satis}\ \mathfrak{M}_2\ n\ \mathit{phi})$$

2.2 Morphisms between models

In this section, we talk about various kinds of 'morphisms' between models. Similar as in mathematics, the notion of 'morphism' here is used to describe maps that preserves structures. For instance, a homomorphism is a weakest notion of 'structure-preserving':

Definition 16 (Homomorphism)**.**

$$\begin{aligned}
&\mathsf{hom}\ f\ \mathfrak{M}_1\ \mathfrak{M}_2 \overset{\text{def}}{=} \\
&\quad \forall w. \\
&\qquad w \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow \\
&\qquad f\ w \in \mathfrak{M}_2.\mathit{frame.world} \land (\forall p.\ w \in \mathfrak{M}_1.\mathit{valt}\ p \Rightarrow f\ w \in \mathfrak{M}_2.\mathit{valt}\ p) \land \\
&\qquad \forall u.\ u \in \mathfrak{M}_1.\mathit{frame.world} \Rightarrow \mathfrak{M}_1.\mathit{frame.rel}\ w\ u \Rightarrow \mathfrak{M}_2.\mathit{frame.rel}\ (f\ w)\ (f\ u)
\end{aligned}$$

The last clause says 'relation in the source model is preserved by a homomorphism'. And we allow the existence of links in the target model which has no counterpart in the source. Because of this, we cannot guarentee any world and its image in the target to satisfy exactly the same set of modal formulas. As an attempt to obtain an equivalence, we can try strengthening the last clause to be a 'two-side' correspondence of links in the source and target. A homomorphism satisfies this strengthened condition is called a strong homomorphism:

Definition 17 (Strong homomorphism)**.**

$$
\begin{array}{l}
\mathsf{strong\_hom}\ f\ \mathfrak{M}_1\ \mathfrak{M}_2 \ \stackrel{\mathsf{def}}{=} \\
\quad \forall\, w. \\
\qquad w \ \in\ \mathfrak{M}_1.\mathit{frame.world} \ \Rightarrow \\
\qquad f\ w \ \in\ \mathfrak{M}_2.\mathit{frame.world} \ \wedge\ (\forall\, p.\ w \ \in\ \mathfrak{M}_1.\mathit{valt}\ p \ \iff\ f\ w \ \in\ \mathfrak{M}_2.\mathit{valt}\ p) \ \wedge \\
\qquad \forall\, u.\ u \ \in\ \mathfrak{M}_1.\mathit{frame.world} \ \Rightarrow\ (\mathfrak{M}_1.\mathit{frame.rel}\ w\ u \ \iff\ \mathfrak{M}_2.\mathit{frame.rel}\ (f\ w)\ (f\ u))
\end{array}
$$

A strong homomorphism that is a bijection on the world set is called an isomorphism. An isomorphism will certainly preserves everthing, including modal equivalence, but it is uninteresting, what we want is an answer of this question: Do we really require isomorphisms for the sake of modal equivalence? The answer is no, something weaker than an isomorphism is enough. The thing we need is just a surjective strong morphism:

Proposition 9 (`prop_2_9`)**.**

$$
\begin{array}{l}
\vdash \mathsf{strong\_hom}\ f\ \mathfrak{M}\ \mathfrak{M}' \ \wedge\ f\ w \ =\ w' \ \wedge\ w \ \in\ \mathfrak{M}.\mathit{frame.world} \ \wedge \\
\quad \mathsf{SURJ}\ f\ \mathfrak{M}.\mathit{frame.world}\ \mathfrak{M}'.\mathit{frame.world} \ \Rightarrow \\
\quad \mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'
\end{array}
$$

Now we have a desired modal equivalence from strong homomorphism. The problem is that the condition of being a strong homomorphism is still too strong. There are many morphisms which preserves modal satisfication which fails satisfying the condition required by a surjective strong homomorphism, so it is still not the minimum. We still want something weaker. Finally, we find a satisfactory answer-the bounded morphism defined as the follows:

Definition 18 (Bounded morphism)**.**

$$
\begin{array}{l}
\mathsf{bounded\_mor}\ f\ \mathfrak{M}\ \mathfrak{M}' \ \stackrel{\mathsf{def}}{=} \\
\quad \forall\, w. \\
\qquad w \ \in\ \mathfrak{M}.\mathit{frame.world} \ \Rightarrow \\
\qquad f\ w \ \in\ \mathfrak{M}'.\mathit{frame.world} \ \wedge\ (\forall\, a.\ \mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{VAR}\ a) \ \iff\ \mathsf{satis}\ \mathfrak{M}'\ (f\ w)\ (\mathsf{VAR}\ a)) \ \wedge \\
\qquad (\forall\, v.\ v \ \in\ \mathfrak{M}.\mathit{frame.world} \ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ w\ v \ \Rightarrow\ \mathfrak{M}'.\mathit{frame.rel}\ (f\ w)\ (f\ v)) \ \wedge \\
\qquad \forall\, v'. \\
\qquad\quad v' \ \in\ \mathfrak{M}'.\mathit{frame.world} \ \wedge\ \mathfrak{M}'.\mathit{frame.rel}\ (f\ w)\ v' \ \Rightarrow \\
\qquad\quad \exists\, v.\ v \ \in\ \mathfrak{M}.\mathit{frame.world} \ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ w\ v \ \wedge\ f\ v \ =\ v'
\end{array}
$$

The invariance result that bounded morphism gives is stated as:

Proposition 10 (`prop_2_14`)**.**

$$
\vdash \mathsf{bounded\_mor}\ f\ \mathfrak{M}\ \mathfrak{M}' \ \wedge\ w \ \in\ \mathfrak{M}.\mathit{frame.world} \ \Rightarrow\ (\mathsf{satis}\ \mathfrak{M}\ w\ \phi \ \iff\ \mathsf{satis}\ \mathfrak{M}'\ (f\ w)\ \phi)
$$

It turns out to be very useful. As an application, we will use it to prove the tree-like property of our modal language. The tree-like property says that for any formula ϕ satisfied on any point in any model, there exists a tree-like model such that ϕ is satisfied at the root of the tree. As the name indicates, a tree-like model is a model such that its underlying frame is a tree. Being a tree is a property for relational structures. The predicate tree takes a frame S and a point r, and tree S_0 r iff S_0 is a tree with root r:

Definition 19 (Tree)**.**

$$\begin{aligned} &\mathsf{tree}\ S\ r \overset{\mathsf{def}}{=} \\ &\quad r \in S.\mathit{world} \wedge (\forall t.\ t \in S.\mathit{world} \Rightarrow (\mathsf{RESTRICT}\ S.\mathit{rel}\ S.\mathit{world})^{*}\ r\ t) \wedge \\ &\quad (\forall r_0.\ r_0 \in S.\mathit{world} \Rightarrow \neg S.\mathit{rel}\ r_0\ r) \wedge \\ &\quad \forall t.\ t \in S.\mathit{world} \wedge t \neq r \Rightarrow \exists! t_0.\ t_0 \in S.\mathit{world} \wedge S.\mathit{rel}\ t_0\ t \end{aligned}$$

Here the superscript $*$ is used to take the reflective and transitive closure of a relation,RESTRICT is a function that takes a relation R and a set s and return the relation such that RESTRICT $R\ s\ x\ y$ iff $x \in s$, $y \in s$ and $R\ x\ y$:

Definition 20 (Restriction of relation)**.**

$$\mathsf{RESTRICT}\ R\ s\ x\ y \overset{\mathsf{def}}{=} R\ x\ y \wedge x \in s \wedge y \in s$$

By directly checking the definitions, any tree like model is rooted:

Proposition 11 (`tree_like_model_rooted`)**.**

$$\vdash \mathsf{tree}\ \mathfrak{M}.\mathit{frame}\ r \Rightarrow \mathsf{rooted\_model}\ \mathfrak{M}\ r\ \mathfrak{M}$$

An induction on reflextive and transitive closure proves a tree has no loop:

Lemma 1 (`tree_no_loop`)**.**

$$\vdash \mathsf{tree}\ s\ r \Rightarrow \forall t_0\ t.\ (\mathsf{RESTRICT}\ s.\mathit{rel}\ s.\mathit{world})^{+}\ t_0\ t \Rightarrow t_0 \neq t$$

We now prove the tree-like property of modal formulas:

Proposition 12 (`prop_2_15_corollary`)**.**

$$\vdash \mathsf{satis}\ \mathfrak{M}\ w\ \phi \Rightarrow \exists\, \mathit{MODEL}\ s.\ \mathsf{tree}\ \mathit{MODEL}.\mathit{frame}\ s \wedge \mathsf{satis}\ \mathit{MODEL}\ s\ \phi$$

Proof. Suppose satis $\mathfrak{M}$ w phi. By the invariance result of rooted model, we have satis $\mathfrak{M}'$ w phi where $\mathfrak{M}'$ is the rooted model generated by w. In our construction, both generated submodel and rooted model are predicates that judge if a model is a generated submodel or a rooted model. But here we require the existence of a model, hence to use rooted model here, we need a function that gives us one, and prove it is rooted and generated submodel in order to use previous theorems.

$$
\begin{array}{l}
\mathsf{point\_GENSUBMODEL}\ \mathfrak{M}\ w \overset{\mathsf{def}}{=} \\
\quad \mathsf{<|frame} := \\
\quad\quad \mathsf{<|world} := \\
\quad\quad\quad \{\, v \mid v \in \mathfrak{M}\mathsf{.frame.world} \wedge (\mathsf{RESTRICT}\ \mathfrak{M}\mathsf{.frame.rel}\ \mathfrak{M}\mathsf{.frame.world})^{*}\ w\ v \,\}\,; \\
\quad\quad\ \mathsf{rel} := \\
\quad\quad\quad (\lambda\, w_1\ w_2.\ w_1 \in \mathfrak{M}\mathsf{.frame.world} \wedge w_2 \in \mathfrak{M}\mathsf{.frame.world} \wedge \mathfrak{M}\mathsf{.frame.rel}\ w_1\ w_2)\mathsf{|>}; \\
\quad \mathsf{valt} := \mathfrak{M}\mathsf{.valt|>}
\end{array}
$$

$$\vdash w \in \mathfrak{M}\mathsf{.frame.world} \Rightarrow \mathsf{GENSUBMODEL}\ (\mathsf{point\_GENSUBMODEL}\ \mathfrak{M}\ w)\ \mathfrak{M}$$

$$\vdash w \in \mathfrak{M}\mathsf{.frame.world} \Rightarrow \mathsf{rooted\_model}\ (\mathsf{point\_GENSUBMODEL}\ \mathfrak{M}\ w)\ w\ \mathfrak{M}$$

The $\mathfrak{M}'$ is taken as $\mathsf{point\_GENSUBMODEL}\ \mathfrak{M}\ w$. By the `prop_2_14`, it suffices to prove $\mathfrak{M}'$ is the image of some bounded morphism from some tree-like model M'' where the root of the tree is mapped to w. We construct M'' as follows: Take the set of worlds to be the finite sequences $[w = u_0; u_1; \cdots; u_n]$ such that $\mathfrak{M}\mathsf{.frame.rel}\ u_i\ u_{i+1}$ for all i. Define $M''\mathsf{.frame.rel}\ [w; u_1; \cdots; u_n]\ [w; v_1; \cdots; v_m]$ iff $m = n+1$, $m_i = v_i$ for $i \leq n$, and $M.frame.rel\ u_n\ v_m$. The valuation is given by $[w; u_1; \cdots; u_n] \in M''.valt\ p$ iff $u_n \in M.valt\ p$. In the HOL, such a model looks like:

$$
\begin{array}{l}
\mathsf{bounded\_preimage\_rooted}\ \mathfrak{M}\ x \overset{\mathsf{def}}{=} \\
\quad \mathsf{<|frame} := \\
\quad\quad \mathsf{<|world} := \\
\quad\quad\quad \{\, l \mid \\
\quad\quad\quad\ \mathsf{HD}\ l = x \wedge \mathsf{LENGTH}\ l > 0 \wedge \\
\quad\quad\quad\ \forall\, m. \\
\quad\quad\quad\quad m < \mathsf{LENGTH}\ l - 1 \Rightarrow \\
\quad\quad\quad\quad \mathsf{RESTRICT}\ \mathfrak{M}\mathsf{.frame.rel}\ \mathfrak{M}\mathsf{.frame.world}\ (\mathsf{EL}\ m\ l)\ (\mathsf{EL}\ (m + 1)\ l)\,\}\,; \\
\quad\quad\ \mathsf{rel} := \\
\quad\quad\quad (\lambda\, l_1\ l_2. \\
\quad\quad\quad\quad \mathsf{LENGTH}\ l_1 + 1 = \mathsf{LENGTH}\ l_2 \wedge \\
\quad\quad\quad\quad \mathsf{RESTRICT}\ \mathfrak{M}\mathsf{.frame.rel}\ \mathfrak{M}\mathsf{.frame.world}\ (\mathsf{LAST}\ l_1)\ (\mathsf{LAST}\ l_2) \wedge \\
\quad\quad\quad\quad \forall\, m.\ m < \mathsf{LENGTH}\ l_1 \Rightarrow \mathsf{EL}\ m\ l_1 = \mathsf{EL}\ m\ l_2)\mathsf{|>}; \\
\quad \mathsf{valt} := (\lambda\, v\ n.\ \mathfrak{M}\mathsf{.valt}\ v\ (\mathsf{LAST}\ n))\mathsf{|>}
\end{array}
$$

It is straightforward to check the map LAST that sends a list in $l \in (\mathsf{bounded\_preimage\_rooted}\ \mathfrak{M}'\ w)\mathsf{.frame.world}$ to its last member is a bounded morphism, and $[w]$ in M'' is sent to w in $\mathfrak{M}'$, as desired.

□

2.3 Bisimulation

The three approachs to obtain modal equivalence has a common feature: all of them leads to a relation between models such that related states satisfies exactly the same set of propositional letters, and once we are able to make a transition in one model, we can make a corresponding transition in the other. This observation leads us to the concept of bisimulation:

Definition 21 (Bisimulation)**.**

$$
\begin{aligned}
&\mathsf{bisim}\ Z\ \mathfrak{M}\ \mathfrak{M}' \overset{\text{def}}{=} \\
&\quad \forall\, w\ w'.\\
&\qquad w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ w'\ \in\ \mathfrak{M}'.\mathit{frame.world}\ \wedge\ Z\ w\ w'\ \Rightarrow\\
&\qquad (\forall\, a.\ \mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{VAR}\ a)\ \iff\ \mathsf{satis}\ \mathfrak{M}'\ w'\ (\mathsf{VAR}\ a))\ \wedge\\
&\qquad (\forall\, v.\\
&\qquad\quad v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ w\ v\ \Rightarrow\\
&\qquad\quad \exists\, v'.\ v'\ \in\ \mathfrak{M}'.\mathit{frame.world}\ \wedge\ Z\ v\ v'\ \wedge\ \mathfrak{M}'.\mathit{frame.rel}\ w'\ v')\ \wedge\\
&\qquad \forall\, v'.\\
&\qquad\quad v'\ \in\ \mathfrak{M}'.\mathit{frame.world}\ \wedge\ \mathfrak{M}'.\mathit{frame.rel}\ w'\ v'\ \Rightarrow\\
&\qquad\quad \exists\, v.\ v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ Z\ v\ v'\ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ w\ v
\end{aligned}
$$

If we have $\mathsf{bisim}\ Z\ \mathfrak{M}\ \mathfrak{M}'$, it means that Z is a bisimulation relation between worlds in $\mathfrak{M}$ and $\mathfrak{M}'$. We require any two worlds related by a bisimulation to have same atomic information and matching transition possibilities. We can see all of the three constructions we introduced before give arise to bisimulations:

Proposition 13 (`prop_2_19_ii`)**.**

$$\vdash i\ \in\ \mathit{dom}\ \wedge\ w\ \in\ (f\ i).\mathit{frame.world}\ \Rightarrow\ \mathsf{bisim\_world}\ (f\ i)\ (\mathsf{DU}\ (f, \mathit{dom}))\ w\ (i, w)$$

Proposition 14 (`prop_2_19_ii`)**.**

$$\vdash \mathsf{GENSUBMODEL}\ \mathfrak{M}\ \mathfrak{M}'\ \Rightarrow\ \forall\, w.\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\ \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w$$

Proposition 15 (`prop_2_19_iv`)**.**

$$\vdash \mathsf{bounded\_mor\_image}\ f\ \mathfrak{M}\ \mathfrak{M}'\ \Rightarrow\ \forall\, w.\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\ \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ (f\ w)$$

Proof: For (i), the bisimulation relation is $\lambda\, a\ b.\ b\ =\ (i, a)$, which is linking a world to its copy in the disjoint union. For (ii), the relation is $\lambda\, n_1\ n_2.\ n_1\ =\ n_2$. And for (iii), the relation is $\lambda\, n_1\ n_2.\ n_2\ =\ f\ n_1$.

The clauses on forth and back condition for a bisimulation relation provide precisely the information to push the induction through when proving worlds related by a bisimulation satisfies the same set of modal formulas:

Theorem 1 (`thm_2_20`)**.**

$$\vdash \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'\ \Rightarrow\ \mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'$$

The theorem above provides alternative proofs to the invariance theorems for disjoint union, generated submodels, and bounded morphisms. We can firstly prove the constructions give arise to bisimulations, and then specialise `thm_2_20`. But it is not the end of the story. A natural question to ask is : is bisimulation and modal equivalence the 'same thing'? More precisely, a bisimulation will always give a modal equivalence, conversely, is that the fact that a modal equivalence always give a bisimulation?

The answer is no. Nonetheless, we can prove the converse of the theorem above with an extra condition on the models. A model $\mathfrak{M}$ is called image finite if for any world $w \in \mathfrak{M}.\mathsf{frame.world}$, there are only finitely many worlds in $\mathfrak{M}$ related to w.

Definition 22 (Image finite)**.**

$$\begin{aligned}&\mathsf{image\_finite}\ \mathfrak{M} \overset{\text{def}}{=}\\ &\quad \forall x.\ x \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \mathsf{FINITE}\ \{\ y\ /\ y \in \mathfrak{M}.\mathit{frame.world} \land \mathfrak{M}.\mathit{frame.rel}\ x\ y\ \}\end{aligned}$$

Our main theorem is called Hennessy-Milner theorem, it says that for image finite models, modal equivalence and bisimulation are indeed the same thing:

Theorem 2 (`thm_2_24`: Hennessy-Milner Theorem)**.**

$$\begin{aligned}&\vdash \mathsf{image\_finite}\ \mathfrak{M} \land \mathsf{image\_finite}\ \mathfrak{M}' \land w \in \mathfrak{M}.\mathit{frame.world} \land w' \in \mathfrak{M}'.\mathit{frame.world} \Rightarrow\\ &\quad (\mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w' \iff \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w')\end{aligned}$$

Proof. The implication from right to left is no more than `thm_2_20`. We prove the implication from left to right. Given w and w' are worlds in $\mathfrak{M}$ and $\mathfrak{M}'$ which are modal equivalent, we prove the relation $\lambda\, n_1\ n_2.\ \forall \phi.\ \mathsf{satis}\ \mathfrak{M}\ n_1\ \phi \iff \mathsf{satis}\ \mathfrak{M}'\ n_2\ \phi$ gives a bisimulation. The first clause is immediate to check. For the second one, assume $\mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ n_1\ n_2$ and $\mathfrak{M}.\mathsf{frame.rel}\ n_1\ n_1'$ for some $n_1' \in \mathfrak{M}.\mathsf{frame.world}$, we prove the existence of the world $n_2' \in \mathfrak{M}'.\mathsf{frame.world}$ such that $\mathfrak{M}'.\mathsf{frame.rel}\ n_2\ n_2'$ and $\mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ n_1'\ n_2'$. Suppose such a n_2' does not exist, we derive a contradiction. Consider the set $S_0 = \{\ u' \mid u' \in \mathfrak{M}'.\mathsf{frame.world} \land \mathfrak{M}'.\mathsf{frame.rel}\ n_2\ u'\ \}$, the first claim is that S_0 is finite and nonempty. Finiteness comes from the fact that $\mathfrak{M}'$ is image finite, and if the set is empty, then $\Box \perp$ will be a formula satisfied at n_2 but not at n_1, contradicts the modal equivalence between n_1 and n_2. By assumption, for each world in S_0, there is a formula phi such that $\mathsf{satis}\ \mathfrak{M}\ n_1'\ phi$ but $\neg\mathsf{satis}\ \mathfrak{M}'\ n_2'\ phi$. As the set S is finite, the set of such phis is finite. Then we can take the conjunction of such phis to obtain a formula psi. Then we will have $\mathsf{satis}\ \mathfrak{M}\ n_1\ (\Diamond\ psi)$ but $\neg\mathsf{satis}\ \mathfrak{M}\ n_2\ (\Diamond\ psi)$.

The trick is what to do to capture the big conjunction. Certainly we can define a big conjunction inductively as a function that takes a finite set and give us the formula that conjuncts them together, but even it is clear that we are using the big conjunction, it can be convenient not having to define the details of constructing such a big conjunction, instead, we may directly obtain the formula with exactly the property we want for the proof here. This idea will be used in our formalisation again and again. With this in mind, we prove:

$$\begin{aligned}&\vdash \mathsf{FINITE}\ s \Rightarrow\\ &\quad s \neq \emptyset \Rightarrow\\ &\quad v \in \mathfrak{M}.\mathsf{frame.world} \land (\forall v'.\ v' \in s \Rightarrow \exists phi.\ \mathsf{satis}\ \mathfrak{M}\ v\ phi \land \neg\mathsf{satis}\ \mathfrak{M}'\ v'\ phi) \Rightarrow\\ &\quad \exists psi.\ \mathsf{satis}\ \mathfrak{M}\ v\ psi \land \forall v'.\ v' \in s \Rightarrow \neg\mathsf{satis}\ \mathfrak{M}'\ v'\ psi\end{aligned}$$

Using this lemma, we obtain the psi we want by plugging in S_0 to be the s. □

3 Finite model property

In this chapter, we tell the story about the Slogan 2 as stated in the introduction: Modal formulas can only capture local information, so any modal formula cannot tell anything about a point which is infinitely many steps form the current state. This is done by proving if a modal formula is satisfied on an arbitary model, then it can be satisfied on a finite model, where finite model means the finitness of the set of worlds. We will discuss two methods for building finite models for satisfiable modal formulas, namely via filteration and selection, in the two sections of this chapter.

3.1 Finite model property via filteration

One way to get finite model from an arbitary model is by filteration. Filtration is just another name of the rather familar terminology 'quotient'. We will take a quotient on the world set of the model we start with by identifying states via the usage of equivalence classes. The equivalence relation that we will use for taking the quotient is defined using the concept of 'closed under subformulas'. A subformula of a formula is 'a part of the formula' which is itself a formula. Each modal formula has a set of subformulas, so *subformulas* is a function takes a formula and give this set:

Definition 23 (Subformula)**.**

$$\begin{aligned}
&\mathsf{subforms}\ (\mathsf{VAR}\ a) \overset{\mathsf{def}}{=} \{\ \mathsf{VAR}\ a\ \}\\
&\mathsf{subforms}\ \bot \overset{\mathsf{def}}{=} \{\ \bot\ \}\\
&\mathsf{subforms}\ (\neg f) \overset{\mathsf{def}}{=} \neg f\ \mathsf{INSERT}\ \mathsf{subforms}\ f\\
&\mathsf{subforms}\ (\mathsf{DISJ}\ f_1\ f_2) \overset{\mathsf{def}}{=} \mathsf{DISJ}\ f_1\ f_2\ \mathsf{INSERT}\ \mathsf{subforms}\ f_1\ \cup\ \mathsf{subforms}\ f_2\\
&\mathsf{subforms}\ (\Diamond\ f) \overset{\mathsf{def}}{=} \Diamond\ f\ \mathsf{INSERT}\ \mathsf{subforms}\ f
\end{aligned}$$

Some properties of subformulas can be proved immediately, for instance, any formula is a subformula of itself. Also the relation 'is subformula of' is transitive, and the set of subformulas for any formula is finite.

Proposition 16 (`subforms_phi_phi`)**.**

$$\vdash phi\ \in\ \mathsf{subforms}\ phi$$

Proposition 17 (`subforms_trans`)**.**

$$\vdash f\ \in\ \mathsf{subforms}\ phi\ \land\ phi\ \in\ \mathsf{subforms}\ psi\ \Rightarrow\ f\ \in\ \mathsf{subforms}\ psi$$

Proposition 18 (`subforms_FINITE`)**.**

$$\vdash \mathsf{FINITE}\ (\mathsf{subforms}\ phi)$$

We say a set Σ is closed under formula if for any formula *phi* in the set, any subformula of *phi* is also in Σ.

Definition 24 (Closed under subformulas).

$$\begin{array}{l}\mathsf{CUS}\ \Sigma \overset{\mathsf{def}}{=} \\ \quad \forall f\ f'. \\ \qquad (\mathsf{DISJ}\ f\ f' \in \Sigma \Rightarrow f \in \Sigma \land f' \in \Sigma) \land (\neg f \in \Sigma \Rightarrow f \in \Sigma) \land \\ \qquad (\Diamond f \in \Sigma \Rightarrow f \in \Sigma)\end{array}$$

Clearly, the set of subformulas of any formula is closed under subformulas:

Proposition 19 (`subforms_phi_CUS`).

$$\vdash \mathsf{CUS}\ (\mathsf{subforms}\ phi)$$

For a model $\mathfrak{M}$, the equivalence relation we will use to filtrate its world set is:

Definition 25 (Relation used to define filtration).

$$\begin{array}{l}\mathsf{REL\_CUS}\ \Sigma\ \mathfrak{M} \overset{\mathsf{def}}{=} \\ \quad (\lambda\, w\ v. \\ \qquad w \in \mathfrak{M}.\mathit{frame.world} \land v \in \mathfrak{M}.\mathit{frame.world} \land \\ \qquad \forall\, phi.\ phi \in \Sigma \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ phi \iff \mathsf{satis}\ \mathfrak{M}\ v\ phi))\end{array}$$

Under this equivalence relation, for any world $w \in \mathfrak{M}$.frame.world, $\mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w$ is the equivalence class it belongs to:

Definition 26 (Equivalence class under the relation).

$$\mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w \overset{\mathsf{def}}{=} \{\, v \mid \mathsf{REL\_CUS}\ \Sigma\ \mathfrak{M}\ w\ v \,\}$$

We can directly take these equivalence class to be the set of worlds for the filtrated model, but then the model we get will have different type from the original model. It is very easy to stay in the same type: Just pick one representative from each equivalence class and collect the representatives to be the world set of the new model. We use the choice function CHOICE to select the representatives. And will use $\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w$ as the representative of the equivalence class where w lives in. $\mathsf{EC\_REP\_SET}\ \Sigma\ \mathfrak{M}$ will be the set of worlds for the filtrated model.

Definition 27 (Representatives of an equivalence class).

$$\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w \overset{\mathsf{def}}{=} \mathsf{CHOICE}\ (\mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w)$$

Definition 28 (World set of the filtered model).

$$\mathsf{EC\_REP\_SET}\ \Sigma\ \mathfrak{M} \overset{\mathsf{def}}{=} \{\, n \mid \exists\, w.\ w \in \mathfrak{M}.\mathit{frame.world} \land n = \mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w \,\}$$

The definition of filtration of a model $\mathfrak{M}$ via a subformula-closed set is given by a relation, we read as $\mathsf{filtration}\ \mathfrak{M}\ \Sigma\ FLT$ as 'FLT is a filteration of $\mathfrak{M}$ under Σ.

Definition 29 (Filtration)**.**

$$
\begin{aligned}
&\mathsf{filtration}\ \mathfrak{M}\ \Sigma\ \mathit{FLT} \stackrel{\mathsf{def}}{=} \\
&\quad \mathsf{CUS}\ \Sigma\ \land\ \mathit{FLT}.\mathit{frame}.\mathit{world}\ =\ \mathsf{EC\_REP\_SET}\ \Sigma\ \mathfrak{M}\ \land \\
&\quad (\forall\, w\ v. \\
&\qquad w\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land\ v\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land\ \mathfrak{M}.\mathit{frame}.\mathit{rel}\ w\ v\ \Rightarrow \\
&\qquad \mathit{FLT}.\mathit{frame}.\mathit{rel}\ (\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w)\ (\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ v))\ \land \\
&\quad (\forall\, w\ v. \\
&\qquad w\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land\ v\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land \\
&\qquad \mathit{FLT}.\mathit{frame}.\mathit{rel}\ (\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w)\ (\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ v)\ \Rightarrow \\
&\qquad \forall\, \mathit{phi}\ \mathit{psi}.\ \mathit{phi}\ \in\ \Sigma\ \land\ \mathit{phi}\ =\ \Diamond\ \mathit{psi}\ \Rightarrow\ \mathsf{satis}\ \mathfrak{M}\ v\ \mathit{psi}\ \Rightarrow\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi})\ \land \\
&\quad \forall\, p\ s.\ \mathit{FLT}.\mathit{valt}\ p\ s\ \iff\ \exists\, w.\ s\ =\ \mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{VAR}\ p)
\end{aligned}
$$

The definition above forces filtration to only make sense for subformula closed sets. We do not encode it as a function since we may have many filterations of thesame model using the same subformula-closed set. Indeed, from the above definition, we see once $\mathfrak{M}$ and Σ is fixed, then the world set and valuation of the filtered model will both be fixed, but it is still possible to get filtrated models with differed relations. Since at the moment we just need the existence of filtration, we do not bother with the different relations here, and fix only one relation satisfies the definition for our usage. Our function that takes a model $\mathfrak{M}$ and a set Σ and give a filteration is defined as:

Definition 30 (The filtration model we will use)**.**

$$
\begin{aligned}
&\mathsf{FLT}\ \mathfrak{M}\ \Sigma \stackrel{\mathsf{def}}{=} \\
&\quad \mathit{</frame}\ := \\
&\quad\ \ \mathit{</world}\ :=\ \mathsf{EC\_REP\_SET}\ \Sigma\ \mathfrak{M}; \\
&\qquad \mathit{rel}\ := \\
&\qquad\ (\lambda\, n_1\ n_2. \\
&\qquad\quad \exists\, w_1\ w_2. \\
&\qquad\qquad w_1\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land\ w_2\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land\ n_1\ =\ \mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w_1\ \land \\
&\qquad\qquad n_2\ =\ \mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w_2\ \land \\
&\qquad\qquad \exists\, w'\ v'. \\
&\qquad\qquad\quad w'\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land\ v'\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \land \\
&\qquad\qquad\quad w'\ \in\ \mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w_1\ \land\ v'\ \in\ \mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w_2\ \land \\
&\qquad\qquad\quad \mathfrak{M}.\mathit{frame}.\mathit{rel}\ w'\ v')\ \mathit{/>}; \\
&\quad \mathit{valt}\ :=\ (\lambda\, p\ s.\ \exists\, w.\ s\ =\ \mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{VAR}\ p))\ \mathit{/>}
\end{aligned}
$$

It is routine to check the above definition does give a filtration:

Proposition 20 (`FLT_EXISTS`)**.**

$$\vdash \mathsf{CUS}\ \Sigma\ \Rightarrow\ \mathsf{filtration}\ \mathfrak{M}\ \Sigma\ (\mathsf{FLT}\ \mathfrak{M}\ \Sigma)$$

We are interested in two properties of filtration of models that directly give the proof of finite model property. Firstly, filtrating a model using a finite set gives a finite model:

Proposition 21 (`prop_2_38`)**.**

$$\vdash \mathsf{FINITE}\ \Sigma\ \land\ \mathsf{filtration}\ \mathfrak{M}\ \Sigma\ FLT\ \Rightarrow\ \mathsf{CARD}\ FLT.\mathit{frame.world}\ \leq\ 2\ **\ \mathsf{CARD}\ \Sigma$$

Proof. As Σ is finite, it suffices to give an injection from the world set of FLT to $\mathsf{POW}\ \Sigma$. The set of formulas in Σ that are satisfied at w can be regarded as a restricted version of τ-theory at w:

$$\mathsf{RESTRICT\_tau\_theory}\ \Sigma\ \mathfrak{M}\ w\ \overset{\mathsf{def}}{=}\ \{\ phi\ \mid\ phi\ \in\ \Sigma\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ phi\ \}$$

The injection is given by $\lambda\, w.\ \mathsf{RESTRICT\_tau\_theory}\ \Sigma\ \mathfrak{M}\ w$. □

Secondly, we need the satisfication of modal formula to be preserved under filtration, in the following sense:

Theorem 3 (`thm_2_39`)**.**

$$\begin{aligned}
&\vdash phi\ \in\ \Sigma\ \Rightarrow\\
&\quad \forall\, w.\\
&\qquad w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ \mathsf{filtration}\ \mathfrak{M}\ \Sigma\ FLT\ \Rightarrow\\
&\qquad (\mathsf{satis}\ \mathfrak{M}\ w\ phi\ \iff\ \mathsf{satis}\ FLT\ (\mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w)\ phi)
\end{aligned}$$

The above is proved by induction on modal formula. Putting the last two theorems together and we get the finite model property via filtration:

Theorem 4 (`thm_2_41`)**.**

$$\begin{aligned}
&\vdash \mathsf{satis}\ \mathfrak{M}\ w\ phi\ \Rightarrow\\
&\quad \exists\, \mathfrak{M}'\ w'.\ w'\ \in\ \mathfrak{M}'.\mathit{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}'\ w'\ phi\ \land\ \mathsf{FINITE}\ \mathfrak{M}'.\mathit{frame.world}
\end{aligned}$$

However, filtration method has its defect: Many interesting property of the model is not preserved by filtration. Therefore, sometimes we need to find out some good filtrations which preserve those properties. Some of them is easy, for instance, symmetry is preserved by the filtration we choose here:

Proposition 22 (`Rs_preserves_SYMM`)**.**

$$\begin{aligned}
&\vdash \mathsf{CUS}\ \Sigma\ \Rightarrow\\
&\quad (\forall\, a\ b.\\
&\qquad a\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ b\ \in\ \mathfrak{M}.\mathit{frame.world}\ \land\ \mathfrak{M}.\mathit{frame.rel}\ a\ b\ \Rightarrow\\
&\qquad \mathfrak{M}.\mathit{frame.rel}\ b\ a)\ \Rightarrow\\
&\quad \forall\, fa\ fb.\\
&\qquad fa\ \in\ (\mathsf{FLT}\ \mathfrak{M}\ \Sigma).\mathit{frame.world}\ \land\ fb\ \in\ (\mathsf{FLT}\ \mathfrak{M}\ \Sigma).\mathit{frame.world}\ \land\\
&\qquad (\mathsf{FLT}\ \mathfrak{M}\ \Sigma).\mathit{frame.rel}\ fa\ fb\ \Rightarrow\\
&\qquad (\mathsf{FLT}\ \mathfrak{M}\ \Sigma).\mathit{frame.rel}\ fb\ fa
\end{aligned}$$

And sometimes, we do need to be smart to find out a filtration that preserves interesting properties. A classical example is what we choose to preserve transitivity. It comes out that in order to preserve transitivity, we can define relation as:

Definition 31 (Relation that preserves transitivity)**.**

$$
\begin{aligned}
&\mathsf{REL\_2\_42}\ \Sigma\ \mathfrak{M} \overset{\mathsf{def}}{=} \\
&\quad (\lambda\, a\ b. \\
&\qquad \exists\, w. \\
&\qquad\quad w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ a\ =\ \mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w\ \wedge \\
&\qquad\quad \exists\, v. \\
&\qquad\qquad v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ b\ =\ \mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ v\ \wedge \\
&\qquad\qquad \forall\, phi. \\
&\qquad\qquad\quad \Diamond\ phi\ \in\ \Sigma\ \wedge\ \mathsf{satis}\ \mathfrak{M}\ v\ (\mathsf{DISJ}\ phi\ (\Diamond\ phi))\ \Rightarrow\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ phi))
\end{aligned}
$$

It does give a filtration:

Theorem 5 (`thm_2_42_i`)**.**

$$
\begin{aligned}
&\vdash (\forall\, u\ v\ w. \\
&\qquad u\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow \\
&\qquad \mathfrak{M}.\mathit{frame.rel}\ u\ v\ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ v\ w\ \Rightarrow \\
&\qquad \mathfrak{M}.\mathit{frame.rel}\ u\ w)\ \Rightarrow \\
&\quad \forall\, \Sigma. \\
&\qquad \mathsf{CUS}\ \Sigma\ \Rightarrow \\
&\qquad \mathsf{filtration}\ \mathfrak{M}\ \Sigma \\
&\qquad\ \ <|\mathit{frame}\ := \\
&\qquad\quad <|\mathit{world}\ :=\ \mathsf{EC\_REP\_SET}\ \Sigma\ \mathfrak{M}; \\
&\qquad\quad\ \ \mathit{rel}\ :=\ (\lambda\, w_1\ w_2.\ \mathsf{REL\_2\_42}\ \Sigma\ \mathfrak{M}\ (\mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w_1)\ (\mathsf{EC\_CUS}\ \Sigma\ \mathfrak{M}\ w_2))\,|>; \\
&\qquad\ \ \mathit{valt}\ :=\ (\lambda\, p\ s.\ \exists\, w.\ s\ =\ \mathsf{EC\_REP}\ \Sigma\ \mathfrak{M}\ w\ \wedge\ \mathsf{satis}\ \mathfrak{M}\ w\ (\mathsf{VAR}\ p))\,|>
\end{aligned}
$$

and moreover, it preserves transitivity:

Theorem 6 (`thm_2_42_ii`)**.**

$$
\begin{aligned}
&\vdash (\forall\, u\ v\ w. \\
&\qquad u\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow \\
&\qquad \mathfrak{M}.\mathit{frame.rel}\ u\ v\ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ v\ w\ \Rightarrow \\
&\qquad \mathfrak{M}.\mathit{frame.rel}\ u\ w)\ \Rightarrow \\
&\quad \forall\, \Sigma. \\
&\qquad \mathsf{CUS}\ \Sigma\ \Rightarrow \\
&\qquad \forall\, a\ b\ c.\ \mathsf{REL\_2\_42}\ \Sigma\ \mathfrak{M}\ a\ b\ \wedge\ \mathsf{REL\_2\_42}\ \Sigma\ \mathfrak{M}\ b\ c\ \Rightarrow\ \mathsf{REL\_2\_42}\ \Sigma\ \mathfrak{M}\ a\ c
\end{aligned}
$$

3.2 Finite model property via selection

Another method to build finite models for an arbitary model is by selection. That is, we start with a model that the formula ϕ is satisfied, and delete points from the model and only leaves finitely many worlds in it. The intuition of this method is any modal formula can only contain finitely many diamond, each can see one step from the current state. Therefore, any formula can only capture the information of finitely depth. To make the notion of 'depth' precise, we define the degree of a modal formula, which counts the number of diamonds appear in a model formula:

Definition 32 (Degree of a modal formula)**.**

$$
\begin{aligned}
&\mathsf{DEG}\ (\mathsf{VAR}\ p) \overset{\text{def}}{=} 0\\
&\mathsf{DEG}\ \bot \overset{\text{def}}{=} 0\\
&\mathsf{DEG}\ (\neg\phi) \overset{\text{def}}{=} \mathsf{DEG}\ \phi\\
&\mathsf{DEG}\ (\mathsf{DISJ}\ \phi_1\ \phi_2) \overset{\text{def}}{=} \mathsf{MAX}\ (\mathsf{DEG}\ \phi_1)\ (\mathsf{DEG}\ \phi_2)\\
&\mathsf{DEG}\ (\Diamond\ \phi) \overset{\text{def}}{=} \mathsf{DEG}\ \phi + 1
\end{aligned}
$$

A modal formula of degree zero is a modal formula without diamond, and hence is a propositional formula:

Proposition 23 (`DEG_0_propform`)**.**

$$\vdash \mathsf{DEG}\ f = 0 \iff \mathsf{propform}\ f$$

For our proof of finite model property, the crucial fact we need about the degree of formulas is that if our mdal language is on a finite set of propositional letters, then for each n, up to logical equivalence, there are only finitely many modal formulas up to degree n. The proof is long and technical, we discuss it in the following interlude.

3.3 Interlude I: Finiteness of non-equivalent modal formulas in each degree

The aim of this interlude is to prove:

Lemma 2 (`prop_2_29_strengthen`)**.**

$$
\begin{aligned}
&\vdash \mathsf{FINITE}\ s \wedge \mathsf{INFINITE}\,\mathcal{U}(:\beta) \Rightarrow\\
&\quad \forall n.\ \mathsf{FINITE}\ (\{\ f\ /\ \mathsf{DEG}\ f \leq n \wedge \forall a.\ \mathsf{VAR}\ a \in \mathsf{subforms}\ f \Rightarrow a \in s\ \}//E\ \mu)
\end{aligned}
$$

Here the '$//E\ \mu$ takes the partition of a set by the equivalence relation $\mathsf{equiv0}\ \mu$. The proof of this lemma is by induction on the degree n.

3.3.1 Base case

For the base case, we need to prove if we only use propositional letters in s where s is a finite set, then up to equivalence, we can only onbtain finitely many propositional formulas. To prove this, it suffices to find out an injection from the set of equivalence class of propositional formulas that only uses propositional letters in s to a finite set. Our claim is that the function $\lambda\, eqc.\ \mathsf{IMAGE}\ (\lambda f.\ \{\ s \mid \mathsf{peval}\ s\ f\ \} \cap \mathsf{POW}\ s)\ eqc$ is an injection from our set from $\{ f \mid \mathsf{propform}\ f \ \wedge\ \forall a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\ f \ \Rightarrow\ a \ \in\ s \}//E\ \mu$ to the set $\mathsf{POW}\ (\mathsf{POW}\ (\mathsf{POW}\ s))$, which is finite because s is.

Let us firstly investigate what the function $\lambda f.\ \{ s \mid \mathsf{peval}\ s\ f \} \ \cap\ \mathsf{POW}\ s$ does. For a formula f, $\{ s \mid \mathsf{peval}\ s\ f \} \ \cap\ \mathsf{POW}\ s$ is the set of the assignment of truth values on all propositional letters in s that makes f hold. If f_1 and f_2 are equivalent, then for any such assignment, it makes f_1 true iff it makes f_2 true:

Proposition 24 (`peval_equiv0`).

$$\begin{aligned}&\vdash \mathsf{propform}\ f_1 \ \wedge\ \mathsf{propform}\ f_2 \ \wedge\ (\forall \mathfrak{M}\ w.\ \mathsf{satis}\ \mathfrak{M}\ w\ f_1 \iff \mathsf{satis}\ \mathfrak{M}\ w\ f_2) \ \Rightarrow\\ &\quad \forall \sigma.\ \mathsf{peval}\ \sigma\ f_1 \iff \mathsf{peval}\ \sigma\ f_2\end{aligned}$$

Proof. Under the assumptions, if we have a $\mathsf{peval}\ \sigma\ f_1 \ \wedge\ \neg\mathsf{peval}\ \sigma\ f_2$, then we can construct a model with only one world w, no relation, with the valuation at w defined for propositional symbols as σ. Then by `peval_satis`, we get $\mathsf{satis}\ \mathfrak{M}\ w\ f_1 \ \wedge\ \neg\mathsf{satis}\ \mathfrak{M}\ w\ f_2$, a contradiction. □

As an easy consequence, we have:

Proposition 25 (`partition_to_peval_well_defined`).

$$\begin{aligned}&\vdash \mathsf{propform}\ f_1 \ \wedge\ \mathsf{propform}\ f_2 \ \wedge\ \mathsf{equiv0}\ \mu\ f_1\ f_2 \ \Rightarrow\\ &\quad (\lambda f\ s.\ \mathsf{peval}\ s\ f)\ f_1 \ =\ (\lambda f\ s.\ \mathsf{peval}\ s\ f)\ f_2\end{aligned}$$

means that if two formulas are equivalent, then the sets of valuations that makes them hold are identical. As a consequence, these two sets will still be equal after taking inter section with $\mathsf{POW}\ s$.

This is saying that the image of $\lambda f.\ \{ s \mid \mathsf{peval}\ s\ f \} \ \cap\ \mathsf{POW}\ s$ on each equivalence class is the singleton $\{ \{ s \mid \mathsf{peval}\ s\ f \} \ \cap\ \mathsf{POW}\ s \}$:

Proposition 26 (`IMAGE_peval_singlton_strengthen`).

$$\begin{aligned}&\vdash x \ \in\ \{ f \mid \mathsf{propform}\ f \ \wedge\ \forall a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\ f \ \Rightarrow\ a \ \in\ s \}//E\ \mu \ \wedge\ \phi \ \in\ x \ \Rightarrow\\ &\quad \mathsf{IMAGE}\ (\lambda f.\ \{ \sigma \mid \mathsf{peval}\ \sigma\ f \} \ \cap\ \mathsf{POW}\ s)\ x \ =\ \{ \{ \sigma \mid \mathsf{peval}\ \sigma\ \phi \} \ \cap\ \mathsf{POW}\ s \}\end{aligned}$$

Recall we proved the condition for a propositional formula to hold as the lemma `peval_satis_strengthen`, we use it here to prove the follows:

Proposition 27 (`equiv0_peval_strengthen`).

$$\begin{aligned}&\vdash \mathsf{propform}\ f_1 \ \wedge\ \mathsf{propform}\ f_2 \ \wedge\ (\forall a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\ f_1 \ \Rightarrow\ a \ \in\ s) \ \wedge\\ &\quad (\forall a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\ f_2 \ \Rightarrow\ a \ \in\ s) \ \Rightarrow\\ &\quad (\forall \sigma.\ \sigma \ \in\ \mathsf{POW}\ s \ \Rightarrow\ (\mathsf{peval}\ \sigma\ f_1 \iff \mathsf{peval}\ \sigma\ f_2)) \ \Rightarrow\\ &\quad \forall \mathfrak{M}\ w.\ \mathsf{satis}\ \mathfrak{M}\ w\ f_1 \iff \mathsf{satis}\ \mathfrak{M}\ w\ f_2\end{aligned}$$

Proof. Suppose $\mathsf{satis}\ \mathfrak{M}\ w\ f_1$ for some model $\mathfrak{M}$, then by `peval_satis_strengthen`, we have $\mathsf{peval}\ ((\lambda a.\ w \ \in\ \mathfrak{M}.\mathsf{valt}\ a) \ \cap\ s)\ f_1$. As $(\lambda a.\ w \ \in\ \mathfrak{M}.\mathsf{valt}\ a) \ \cap\ s$ gives a subset of s, we conclude $\mathsf{peval}\ ((\lambda a.\ w \ \in\ \mathfrak{M}.\mathsf{valt}\ a) \ \cap\ s)\ f_2$ by assumption, and then $\mathsf{satis}\ \mathfrak{M}\ w\ f_2$ by `peval_satis_strengthen` again. The other direction is the same. □

We can now prove the injection:

Proposition 28 (`INJ_peval_partition_strengthen`)**.**

$$\begin{aligned}&\vdash \mathsf{INJ}\ (\lambda\, eqc.\ \mathsf{IMAGE}\ (\lambda f.\ \{\ s\ \mid \mathsf{peval}\ s\ f\ \}\ \cap\ \mathsf{POW}\ s)\ eqc)\\ &\quad (\{\ f\ \mid \mathsf{propform}\ f\ \wedge\ \forall\, a.\ \mathsf{VAR}\ a\ \in\ \mathsf{subforms}\ f\ \Rightarrow\ a\ \in\ s\ \}//E\ \mu)\\ &\quad (\mathsf{POW}\ (\mathsf{POW}\ (\mathsf{POW}\ s)))\end{aligned}$$

Proof. If two equivalence classes represented by x and x' respectively has identical image under the given function, by `IMAGE_peval_singlton_strengthen`, it gives $\{\ s\ \mid \mathsf{peval}\ s\ x\ \}\ \cap\ \mathsf{POW}\ s\ =\ \{\ s\ \mid \mathsf{peval}\ s\ x'\ \}\ \cap\ \mathsf{POW}\ s$. This means $\forall\sigma.\ \sigma\ \in\ \mathsf{POW}\ s\ \Rightarrow\ \mathsf{peval}\ \sigma\ x\ \iff\ \mathsf{peval}\ \sigma\ x'$, which means x and x' are equivalent by `equiv0_peval_strengthen`. □

3.3.2 Step case

The idea of the step case is completely different from that of base case. The key observation that any formulas which only uses propositional letters in a fixed finite set ss and of degree no more than $n+1$ is obtained by applying $\neg$ and $\vee$ to combin the propositional letters and formulas of form $\Diamond\ phi$ for phi of degree no more than n. Such a combination is called a boolean combination. We define boolean combination as an inductive relation, where $\mathsf{IBC}\ f\ s$ reads 'f is a boolean combination of elements in the set s':

Definition 33 (Rules of boolean combination)**.**

$$\frac{\mathsf{IBC}\ f_1\ s \quad \mathsf{IBC}\ f_2\ s}{\mathsf{IBC}\ (\mathsf{DISJ}\ f_1\ f_2)\ s} \qquad \frac{}{\mathsf{IBC}\ \bot\ s} \qquad \frac{\mathsf{IBC}\ f\ s}{\mathsf{IBC}\ (\neg f)\ s} \qquad \frac{f\ \in\ s}{\mathsf{IBC}\ f\ s}$$

Our claim about in the last paragraph can be proved by induction:

Proposition 29 (`DEG_IBC_strengthen`)**.**

$$\begin{aligned}&\vdash \mathsf{DEG}\ x\ \le\ n\ +\ 1\ \wedge\ (\forall\, a.\ \mathsf{VAR}\ a\ \in\ \mathsf{subforms}\ x\ \Rightarrow\ a\ \in\ s)\ \iff\\ &\quad \mathsf{IBC}\ x\\ &\qquad (\{\ \mathsf{VAR}\ v\ \mid v\ \in\ s\ \}\ \cup\\ &\qquad \{\ \Diamond\ psi\ \mid \mathsf{DEG}\ psi\ \le\ n\ \wedge\ \forall\, a.\ \mathsf{VAR}\ a\ \in\ \mathsf{subforms}\ psi\ \Rightarrow\ a\ \in\ s\ \})\end{aligned}$$

Observe a formula which is a boolean combinations on an empty set is either equivalent to $\bot$ or $\top$:

Lemma 3 (`IBC_EMPTY_Lemma`)**.**

$$\vdash \mathsf{IBC}\ f\ s\ \Rightarrow\ s\ =\ \emptyset\ \Rightarrow\ \mathsf{equiv0}\ \mu\ f\ \mathsf{TRUE}\ \vee\ \mathsf{equiv0}\ \mu\ f\ \bot$$

Hence it is reasonable to include the assumption that the set s is non-empty in many lemmas along the proof of main result, since the empty case can be left to treat separately at the very end. Our aim now is to prove the set of boolean combinations on a set which contains only finitely many non-equivalent formulas only contain finitely many non-equivalent formulas:

Lemma 4 (`FINITE_FINITE_IBC`)**.**

$$\vdash fs\ \ne\ \emptyset\ \Rightarrow\ \mathsf{FINITE}\ (fs//E\ \mu)\ \Rightarrow\ \mathsf{FINITE}\ (\{\ f\ \mid \mathsf{IBC}\ f\ fs\ \}//E\ \mu)$$

It suffices to prove the set of formulas obtained by boolean combination on a finite set is finite up to equivalence. Since then as fs have only finitely many non-equivalent formulas, the set $\mathsf{IMAGE\ CHOICE}\ (fs//E\ \mu)$ of representatives of the equivalence classes is finite. There is a surjection $\lambda\, s.\ \{\ y\ \mid\ \mathsf{IBC}\ y\ fs\ \wedge\ \forall f.\ f\ \in\ s\ \Rightarrow\ \mathsf{equiv0}\ \mu\ y\ f\ \}$ from $\{\ f\ \mid\ \mathsf{IBC}\ f\ (\mathsf{IMAGE\ CHOICE}\ (fs//E\ \mu))\ \}//E\ \mu$ to $\{\ f\ \mid\ \mathsf{IBC}\ f\ fs\ \}//E\ \mu$ is a surjection, showing the codomain is finite.

The strategy we used to prove this is to prove that any formula obtained by a boolean combination on a set is equivalent to a formula in disjunction normal form on the same set, and the formulas of disjunction normal form on a finite set is finite. The disjunction normal form is defined by:

Definition 34 (Disjunction normal form)**.**

$$\mathsf{DNF\_OF}\ f\ fs\ \overset{\text{def}}{=}\ \mathsf{DISJ\_OF}\ f\ \{\ c\ /\ \mathsf{CONJ\_OF}\ c\ fs\ \}$$

The definition has two layers, the outer layer $\mathsf{CONJ\_OF}$ is also defined inductively:

Definition 35 ($\mathsf{CONJ\_OF}$)**.**

$$\frac{c\ =\ c_0\ \vee\ c\ =\ \neg c_0}{\mathsf{CONJ\_OF}\ c\ \{\ c_0\ \}}$$

$$\frac{f_1\ =\ f_0\ \vee\ f_1\ =\ \neg f_0\quad f_0\ \in\ fs\quad \mathsf{CONJ\_OF}\ f_2\ (fs\ \mathsf{DELETE}\ f_0)}{\mathsf{CONJ\_OF}\ (\mathsf{AND}\ f_1\ f_2)\ fs}$$

For the inner layer, we say $\mathsf{DISJ\_OF}\ f\ fs$ when f is either the '$\bot$' or a formula such that $\mathsf{DISJ\_OF0}\ f\ fs$, where the inductive relation is given by:

Definition 36 ($\mathsf{DISJ\_OF0}$)**.**

$$\frac{f\ \in\ fs}{\mathsf{DISJ\_OF0}\ f\ fs}$$

$$\frac{f_1\ \in\ fs\quad \mathsf{DISJ\_OF0}\ f_2\ (fs\ \mathsf{DELETE}\ f_1)}{\mathsf{DISJ\_OF0}\ (\mathsf{DISJ}\ f_1\ f_2)\ fs}$$

As an example, the four $\mathsf{CONJ\_OF}$ formulas on the set $\{f1, f2\}$ are $\mathsf{AND}\ f_1\ f_2$, $\mathsf{AND}\ (\neg f_1)\ f_2$, $\mathsf{AND}\ f_1\ (\neg f_2)$ and $\mathsf{AND}\ (\neg f_1)\ (\neg f_2)$. For another example, consider the set $fs := \{f1, f2, f3\}$ of three formulas, we can say $\mathsf{CONJ\_OF}\ (\mathsf{AND}\ f_3\ (\mathsf{AND}\ f_1\ f_2))\ fs$, $\mathsf{CONJ\_OF}\ (\mathsf{AND}\ f_2\ (\mathsf{AND}\ (\neg f_1)\ f_3))\ fs$, $\mathsf{CONJ\_OF}\ (\mathsf{AND}\ (\neg f_3)\ (\mathsf{AND}\ f_2\ f_1))\ fs$ and so on. But it is not the fact that $\mathsf{CONJ\_OF}\ f_1\ fs$, since from the definition, any element of fs or its negation must appear exactly once in a $\mathsf{CONJ\_OF}$ formula on fs. Let $A\ =\ \{\ f\ \mid\ \mathsf{CONJ\_OF}\ f\ fs\ \}$, then we can say $\mathsf{DISJ\_OF}\ (\mathsf{DISJ}\ (\mathsf{AND}\ f_3\ (\mathsf{AND}\ f_1\ f_2))\ (\mathsf{AND}\ f_2\ (\mathsf{AND}\ (\neg f_1)\ f_3)))\ A$, $\mathsf{DISJ\_OF}\ (\mathsf{AND}\ f_2\ (\mathsf{AND}\ (\neg f_1)\ f_3))\ A$ and so on, so both these two formulas are disjunction normal form on fs, but we cannot say $\mathsf{DISJ\_OF}\ (\mathsf{DISJ}\ (\mathsf{AND}\ f_2\ (\mathsf{AND}\ (\neg f_1)\ f_3))\ (\mathsf{AN}$ or $\mathsf{DISJ\_OF}\ (\mathsf{DISJ}\ (\mathsf{AND}\ f_2\ (\mathsf{AND}\ (\neg f_1)\ f_3))\ (\neg\mathsf{AND}\ f_2\ (\mathsf{AND}\ (\neg f_1)\ f_3)))\ A$, since any formula in K or its negation is only allowed to appear in a $\mathsf{DISJ\_OF}$ formula on K for at most once, so these two are not in disjunction normal form.

If we start with a finite set, by using `FINITE_COMPLETE_INDUCTION` , we conclude the set of $\mathsf{CONJ\_OF}$ formulas and $\mathsf{DISJ\_OF0}$ are both finite, hence the $\mathsf{DNF\_OF}$ formulas on this set is finite:

Proposition 30 (`FINITE_CONJ_OF`)**.**

$$\vdash \mathsf{FINITE}\ s\ \Rightarrow\ \mathsf{FINITE}\ \{\ f\ /\ \mathsf{CONJ\_OF}\ f\ s\ \}$$

Proposition 31 (`FINITE_DISJ_OF0`).

$$\vdash \mathsf{FINITE}\ s \ \Rightarrow\ \mathsf{FINITE}\ \{\, f \ /\ \mathsf{DISJ\_OF0}\, f\ s \,\}$$

Proposition 32 (`FINITE_DNF`).

$$\vdash \mathsf{FINITE}\, \mathit{fs} \ \Rightarrow\ \mathsf{FINITE}\ \{\, f \ /\ \mathsf{DNF\_OF}\, f\ \mathit{fs} \,\}$$

As discussed earlier, once we prove:

Theorem 7 (`IBC_DNF_EXISTS`).

$$\vdash \mathsf{IBC}\, f\ \mathit{fs} \ \Rightarrow\ \mathsf{FINITE}\, \mathit{fs} \ \wedge\ \mathit{fs} \neq \emptyset \ \Rightarrow\ \exists\, p.\ \mathsf{DNF\_OF}\ p\ \mathit{fs} \ \wedge\ \mathsf{equiv0}\ \mu\, f\ p$$

we get `FINITE_FINITE_IBC`, which leads to a proof of the step case of `prop_2_29_strengthen` goes as follows:

Proof. Suppose $\{\, f \mid \mathsf{DEG}\, f \ \leq\ n \ \wedge\ \forall\, a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\, f \ \Rightarrow\ a \ \in\ s \,\}//E\ \mu$ is finite and let $A\ =\ \{\, \mathsf{VAR}\ v \mid v \ \in\ s \,\} \ \cup\ \{\, \Diamond\ \mathit{psi} \mid \mathsf{DEG}\ \mathit{psi} \ \leq\ n \ \wedge\ \forall\, a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\ \mathit{psi} \ \Rightarrow\ a \ \in\ s \,\}$. By `DEG_IBC_strengthen`, we have $B :=\{\, f \mid \mathsf{DEG}\, f \ \leq\ n\ +\ 1 \ \wedge\ \forall\, a.\ \mathsf{VAR}\ a \ \in\ \mathsf{subforms}\, f \ \Rightarrow\ a \ \in\ s \,\} \ =\ \{\, \mathit{phi} \mid \mathsf{IBC}\ \mathit{phi}\ A \,\}$, we prove B is finite up to equivalence. A cannot be empty, since it must contains $\Diamond$ TRUE and $\Diamond\ \bot$. By `FINITE_FINITE_IBC`, it suffices to prove A is finite up to equivalence. Under the assumption $\mathsf{INFINITE}\ \mathcal{U}(:\beta)$, $\mathsf{equiv0}\ \mu\ (\Diamond\ f)\ (\Diamond\ g)$ iff $\mathsf{equiv0}\ \mu\ f\ g$, so the inductive hypothesis together with the assumption $\mathsf{FINITE}\ s$ gives the finiteness of A. □

Therefore, it remains to give a proof of `IBC_DNF_EXISTS`. The proof is by rule induction on IBC, the things to prove are:

Base case:

- The falsity $\bot$ is equivalent to a disjunction normal form on fs.
- For any element $f \in fs$, it is equivalent to a disjunction normal form on fs.

Step case:

- Under the assumption that fs is finite and non-empty, if $f1, f2$ are boolean combination of the set fs which are equivalent to $p1, p2$ of disjunction normal form respectively, then $\mathsf{DISJ}\ f_1\ f_2$ is equivalent to a formula in disjunction normal form.
- With the same assumption on fs, if f is a boolean combination of fs and equivalent to p in disjunction normal form, then $\neg f$ is equivalent to a formula in disjunction normal form.

The first item of base case is trivial since $\bot$ itself is in disjunction normal form, we begin by proving the second item of the base case.

3.3.3 Case for $f \in fs$

We aim to prove:

Lemma 5 (`IBC_DNF_EXISTS_case4`)**.**

$$\vdash \mathsf{FINITE}\, fs \ \wedge\ fs \ \neq\ \emptyset \ \Rightarrow\ \forall f.\, f \ \in\ fs \ \Rightarrow\ \exists\, p.\, \mathsf{DNF\_OF}\ p\ fs \ \wedge\ \mathsf{equiv0}\ \mu\ f\ p$$

Let us consider what does the DNF_OF formula p on fs that is equivalent to an element of fs look like: We require p to be satisfied if and only if f is satisfied. As p is of disjunction normal form, p is satisfied once some of its disjuncts is satisfied. Hence we require the satisfication of each disjuncts of p to imply the satisfication of f, that is, f is a conjunct of each disjunct of p. So up to rearrangement, p is a disjunction of conjunctions $f \wedge f_1 \cdots f_n$ where each formula in $fs/\{f\}$ or its negation appears exactly once in these f_i's. Such a formula is equivalent to $f \wedge g$, where g is the disjunction of the formulas obtained by taking f out of each conjunct of p. And $f \wedge g$ is equivalent to f if and only if g is equivalent to $\top$. Hence, g must be the disjunction of all the CONJ_OF formulas on $fs/\{f\}$. By conclusion, a disjunction normal form we need can be taken as the disjunction of conjunctions starting with f, with its tail ranging over all possible combination of negated and unnegated formulas in $fs/\{f\}$. Use the set $\{f_1, f_2, f_3\}$ as example again, an example of disjunction normal form equivalent to f_1 is the formula $(f_1 \wedge f_2 \wedge f_3) \vee (f_1 \wedge \neg f_2 \wedge f_3) \vee (f_1 \wedge f_2 \wedge \neg f_3) \vee (f_1 \wedge \neg f_2 \wedge \neg f_3)$, note that such a formula is equivalent to $f_1 \wedge ((f_2 \wedge f_3) \vee (\neg f_2 \wedge f_3) \vee (f_2 \wedge \neg f_3) \vee (\neg f_2 \wedge \neg f_3))$, where $(f_2 \wedge f_3) \vee (\neg f_2 \wedge f_3) \vee (f_2 \wedge \neg f_3) \vee (\neg f_2 \wedge \neg f_3)$ is equivalent to TRUE.

To formalise the idea above, we want to be able to building disjunction normal forms piece by piece. We use the following definitions as our tool:

Definition 37 (A literal to a formula)**.**

$$\begin{aligned}\mathsf{negf}\ (f, \mathsf{T}) &\overset{\text{def}}{=} f\\ \mathsf{negf}\ (f, \mathsf{F}) &\overset{\text{def}}{=} \neg f\end{aligned}$$

Definition 38 (A list of literals to a conjunction)**.**

$$\begin{aligned}\mathsf{lit\_list\_to\_form}\ [] &\overset{\text{def}}{=} \mathsf{TRUE}\\ \mathsf{lit\_list\_to\_form}\ [fb] &\overset{\text{def}}{=} \mathsf{negf}\ fb\\ \mathsf{lit\_list\_to\_form}\ (fb :: v_2 :: v_3) &\overset{\text{def}}{=} \mathsf{AND}\ (\mathsf{negf}\ fb)\ (\mathsf{lit\_list\_to\_form}\ (v_2 :: v_3))\end{aligned}$$

Definition 39 (A list of literals to a disjunction)**.**

$$\begin{aligned}\mathsf{lit\_list\_to\_form2}\ [] &\overset{\text{def}}{=} \bot\\ \mathsf{lit\_list\_to\_form2}\ [fb] &\overset{\text{def}}{=} fb\\ \mathsf{lit\_list\_to\_form2}\ (fb :: v_2 :: v_3) &\overset{\text{def}}{=} \mathsf{DISJ}\ fb\ (\mathsf{lit\_list\_to\_form2}\ (v_2 :: v_3))\end{aligned}$$

A pair (f, tv) where $f \in fs$ and tv is a truth value is called a literal, it encodes a formula in fs or its negation. Such a pair is turned to a formula f or $\neg f$ by negf, depends on the truth value given in its second coordinate. For a finite fs, a non-empty list l of literals with the condition that set (MAP FST l) $=$ fs and ALL_DISTINCT (MAP FST l) is interesting. Here MAP takes a function f and a list $[a_1; \cdots, a_n]$, and return the list $[f(a_1); \cdots ; f(a_n)]$. The function set takes a list and gives the set of its members, ALL_DISTINCT also takes a list, and return $\top$ iff all members of the list are distinct. These conditions precisely say that each member of fs occurs exactly once in the list. As a consequence, we can build CONJ_OF formulas from such list using `lit_list_to_form`. Conversely, each CONJ_OF formula can be obtained from such a list.

Proposition 33 (`list_to_CONJ_OF`).

$$\begin{array}{l} \vdash l \neq [] \Rightarrow \\ \quad \mathsf{set}\ (\mathsf{MAP\ FST}\ l) = fs \wedge \mathsf{ALL\_DISTINCT}\ (\mathsf{MAP\ FST}\ l) \Rightarrow \\ \quad \mathsf{CONJ\_OF}\ (\mathsf{lit\_list\_to\_form}\ l)\ fs \end{array}$$

Proposition 34 (`CONJ_OF_AND_lemma`).

$$\begin{array}{l} \vdash \mathsf{CONJ\_OF}\ f\ fs \Rightarrow \\ \quad \exists\, l. \\ \qquad \mathsf{set}\ (\mathsf{MAP\ FST}\ l) = fs \wedge \mathsf{ALL\_DISTINCT}\ (\mathsf{MAP\ FST}\ l) \wedge \\ \qquad f = \mathsf{lit\_list\_to\_form}\ l \end{array}$$

Under the same condition on the list, the corresponding result holds for `lit_list_to_form` and DISJ_OF0:

Proposition 35 (`list_to_DISJ_OF0`).

$$\vdash l \neq [] \wedge \mathsf{set}\ l \subseteq fs \wedge \mathsf{ALL\_DISTINCT}\ l \Rightarrow \mathsf{DISJ\_OF0}\ (\mathsf{lit\_list\_to\_form2}\ l)\ fs$$

Proposition 36 (`DISJ_OF0_DISJ_lemma_EQ`).

$$\begin{array}{l} \vdash \mathsf{DISJ\_OF0}\ f\ fs \Rightarrow \\ \quad \exists\, l.\ l \neq [] \wedge \mathsf{set}\ l \subseteq fs \wedge f = \mathsf{lit\_list\_to\_form2}\ l \wedge \mathsf{ALL\_DISTINCT}\ l \end{array}$$

Put the two things together, we can build DNF_OF formulas using `lit_list_to_form` and `list_list_-to_form2`:

Proposition 37 (`list_to_DNF_lemma`).

$$\begin{array}{l} \vdash ld \neq [] \wedge \mathsf{ALL\_DISTINCT}\ ld \wedge \\ \quad (\forall\, d. \\ \qquad \mathsf{MEM}\ d\ ld \Rightarrow \\ \qquad \exists\, lc. \\ \qquad\quad lc \neq [] \wedge d = \mathsf{lit\_list\_to\_form}\ lc \wedge \mathsf{set}\ (\mathsf{MAP\ FST}\ lc) = fs \wedge \\ \qquad\quad \mathsf{ALL\_DISTINCT}\ (\mathsf{MAP\ FST}\ lc)) \Rightarrow \\ \quad \mathsf{DNF\_OF}\ (\mathsf{lit\_list\_to\_form2}\ ld)\ fs \end{array}$$

`lit_list_to_form2` is distributive and symmetric, both are proved by induction on list. Certainly, similar results hold for `lit_list_to_form`, but we do not need them here:

Proposition 38 (`list_demorgan`).

$$\begin{array}{l} \vdash l \neq [] \Rightarrow \\ \quad \mathsf{equiv0}\ \mu\ (\mathsf{AND}\ e\ (\mathsf{lit\_list\_to\_form2}\ l)) \\ \qquad (\mathsf{lit\_list\_to\_form2}\ (\mathsf{MAP}\ (\lambda\, a.\ \mathsf{AND}\ e\ a)\ l)) \end{array}$$

Proposition 39 (`lit_list_to_form2_SYM`).

$$\vdash \mathsf{set}\ l_1 \ = \ \mathsf{set}\ l_2 \ \Rightarrow\ \mathsf{equiv0}\ \mu\ (\mathsf{lit\_list\_to\_form2}\ l_1)\ (\mathsf{lit\_list\_to\_form2}\ l_2)$$

Use the lemmas above, by induction on finiteness, we can show the thing we illustrated in an example before: the disjunction of all possible conjunctions is equivalent to the truth:

Proposition 40 (`ALL_POSSIBLE_VALUE_TRUE`).

$$\begin{aligned}&\vdash \mathsf{FINITE}\ fs\ \Rightarrow\\ &\quad fs\ \neq\ \emptyset\ \Rightarrow\\ &\quad \mathsf{equiv0}\ \mu\ (\mathsf{lit\_list\_to\_form2}\ (\mathsf{SET\_TO\_LIST}\ \{\ c\ \ /\ \mathsf{CONJ\_OF}\ c\ fs\ \}))\ \mathsf{TRUE}\end{aligned}$$

Now we launch on the proof of `IBC_DNF_EXISTS_case4`:

Proof. Given an $f \in fs$, the desired p is given by $\phi := \mathsf{lit\_list\_to\_form2}$
$(\mathsf{SET\_TO\_LIST}\ \{\ \mathsf{AND}\ f\ c\ \mid\ c\ \mid\ \mathsf{CONJ\_OF}\ c\ (fs\ \mathsf{DELETE}\ f)\ \}\,)$. Here SET_TO_LIST is a function takes a set and form a list with all the elements of the set as members. There are two things to check: Using `CONJ_OF_-AND_lemma` and `list_to_DNF_lemma`, it is immediate to prove ϕ is in disjunction normal form. To prove ϕ is equivalent to f, we have a sequence of equivalence:

$\equiv\mathsf{lit\_list\_to\_form2}$
$(\mathsf{SET\_TO\_LIST}\ \{\ \mathsf{AND}\ f\ c\ \mid\ c\ \mid\ \mathsf{CONJ\_OF}\ c\ (fs\ \mathsf{DELETE}\ f)\ \}\,)$

$\equiv\mathsf{lit\_list\_to\_form2}$
$(\mathsf{MAP}\ (\lambda\, a.\ \mathsf{AND}\ f\ \ a)\ (\mathsf{SET\_TO\_LIST}\ \{\ c\ \mid\ \mathsf{CONJ\_OF}\ c\ (fs\ \mathsf{DELETE}\ f)\ \}\,))$

$\equiv\mathsf{AND}\ f\ \ (\mathsf{lit\_list\_to\_form2}\ (\mathsf{SET\_TO\_LIST}\ \{\ c\ \mid\ \mathsf{CONJ\_OF}\ c\ (fs\ \mathsf{DELETE}\ f)\ \}\,))$

$\equiv\mathsf{AND}\ f\ \mathsf{TRUE}$

$\equiv f$ □

Hence we are done with the base case of `IBC_DNF_EXISTS`.

3.3.4 Case for disjunction

The following lemma directly implies our goal by the fact that if $f1$ is equivalent $p1$ and $f2$ is equivalent to $p2$, then $f1 \vee f2$ is equivalent to $p1 \vee p2$ and the transitivity of being equivalent.

Lemma 6 (`DNF_OF_DISJ_equiv0`).

$$\vdash \mathsf{DNF\_OF}\ p_1\ fs\ \wedge\ \mathsf{DNF\_OF}\ p_2\ fs\ \Rightarrow\ \exists f.\ \mathsf{DNF\_OF}\ f\ fs\ \wedge\ \mathsf{equiv0}\ \mu\ f\ \ (\mathsf{DISJ}\ p_1\ p_2)$$

Expanding the definition of DNF_OF and DISJ_OF0 gives four cases, we prove the only interesting one separately:

Proposition 41 (`DNF_OF_DISJ_equiv0_case4`).

$$\begin{aligned}&\vdash \mathsf{DISJ\_OF0}\ p_1\ fs\ \Rightarrow\\ &\quad \forall\, p_2.\ \mathsf{DISJ\_OF0}\ p_2\ fs\ \Rightarrow\ \exists f.\ \mathsf{DISJ\_OF0}\ f\ fs\ \wedge\ \mathsf{equiv0}\ \mu\ f\ \ (\mathsf{DISJ}\ p_1\ p_2)\end{aligned}$$

Proof. By induction on $\mathsf{DISJ\_OF0}$, the base case is by a straightfoward two-layer induction, for the step case, suppose $f_1 \in fs$ and $\mathsf{DISJ\_OF0}\ p_1\ (fs\ \mathsf{DELETE}\ f_1)$ and $\mathsf{DISJ\_OF0}\ p_2\ fs$, we are asked to prove:

$\exists f.\ \mathsf{DISJ\_OF0}\ f\ fs \ \wedge\ \mathsf{equiv0}\ \mu\ f\ (\mathsf{DISJ}\ (\mathsf{DISJ}\ f_1\ p_1)\ p_2)$ from the inductive hypothesis

$\forall p_2.$

$\mathsf{DISJ\_OF0}\ p_2\ (fs\ \mathsf{DELETE}\ f_1) \ \Rightarrow$

$\exists f.\ \mathsf{DISJ\_OF0}\ f\ (fs\ \mathsf{DELETE}\ f_1) \ \wedge\ \mathsf{equiv0}\ \mu\ f\ (\mathsf{DISJ}\ p_1\ p_2)$

If $\mathsf{DISJ\_OF0}\ p_2\ (fs\ \mathsf{DELETE}\ f_1)$, then we are done by inductive hypothesis, otherwise we need a trick: If $\mathsf{DISJ\_OF0}\ p_2\ fs$ but $\neg\mathsf{DISJ\_OF0}\ p_2\ (fs\ \mathsf{DELETE}\ f_1)$, it must be the case that f_1 appears in p_2, hence we can extract f_1 out of p_2 to split p_2 as a disjunction $f_1 \vee t$, using the following lemma proved by induction on $\mathsf{DISJ\_OF0}$:

Lemma 7 (`DISJ_OF0_split`).

$$\begin{array}{l} \vdash \mathsf{DISJ\_OF0}\ f\ fs \ \Rightarrow \\ \quad \forall t. \\ \qquad t \in fs \ \wedge\ \neg\mathsf{DISJ\_OF0}\ f\ (fs\ \mathsf{DELETE}\ t) \ \Rightarrow \\ \qquad \exists p.\ \mathsf{DISJ\_OF}\ p\ (fs\ \mathsf{DELETE}\ t) \ \wedge\ \mathsf{equiv0}\ \mu\ f\ (\mathsf{DISJ}\ t\ p) \end{array}$$

Applying the lemma above allows us to finish the proof by using the inductive hypothesis. □

3.3.5 Case for negation

Let us consider our last case, which amounts to prove:

Lemma 8 (`IBC_DNF_EXISTS_case3`).

$$\vdash \mathsf{DNF\_OF}\ p\ fs \ \wedge\ \mathsf{FINITE}\ fs \ \wedge\ fs \neq \emptyset \ \Rightarrow\ \exists f.\ \mathsf{DNF\_OF}\ f\ fs \ \wedge\ \mathsf{equiv0}\ \mu\ (\neg p)\ f$$

The idea of this proof is to use 'complement'. Consider a disjunction normal form p on a finite set fs, we find a formula of disjunction normal form f which is its negation, that is, we require f to be satisfied if and only if $\neg p$ is satisfied. As p is a disjunction, $\neg p$ is satisfied once none of p's conjuncts is satisfied. As a disjunction normal form, all of p's disjuncts are taken from the set of all of the $\mathsf{CONJ\_OF}$ formulas on fs, and these disjuncts form a subset ss of $\{\ c \mid \mathsf{CONJ\_OF}\ c\ fs\ \}$. Since none of these conjunctions is satisfied, it must be the case that the satisfied conjunction lives in $\{\ c \mid \mathsf{CONJ\_OF}\ c\ fs\ \}\ \mathsf{DIFF}\ ss$. Again, let $fs = \{f_1, f_2\}$ and $p := (f_1 \wedge f_2) \vee (\neg f_1 \wedge f_2)$ is in disjunction normal form. For $\neg p$ is satisfied, neither $f_1 \wedge f_2$ nor $f_1 \wedge \neg f_2$ is satisfied, so it must be the case that either $\neg f_1 \wedge \neg f_2$ or $\neg f_1 \wedge f_2$. We take the disjunction of the elements in the set $\{\neg f_1 \wedge \neg f_2, \neg f_1 \wedge f_2\}$ formed by taking out the disjuncts appear in p from the total set $\{f_1 \wedge f_2, \neg f_1 \wedge f_2, \neg f_1 \wedge \neg f_2, \neg f_1 \wedge f_2\}$, it gives the formula $(\neg f_1 \wedge \neg f_2) \vee (\neg f_1 \wedge f_2)$, which is equivalent to the negation of p.

Although it may be possible to stick to using our former tools `lit_list_to_form` and `list_list_to_-form2`, it is much more natural to directly use the completement of set to deal with the issue. Again, for a finite set fs, we will use literals to encode formulas in fs or its negation. But instead of using list of literals, we will use sets of literals this time.

We define satisfication of a literal as:

Definition 40 (Satisfication of a literal).

$$\mathsf{lsatis}\ \mathfrak{M}\ w\ (f, b) \stackrel{\mathsf{def}}{=} (\mathsf{satis}\ \mathfrak{M}\ w\ f \iff b)$$

That is, for a model $\mathfrak{M}$ and a world w, a literal $(f, \top)$ is satisfied at w if $\mathsf{satis}\ \mathfrak{M}\ w\ f$, and $(f, \bot)$ is satisfied at w if $\mathsf{satis}\ \mathfrak{M}\ w\ (\neg f)$. We want to construct $\mathsf{CONJ\_OF}$ formulas from well-formed sets of literals, such a well-formed set is called an 'lset':

Definition 41 (Well-formed literal set).

$$\mathsf{is\_lset}\ c\ fs \stackrel{\mathsf{def}}{=} \mathsf{FINITE}\ fs\ \wedge\ \mathsf{FINITE}\ c\ \wedge\ \mathsf{CARD}\ c\ =\ \mathsf{CARD}\ fs\ \wedge\ \mathsf{IMAGE\ FST}\ c\ =\ fs$$

The condition of being an `lset` corresponds the condition on an interesting list as in the story about the base case. An `lset` corresponds a conjunction, for instance, $ls := \{(f_1, \top), (f_2, \top)\}$ corresponds to the formula $f_1 \wedge f_2$. We care about when all the literals in the set are satisfied, and call a set c-satisfied for this situation.

Definition 42 (c-satisfication).

$$\mathsf{csatis}\ \mathfrak{M}\ w\ c \stackrel{\mathsf{def}}{=} \forall\, l.\ l\ \in\ c\ \Rightarrow\ \mathsf{lsatis}\ \mathfrak{M}\ w\ l$$

If two `lset`s are different, the union of them must contains a literal of opposite sign, hence the union of two distinct `lsets` can never be c-satisfied:

Proposition 42 (`NEQ_lsets_FALSE`).

$$\begin{aligned}&\vdash \mathsf{is\_lset}\ c_1\ fs\ \wedge\ \mathsf{is\_lset}\ c_2\ fs\ \wedge\ c_1\ \neq\ c_2\ \Rightarrow\\ &\quad \forall\, \mathfrak{M}\ w.\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\ \neg\mathsf{csatis}\ \mathfrak{M}\ w\ (c_1\ \cup\ c_2)\end{aligned}$$

By induction on $\mathsf{CONJ\_OF}$ and the finiteness of `lset` respectively, we prove for any `lset`, it is c-satisfied at a world in a model iff its corresponding $\mathsf{CONJ\_OF}$ formula is satisfied at the same point. Conversely, for any $\mathsf{CONJ\_OF}$ formula, there is an `lset` that corresponds to it. The follows are the set version of previous theorems about lists:

Proposition 43 (`CONJ_OF_lset`).

$$\begin{aligned}&\vdash \mathsf{CONJ\_OF}\ c\ fs\ \Rightarrow\\ &\quad \exists\, ls.\\ &\qquad \mathsf{is\_lset}\ ls\ fs\ \wedge\\ &\qquad \forall\, \mathfrak{M}\ w.\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\ (\mathsf{csatis}\ \mathfrak{M}\ w\ ls\ \iff\ \mathsf{satis}\ \mathfrak{M}\ w\ c)\end{aligned}$$

Proposition 44 (`is_lset_CONJ_OF_EXISTS`).

$$\begin{aligned}&\vdash \mathsf{is\_lset}\ c\ fs\ \wedge\ c\ \neq\ \emptyset\ \Rightarrow\\ &\quad \exists\, f.\\ &\qquad (\forall\, \mathfrak{M}\ w.\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\ (\mathsf{satis}\ \mathfrak{M}\ w\ f\ \iff\ \mathsf{csatis}\ \mathfrak{M}\ w\ c))\ \wedge\\ &\qquad \mathsf{CONJ\_OF}\ f\ fs\end{aligned}$$

A disjunction of $\mathsf{CONJ\_OF}$ formula is captured by sets of literal sets. For instance, the set $\{\{(f_1, \top), (f_2, \top)\}, \{(f_1, \bot), (f_2, \top)$
encodes the formula $(f_1 \wedge f_2) \vee (\neg f_1 \wedge f_2)$ of disjunction normal form on the set $\{f_1, f_2\}$. And the satisfication of $\mathsf{DISJ\_OF0}$ formulas is captured by d-satisfication:

Definition 43 (d-satisfication)**.**

$$\mathsf{dsatis}\ \mathfrak{M}\ w\ cs \overset{\mathsf{def}}{=} \exists\, c.\ c \in cs \wedge \mathsf{csatis}\ \mathfrak{M}\ w\ c$$

The set-version of the the previous proposition `ALL_POSSIBLE_VALUE_TRUE` looks like:

Proposition 45 (`dsatis_ALL_POSSIBLE_VALUE`)**.**

$$\begin{aligned}
&\vdash \mathsf{FINITE}\ fs \Rightarrow \\
&\quad fs \neq \emptyset \Rightarrow \\
&\quad \forall \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \mathsf{dsatis}\ \mathfrak{M}\ w\ \{\ c\ /\ \mathsf{is\_lset}\ c\ fs\ \}
\end{aligned}$$

Using this as a lemma, we can prove the a critical statement about our idea of 'complement', saying that a set of `lset`s are d-satisfied iff its complement in the 'total set' of all the `lset`s is not d-satisfied.

Proposition 46 (`dsatis_is_lset_complement`)**.**

$$\begin{aligned}
&\vdash \mathsf{FINITE}\ fs \wedge fs \neq \emptyset \wedge (\forall\, c.\ c \in cs \Rightarrow \mathsf{is\_lset}\ c\ fs) \Rightarrow \\
&\quad \forall \mathfrak{M}\ w. \\
&\qquad w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \\
&\qquad (\mathsf{dsatis}\ \mathfrak{M}\ w\ cs \iff \neg\mathsf{dsatis}\ \mathfrak{M}\ w\ (\{\ c\ /\ \mathsf{is\_lset}\ c\ fs\ \}\ \mathsf{DIFF}\ cs))
\end{aligned}$$

Proof. For the implication from left to right, suppose both $\mathsf{dsatis}\ \mathfrak{M}\ w\ cs$ and $\mathsf{dsatis}\ \mathfrak{M}\ w\ (\{\ c \mid \mathsf{is\_lset}\ c\ fs\ \}\ \mathsf{DIFF}\ cs)$, then by definition of dsatis, it means two distinct `lset`s are c-satisfied at the same point, which contradicts `NEQ_lsets_FALSE`. For the implication from right to left, suppose $\neg\mathsf{dsatis}\ \mathfrak{M}\ w\ cs$, by `dsatis_ALL_-POSSIBLE_VALUE_TRUE`, we will be able to find the `lset` which holds on w in $\{\ c \mid \mathsf{is\_lset}\ c\ fs\ \}\ \mathsf{DIFF}\ cs$. This completes the proof. □

The correspondence of set of `lset`s and $\mathsf{DISJ\_OF0}$ formulas is given as:

Proposition 47 (`DISJ_OF0_cset`)**.**

$$\begin{aligned}
&\vdash \mathsf{DISJ\_OF0}\ d\ fs \Rightarrow \\
&\quad \forall fs_0. \\
&\qquad fs \subseteq \{\ c\ /\ \mathsf{CONJ\_OF}\ c\ fs_0\ \} \Rightarrow \\
&\qquad \exists\, cs. \\
&\qquad\quad (\forall\, c.\ c \in cs \Rightarrow \mathsf{is\_lset}\ c\ fs_0) \wedge \\
&\qquad\quad \forall \mathfrak{M}\ w. \\
&\qquad\qquad w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ d \iff \exists\, c.\ c \in cs \wedge \mathsf{csatis}\ \mathfrak{M}\ w\ c)
\end{aligned}$$

Altogether, here comes the correspondence results of set of `lset`s and $\mathsf{DNF\_OF}$ formulas. The first one, which is a immediate consequence of `DISJ_OF0_cset`, says each $\mathsf{DNF\_OF}$ formula corresponds a set of `lset`s.

Proposition 48 (`DNF_OF_set`).

$$\begin{aligned}
&\vdash \mathsf{DNF\_OF}\ d\ fs \Rightarrow \\
&\quad \exists\, cs. \\
&\qquad (\forall\, c.\ c \in cs \Rightarrow \mathsf{is\_lset}\ c\ fs) \wedge \\
&\qquad \forall\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame}.\mathit{world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ d \iff \exists\, c.\ c \in cs \wedge \mathsf{csatis}\ \mathfrak{M}\ w\ c)
\end{aligned}$$

The second one says each set of `lset`s gives a $\mathsf{DNF\_OF}$ formula:

Proposition 49 (`is_lset_DNF_OF_EXISTS`).

$$\begin{aligned}
&\vdash \mathsf{FINITE}\ s \Rightarrow \\
&\quad \forall\, fs. \\
&\qquad \mathsf{FINITE}\ fs \wedge fs \neq \emptyset \Rightarrow \\
&\qquad (\forall\, c.\ c \in s \Rightarrow \mathsf{is\_lset}\ c\ fs) \Rightarrow \\
&\qquad \exists\, f. \\
&\qquad\quad (\forall\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame}.\mathit{world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ f \iff \mathsf{dsatis}\ \mathfrak{M}\ w\ s)) \wedge \\
&\qquad\quad \mathsf{DNF\_OF}\ f\ fs
\end{aligned}$$

Proof. The proof is by induction on $\mathsf{FINITE}\ s$. For the base case that $s = \emptyset$, the required formula f is obviously $\bot$. Suppose for s a finite set of `lset`s, there exists a formula f in disjunction normal form such that $\mathsf{satis}\ \mathfrak{M}\ w\ f \iff \mathsf{dsatis}\ \mathfrak{M}\ w\ s$ for any model $\mathfrak{M}$ and any $w \in \mathfrak{M}.\mathsf{frame}.\mathsf{world}$, we find a formula g such that $\mathsf{satis}\ \mathfrak{M}\ w\ g \iff \mathsf{dsatis}\ \mathfrak{M}\ w\ (e\ \mathsf{INSERT}\ s)$. By `is_lset_CONJ_OF_EXISTS`, there is a $\mathsf{CONJ\_OF}$ formula f' on fs with the property that $\mathsf{satis}\ \mathfrak{M}\ w\ f' \iff \mathsf{csatis}\ \mathfrak{M}\ w\ e$ for arbitary $\mathfrak{M}$. If f is equivalent to $\bot$, then f' is the g we want, and vice versa. If both of them are not equivalent to $\bot$, we can take $f' \vee f$ as the g we want. The first desired condition is easy to check, it remains to prove $f' \vee f$ is in disjunction normal form, which is suffices by proving $\mathsf{DISJ\_OF0}\ f\ (\{\ c \mid \mathsf{CONJ\_OF}\ c\ fs\ \}\ \mathsf{DELETE}\ f')$. Suppose not, then there exists a p such that $\mathsf{DISJ\_OF0}\ p\ (\{\ c \mid \mathsf{CONJ\_OF}\ c\ fs\ \}\ \mathsf{DELETE}\ f')$ and $\mathsf{equiv0}\ \mu\ f\ (\mathsf{DISJ}\ f'\ p)$ by `DISJ_OF0_split`. To obtain a contradiction, we find a model with a world satisfies f' but not f. With the equivalence conditions on f and f', it amounts to find a modal $\mathfrak{M}$ with a world $w \in \mathfrak{M}.\mathsf{frame}.\mathsf{world}$ such that $\mathsf{csatis}\ \mathfrak{M}\ w\ e$ but $\neg\mathsf{dsatis}\ \mathfrak{M}\ w\ s$. The claim is any world c-satisfying e cannot d-satisfy s, otherwise it contradicts `dsatis_is_lset_completment`. Also since f' is not equivalent to $\bot$, a model satisfying f' exists, so we are done. □

Now we have all the ingredients to prove our last case.

Proof. As $\mathsf{DNF\_OF}\ p\ fs$, by `DNF_OF_cset` we obtain a `lset` cs such that $\mathsf{satis}\ \mathfrak{M}\ w\ p \iff \mathsf{dsatis}\ \mathfrak{M}\ w\ cs$ for any $\mathfrak{M}$ and w. Consider its complement $\{\ c \mid \mathsf{is\_lset}\ c\ fs\ \}\ \mathsf{DIFF}\ cs$, by `is_lset_DNF_OF_EXISTS` we obtain a formula f such that $\mathsf{DNF\_OF}\ f\ fs$ and $\mathsf{satis}\ \mathfrak{M}\ w\ f$ iff $\mathsf{dsatis}\ \mathfrak{M}\ w\ (\{\ c \mid \mathsf{is\_lset}\ c\ fs\ \}\ \mathsf{DIFF}\ cs)$ for any $\mathfrak{M}$ and w. By `dsatis_is_lset_complement`, $\mathsf{dsatis}\ \mathfrak{M}\ w\ (\{\ c \mid \mathsf{is\_lset}\ c\ fs\ \}\ \mathsf{DIFF}\ cs)$ iff $\neg\mathsf{dsatis}\ \mathfrak{M}\ w\ cs$, so f is equivalent to $\neg p$. □

Here we are done with the proof of `prop_2_29_strengthen`. This is the end of the interlude.

This is time to return to discussion about the proof of finite model property. Recall in the last chapter, we have seen that a bisimulation gives arise to modal equivalence, modal equivalence is 'satisfiying exactly the same formulas', but when we are building a finite model for a formula ϕ, we do not care about the satisfication of any formula of degree above DEG phi, since such formula cannot affect the satisfication of ϕ. Therefore, a finite approximation of bisimulation is enough, a finite approximation of depth n is called an n-bisimulation. Let $w \in \mathfrak{M}$.frame.world and $w' \in \mathfrak{M}'$.frame.world, w and w' are n-bisimilar if there exists a sequence of relations $Z_n \subseteq \cdots \subseteq Z_0$ such that:

- w and w' are related by Z_n
- If $v \in \mathfrak{M}$.frame.world and $v' \in \mathfrak{M}'$.frame.world are related by Z_0, then v and v' satisfy the same propositional letters.
- If $v \in \mathfrak{M}$.frame.world and $v' \in \mathfrak{M}'$.frame.world are related by Z_{i+1} and we have $\mathfrak{M}$.frame.rel v u for $u \in \mathfrak{M}$.frame.world, then there exists $u' \in \mathfrak{M}'$.frame.world such that $\mathfrak{M}'$.frame.rel v' u' with u and u' related by Z_i.
- If $v \in \mathfrak{M}$.frame.world and $v' \in \mathfrak{M}'$.frame.world are related by Z_{i+1} and we have $\mathfrak{M}'$.frame.rel v' u' for $u' \in \mathfrak{M}$.frame.world, then there exists $u \in \mathfrak{M}$.frame.world such that $\mathfrak{M}$.frame.rel v u with u and u' related by Z_i.

Such a sequence of Z_i is a family of relations indexed by natural numbers. When the world set of $\mathfrak{M}$ has type β and the world set of $\mathfrak{M}'$ has type γ, we encode such a family using functions $f : num \to \beta \to \gamma \to bool$. Such a function assigns each natural number a relation, so the f i is the relation Z_i in the usual mathematical definition, and nbisim $\mathfrak{M}$ $\mathfrak{M}'$ f n w w' means w and w' are worlds in $\mathfrak{M}$ and $\mathfrak{M}'$ respectively which are n-bisimilar via the family of relations given by f.

Definition 44 (n-bisimulation)**.**

$$
\begin{aligned}
&\mathsf{nbisim}\ \mathfrak{M}\ \mathfrak{M}'\ f\ n\ w\ w' \overset{\mathsf{def}}{=} \\
&\quad w \in \mathfrak{M}.\mathit{frame.world} \land w' \in \mathfrak{M}'.\mathit{frame.world} \land \\
&\quad (\forall\, m\ a\ b. \\
&\qquad a \in \mathfrak{M}.\mathit{frame.world} \land b \in \mathfrak{M}'.\mathit{frame.world} \Rightarrow \\
&\qquad m + 1 \le n \Rightarrow \\
&\qquad f\ (m + 1)\ a\ b \Rightarrow \\
&\qquad f\ m\ a\ b) \land f\ n\ w\ w' \land \\
&\quad (\forall\, v\ v'. \\
&\qquad v \in \mathfrak{M}.\mathit{frame.world} \land v' \in \mathfrak{M}'.\mathit{frame.world} \Rightarrow \\
&\qquad f\ 0\ v\ v' \Rightarrow \\
&\qquad \forall\, p.\ \mathsf{satis}\ \mathfrak{M}\ v\ (\mathsf{VAR}\ p) \iff \mathsf{satis}\ \mathfrak{M}'\ v'\ (\mathsf{VAR}\ p)) \land \\
&\quad (\forall\, v\ v'\ u\ i. \\
&\qquad i + 1 \le n \land v \in \mathfrak{M}.\mathit{frame.world} \land v' \in \mathfrak{M}'.\mathit{frame.world} \land u \in \mathfrak{M}.\mathit{frame.world} \land \\
&\qquad \mathfrak{M}.\mathit{frame.rel}\ v\ u \land f\ (i + 1)\ v\ v' \Rightarrow \\
&\qquad \exists\, u'.\ u' \in \mathfrak{M}'.\mathit{frame.world} \land \mathfrak{M}'.\mathit{frame.rel}\ v'\ u' \land f\ i\ u\ u') \land \\
&\quad \forall\, v\ v'\ u'\ i. \\
&\qquad i + 1 \le n \land v \in \mathfrak{M}.\mathit{frame.world} \land v' \in \mathfrak{M}'.\mathit{frame.world} \land u' \in \mathfrak{M}'.\mathit{frame.world} \land \\
&\qquad \mathfrak{M}'.\mathit{frame.rel}\ v'\ u' \land f\ (i + 1)\ v\ v' \Rightarrow \\
&\qquad \exists\, u.\ u \in \mathfrak{M}.\mathit{frame.world} \land \mathfrak{M}.\mathit{frame.rel}\ v\ u \land f\ i\ u\ u'
\end{aligned}
$$

By induction on n, we see if two worlds $w \in \mathfrak{M}.\mathsf{frame.world}$ and $w' \in \mathfrak{M}'.\mathsf{frame.world}$ are related by an n-bisimulation, then they agree on all modal formulas up to degree n:

Proposition 50 (`prop_2_31_half1`)**.**

$$
\begin{aligned}
&\vdash (\exists\, f.\ \mathsf{nbisim}\ \mathfrak{M}\ \mathfrak{M}'\ f\ n\ w\ w') \Rightarrow \\
&\quad \forall\, \mathit{phi}.\ \mathsf{DEG}\ \mathit{phi} \le n \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi} \iff \mathsf{satis}\ \mathfrak{M}'\ w'\ \mathit{phi})
\end{aligned}
$$

The converse of the above holds when our modal language is defined on a finite set of propositional letters and the type of world sets of our models are infinite. It can be proved by an argument analogue to the proof of Hennessy-Milner theorem, with $\lambda\, n\ n_1\ n_2.\ \forall\, \mathit{phi}.\ \mathsf{DEG}\ \mathit{phi} \le n \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ n_1\ \mathit{phi} \iff \mathsf{satis}\ \mathfrak{M}'\ n_2\ \mathit{phi})$ gives an n-bisimulation relation linking w and w':

Proposition 51 (`prop_2_31_half2`)**.**

$$
\begin{aligned}
&\vdash \mathsf{INFINITE}\ \mathcal{U}(:\beta) \land \mathsf{INFINITE}\ \mathcal{U}(:\gamma) \land \mathsf{FINITE}\ \mathcal{U}(:\alpha) \land w \in \mathfrak{M}.\mathit{frame.world} \land \\
&\quad w' \in \mathfrak{M}'.\mathit{frame.world} \Rightarrow \\
&\quad (\forall\, \mathit{phi}.\ \mathsf{DEG}\ \mathit{phi} \le n \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi} \iff \mathsf{satis}\ \mathfrak{M}'\ w'\ \mathit{phi})) \Rightarrow \\
&\quad \exists\, f.\ \mathsf{nbisim}\ \mathfrak{M}\ \mathfrak{M}'\ f\ n\ w\ w'
\end{aligned}
$$

DEG is a concept that measure the depth of a formula, we also want a concept that measure the depth of a model, as 'depth' is a relative concept measuring the distance of two points. To talk about the depth of a world $w \in \mathfrak{M}.\mathsf{frame.world}$, we need $\mathfrak{M}$ to be naturally equipped with a base point, so the 'height' of a world only makes sense to rooted model. To tell the HOL about this definition, we start by defining heightLE:

Definition 45 (Upper bound of height)**.**

$$\frac{}{\mathsf{heightLE}\ \mathfrak{M}\ x\ \mathfrak{M}'\ x\ n}$$

$$\frac{v \in \mathfrak{M}.\mathit{frame.world} \quad \exists\, w.\ w \in \mathfrak{M}.\mathit{frame.world} \land \mathfrak{M}.\mathit{frame.rel}\ w\ v \land \mathsf{heightLE}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w\ n}{\mathsf{heightLE}\ \mathfrak{M}\ x\ \mathfrak{M}'\ v\ (n + 1)}$$

Recall how we defined rooted model: rooted_model $\mathfrak{M}\ x\ \mathfrak{M}'$ means '$\mathfrak{M}$ is a rooted model generated by the world x in the ambient model $\mathfrak{M}'$'. As heightLE is designed to be only make sense for rooted models, we encode the information about the rootedness of the model we are talking about into this definition. heightLE $\mathfrak{M}\ x\ \mathfrak{M}'\ w\ n$ reads 'for the rooted model $\mathfrak{M}$ generated by the root x in $\mathfrak{M}'$, the distance from the world w to the root x is less or equal to n', and we will always have an assumption on rootedness of $\mathfrak{M}$ whence this definition is involved. The height of a world w is the smallest natural number n such that heightLE $\mathfrak{M}\ x\ \mathfrak{M}'\ w\ n$. And a height of model is the maximum n such that there is a world of height n in the model.

Definition 46 (Height of a world)**.**

$$\mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w \overset{\mathsf{def}}{=} \mathsf{MIN\_SET}\ \{\, n \ /\ \mathsf{heightLE}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w\ n \,\}$$

Definition 47 (Height of a model)**.**

$$\mathsf{model\_height}\ \mathfrak{M}\ x\ \mathfrak{M}' \overset{\mathsf{def}}{=} \mathsf{MAX\_SET}\ \{\, n \ /\ (\exists\, w.\ w \in \mathfrak{M}.\mathit{frame.world} \land \mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w = n) \,\}$$

Obviously, the root of any rooted model have height 0. We are particularly interested in talking about heights in tree-like model. When $\mathfrak{M}$ is tree-like, if $w \in \mathfrak{M}.\mathsf{frame.world}$ has height n, then any world $w' \in \mathfrak{M}.\mathsf{frame.world}$ such that $\mathfrak{M}.\mathsf{frame.rel}\ w\ w'$ will have height $n+1$, this is proved using the induction principle `RTC_INDUCT_RIGHT1`:

Lemma 9 (`tree_height_rel_lemma`)**.**

$$\begin{array}{l}
\vdash \mathsf{tree}\ \mathfrak{M}.\mathit{frame}\ x \Rightarrow \\
\quad \forall\, w. \\
\qquad w \in \mathfrak{M}.\mathit{frame.world} \land \mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}\ w = n \Rightarrow \\
\qquad \forall\, v.\ \mathfrak{M}.\mathit{frame.rel}\ w\ v \land v \in \mathfrak{M}.\mathit{frame.world} \Rightarrow \mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}\ v = n + 1
\end{array}$$

The restriction of a rooted model $\mathfrak{M}$ to the height k is the submodel consisting of all the worlds in $\mathfrak{M}$ of height up to k. The restriction of a tree-like model is again a tree-like model:

Definition 48 (Restriction a model up to some height)**.**

$$\begin{array}{l}
\mathsf{hrestriction}\ \mathfrak{M}\ x\ \mathfrak{M}'\ n \overset{\mathsf{def}}{=} \\
\quad \mathsf{<|}\mathit{frame} := \\
\qquad \mathsf{<|}\mathit{world} := \{\, w \ /\ w \in \mathfrak{M}.\mathit{frame.world} \land \mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w \leq n \,\}; \\
\qquad \mathit{rel} := (\lambda\, n_1\ n_2.\ \mathfrak{M}.\mathit{frame.rel}\ n_1\ n_2)\mathsf{|>};\ \mathit{valt} := (\lambda\, phi\ n.\ \mathfrak{M}.\mathit{valt}\ phi\ n)\mathsf{|>}
\end{array}$$

Proposition 52 (`tree_hrestriction_tree`).

$$\vdash \mathsf{tree}\ \mathfrak{M}.\mathit{frame}\ x\ \Rightarrow\ \forall n.\ \mathsf{tree}\ (\mathsf{hrestriction}\ \mathfrak{M}\ x\ \mathfrak{M}\ n).\mathit{frame}\ x$$

Restriction of rooted model gives arise of n-bisimulation: If we restrict a rooted model $\mathfrak{M}$ to height k, then a world w of height m in the restricted model is $k-m$-bisimilar to itself in the original model:

Lemma 10 (`lemma_2_33`).

$$\begin{aligned}&\vdash \mathsf{rooted\_model}\ \mathfrak{M}\ x\ \mathfrak{M}'\ \Rightarrow\\ &\quad \forall w.\\ &\qquad w\ \in\ (\mathsf{hrestriction}\ \mathfrak{M}\ x\ \mathfrak{M}'\ k).\mathit{frame}.\mathit{world}\ \Rightarrow\\ &\qquad \exists f.\ \mathsf{nbisim}\ (\mathsf{hrestriction}\ \mathfrak{M}\ x\ \mathfrak{M}'\ k)\ \mathfrak{M}\ f\ (k\ -\ \mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w)\ w\ w\end{aligned}$$

Proof. The n-bisimilar relation is give as $\lambda\, n\ w_1\ w_2.\ w_1\ =\ w_2\ \wedge\ \mathsf{height}\ \mathfrak{M}\ x\ \mathfrak{M}'\ w_1\ \leq\ k\ -\ n$. □

Now we can start building a finite model via selection:

Theorem 8 (`thm_2_34`).

$$\begin{aligned}&\vdash \mathsf{satis}\ \mathfrak{M}_1\ w_1\ \mathit{phi}\ \Rightarrow\\ &\quad \exists\, FM\ v.\ \mathsf{FINITE}\ FM.\mathit{frame}.\mathit{world}\ \wedge\ v\ \in\ FM.\mathit{frame}.\mathit{world}\ \wedge\ \mathsf{satis}\ FM\ v\ \mathit{phi}\end{aligned}$$

Proof: Suppose $\mathsf{satis}\ \mathfrak{M}_1\ w_1\ \mathit{phi}$ where $w_1 : \beta, \phi : \alpha\ form$, then by `prop_2_15_corollary`, there exists a tree-like model $\mathfrak{M}_2$ with phi satisfied at its root w_2. Such an $\mathfrak{M}_2$ obtained from `prop_2_15_corollary` has its world set of type $\beta\ list$, so all the lemmas proved before with a infinite universe assumption applies for any model with its world set a subset of $\mathfrak{M}_2$.frame.world . Define M_3 :=$\mathsf{hrestriction}\ \mathfrak{M}_2\ w_2\ \mathfrak{M}_2\ k$ to be the restriction of $\mathfrak{M}_2$ to height k, then M_3 is rooted and there is a $\mathsf{nbisim}\ M_3\ \mathfrak{M}_2\ f\ k\ w_2\ w_2$ by `lemma_-2_33`. Hence $\mathsf{satis}\ M_3\ w_2\ \mathit{phi}$ by `prop_2_31_half1`. By `exercise_1_3_1` proved in the first chapter, if a propositional letter does not appear in ϕ, then it has no effect to the satisfication of ϕ, so we can discard all the propositional letters in M_3 which does not occur in ϕ to obtain the model $M_3' =$
<|frame := <|world := M_3.frame.world; rel := M_3.frame.rel|>;
valt := $(\lambda\, p\ v.$ `if` VAR $p\ \in$ subforms phi `then` M_3.valt $p\ v$ `else` F)|>, and still have $\mathsf{satis}\ M_3'\ w_2\ \mathit{phi}$. We select a finite model inductively from M_3'.

Let s denote the set of propositional letters used by ϕ, so s is finite. By `prop_2_29_strengthen`, there are only finitely many non-equivalent formulas of degree less or equal to k which only use propositional letters in s, that is, the set $\mathit{distfp} = \{\ f\ \mid\ \mathsf{DEG}\ f\ \leq\ k\ \wedge\ \forall a.\ \mathsf{VAR}\ a\ \in\ \mathsf{subforms}\ f\ \Rightarrow\ a\ \in\ s\ \} // E\ \mu$ is a finite. We care about the equivalence classes in distfp which are equivalence classes of formulas starting with a $\Diamond$. For such a equivalence class, taking the intersection with the set $\{\ d\ \mid\ \exists\, d_0.\ d\ =\ \Diamond\ d_0\ \}$ does not give the empty set. Take the image of distfp under the function $\lambda\, s.\ s\ \cap\ \{\ d\ \mid\ \exists\, d_0.\ d\ =\ \Diamond\ d_0\ \}$ and delete the empty set from the image, we obtain a set fs of sets of formulas, where for each $x \in fs$, x consists of equivalent formulas of degree less or equal to k, only use propositional letters in s, and starts with diamond. Hence $rs = \mathsf{IMAGE\ CHOICE}\ (\mathsf{IMAGE}\ (\lambda\, s.\ s\ \cap\ \{\ d\ \mid\ \exists\, d_0.\ d\ =\ \Diamond\ d_0\ \})\ \mathit{distfp}\ \mathsf{DELETE}\ \emptyset)$ can be taken as the set of representatives of formulas with desired properties, it is finite as an image of a finite set.

We will construct sets $S_0, \cdots S_k$ of worlds in M_3', where the points in S_n have height n. Start with $S_0 := \{w_2\}$, and inductively, assume $S_0, \cdots S_n$ has been defined, construct S_{n+1} as follows: Consider an element in $v \in S_n$, for each $\Diamond\phi \in rs$ such that $\mathsf{satis}\ M_3'\ w_2\ (\Diamond\ phi)$, pick a world $u \in M_3'.\mathsf{frame.world}$ such that $M_3'.\mathsf{frame.rel}\ v\ u$ and $\mathsf{satis}\ M_3'\ u\ phi$. Do the same thing to all the $v \in S_n$, then S_{n+1} is the set of all the such u's which are seleted in this way.

The way we taken to formalise the definition of such S_i's is to define a primitive recursive function:

```
PRIM_REC $\{ w_2 \}$
 $(\lambda\, s_0\ n.$
   $\{$ CHOICE $uset$ |
   $\exists\, phi\ v.$
     satis $M_3'\ v\ (\Diamond\ phi)\ \wedge$
     $\Diamond\ phi\ \in$ IMAGE CHOICE (IMAGE $(\lambda\, s.\ s\ \cap\ \{\ d$ | $\exists\, d_0.\ d\ =\ \Diamond\ d_0\ \})$ $distfp$ DELETE $\emptyset)\ \wedge$
     $v\ \in\ s_0\ \wedge\ uset\ =\ \{\ u$ | $M_3'.$frame.rel $v\ u\ \wedge\ u\ \in\ M_3'.$frame.world $\wedge$ satis $M_3'\ u\ phi\ \}\ \})$
```

For each $i \leq k$, $ss\ i$ will be our S_i. By induction on i, we can prove each $ss\ i$ is finite, so the set $W4 := \bigcup_i \{ss\ i \mid i \leq k\}$ is finite. The resultant finite model we select is: $M_4\ =$
<|frame := <|world := W_4; rel := M_3.frame.rel|>; valt := M_3'.valt|>.

To prove $\mathsf{satis}\ M_4\ w_2\ phi$, it suffices to give a k-bisimulation between M_4 and M_3' relating $w2$ to itself. Such a k-bisimulation is: $\lambda\, n\ a_1\ a_2.$
$a_1 \in M_4.\mathsf{frame.world} \wedge a_2 \in M_3'.\mathsf{frame.world} \wedge \mathsf{height}\ M_3'\ w_2\ M_3'\ a_1 = \mathsf{height}\ M_3'\ w_2\ M_3'\ a_2\ \wedge$
$\mathsf{height}\ M_3'\ w_2\ M_3'\ a_1 \leq k - n\ \wedge$
$\forall\, phi.\ \mathsf{DEG}\ phi \leq n \wedge (\forall\, a.\ \mathsf{VAR}\ a \in \mathsf{subforms}\ phi \Rightarrow a \in s) \Rightarrow (\mathsf{satis}\ M_3'\ a_1\ phi \iff \mathsf{satis}\ M_3'\ a_2\ phi)$ The rest of the proof amounts to check the above indeed gives a k-bisimulation, the proof is not hard using a similar argument as we proved the Hennessy-Milner theorem.

As we took a detour through `prop_2_15_corollary`, this construction of finite model changes the type of model.

4 Reaching out to the world of first order logic

As Slogan 3 in the introduction claims, modal logic is not an isolated formal system, in this chapter, we start linking modal logic with the wider logical world by discussing about the relation to first order logic. In the first half of the chapter, we define standard translation as our link between modal logic and first order logic, and in the second half of the chapter, with the help of standard translation, we introduce another construction on models which will give modal equivalence, and lead to an elegant result about bisimulation: modal equivalence implies bisimilarity-somewhere-else.

4.1 Standard Translation

To discuss the relationship between modal logic and first order logic, firstly we need to build some basics of first order logic in the HOL. First order logic is formalised in HOL light in 1998 as in (reference), we take our construction of first-order model theory as in the paper.

For a first order language, a term is either a variable letter x or a function symbol f applied on a list of terms, which looks like $f(t_1, \cdots, t_n)$, where the t's can either be variable letters or itself a function applied on some terms. To avoid specifying the type every where, we restrict our scope to countable language, by using only natural numbers to denote variable and function symbols:

Definition 49 (First order terms)**.**

$$\mathit{term} \;=\; \mathsf{fV}\ \mathit{num}\ /\ \mathsf{Fn}\ \mathit{num}\,(\mathit{term\ list})$$

Hence our terms will look like fV 6, Fn 1 [fV 1; fV 2], Fn 2 [Fn 0 []], etc. A constant is just a nullary function symbol.

Our formulas are defined inductively as well, using minimal amount of logical connectives, by choosing the falsity, atoms, implication and universal quantification as primitive, predicate symbols are also represented by natural numbers. In particular, an n-ary relation symbol is a predicate symbol that takes lists of length n:

Definition 50 (First order formulas)**.**

$$\phi \;=\; \mathsf{fFALSE}\ /\ \mathsf{Pred}\ \mathit{num}\,(\mathit{term\ list})\ /\ \mathsf{IMP}\ \phi\ \phi\ /\ \mathsf{FALL}\ \mathit{num}\ \phi$$

A quantified variable is called a bounded variable, otherwise it is called free. We define a function that returns the set of free variables of the formula, starting by collecting variables occur in the terms of formulas, and then delete the bounded ones. For bounded variable, since we just need to detect the occurance of quantifiers, we do not need to start with first order terms :

Definition 51 (Free variables)**.**

$$\begin{aligned}
&\mathsf{FVT}\ (\mathsf{fV}\ v) \overset{\mathsf{def}}{=} \{\ v\ \}\\
&\mathsf{FVT}\ (\mathsf{Fn}\ s\ ts) \overset{\mathsf{def}}{=} \mathsf{LIST\_UNION}\ (\mathsf{MAP}\ (\lambda\, a.\ \mathsf{FVT}\ a)\ ts)\\
\\
&\mathsf{FV}\ \mathsf{fFALSE} \overset{\mathsf{def}}{=} \emptyset\\
&\mathsf{FV}\ (\mathsf{Pred}\ v_0\ ts) \overset{\mathsf{def}}{=} \mathsf{LIST\_UNION}\ (\mathsf{MAP}\ \mathsf{FVT}\ ts)\\
&\mathsf{FV}\ (f_1\ \to\ f_2) \overset{\mathsf{def}}{=} \mathsf{FV}\ f_1\ \cup\ \mathsf{FV}\ f_2\\
&\mathsf{FV}\ (\mathsf{FALL}\ x\ f) \overset{\mathsf{def}}{=} \mathsf{FV}\ f\ \mathsf{DELETE}\ x
\end{aligned}$$

Definition 52 (Bounded variables)**.**

$$\begin{aligned}
&\mathsf{BV}\ \mathsf{fFALSE} \overset{\mathsf{def}}{=} \emptyset\\
&\mathsf{BV}\ (\mathsf{Pred}\ v_0\ v_1) \overset{\mathsf{def}}{=} \emptyset\\
&\mathsf{BV}\ (f_1\ \to\ f_2) \overset{\mathsf{def}}{=} \mathsf{BV}\ f_1\ \cup\ \mathsf{BV}\ f_2\\
&\mathsf{BV}\ (\mathsf{FALL}\ x\ f) \overset{\mathsf{def}}{=} x\ \mathsf{INSERT}\ \mathsf{BV}\ f
\end{aligned}$$

Here LIST_UNION is a function that takes a list of sets, and give us the union of all the sets in the list:

Definition 53 (Union of a list of sets).

$$\begin{aligned}\mathsf{LIST\_UNION}\ [] &\overset{\text{def}}{=} \emptyset\\ \mathsf{LIST\_UNION}\ (h :: t) &\overset{\text{def}}{=} h \cup \mathsf{LIST\_UNION}\ t\end{aligned}$$

Similarly, we have functions called `form_functions` and `form_predicates`, that take a formula and give the set of functions and predicates appear in the formula respectively.

The non-primitive connectives are defined in the canonical way:

Definition 54 (Non-primitive first-order connectives).

$$\begin{aligned}&\mathsf{fNOT}\ f \overset{\text{def}}{=} f \to \mathsf{fFALSE}\\ &\mathsf{True} \overset{\text{def}}{=} \mathsf{fNOT\ fFALSE}\\ &\mathsf{fDISJ}\ p\ q \overset{\text{def}}{=} (p \to q) \to q\\ &\mathsf{fAND}\ p\ q \overset{\text{def}}{=} \mathsf{fNOT}\ (\mathsf{fDISJ}\ (\mathsf{fNOT}\ p)\ (\mathsf{fNOT}\ q))\\ &\mathsf{fEXISTS}\ x\ p \overset{\text{def}}{=} \mathsf{fNOT}\ (\mathsf{FALL}\ x\ (\mathsf{fNOT}\ p))\end{aligned}$$

To interpret these formulas, we need models for first order logic. A first order model of type α is a triple consists of an α-set which is its domain, a interpretation of function symbols, and a interpretation of predicate symbols.

Definition 55 (First order model).

$$\begin{aligned}&\alpha\ \mathtt{model} = \mathsf{<|}\\ &\quad \mathit{Dom} : \alpha \to \mathtt{bool};\\ &\quad \mathit{Fun} : \mathtt{num} \to \alpha\ \mathtt{list} \to \alpha;\\ &\quad \mathit{Pred} : \mathtt{num} \to \alpha\ \mathtt{list} \to \mathtt{bool}\\ &\mathsf{|>}\end{aligned}$$

Given a first order model $\mathfrak{M}$, we can interpret formulas or terms by assigning each variable symbol an element in $\mathfrak{M}.\mathsf{Dom}$. Such an assignment is given by a valuation, which is a function from the universe of natural numbers to the domain of $\mathfrak{M}$:

Definition 56 (Valuation).

$$\mathsf{valuation}\ \mathfrak{M}\ v \overset{\text{def}}{=} \forall n.\ v\ n \in \mathfrak{M}.\mathit{Dom}$$

Interpretation of terms and formulas are given as termval and `feval`, where termval takes a model, an assignment of variable letters and a term, gives us an element of $\mathfrak{M}.\mathsf{Dom}$. And `feval` takes a model, an assignment of variable letters and a first order formula, gives us the truth value whether the formula we give holds on the model under the current assignment of variable letters, we only care about when the assignment of variable letters is indeed a valuation as defined above, when σ is a valuation and feval $\mathfrak{M}$ σ *fform*, we say '*fform* is satisfied in $\mathfrak{M}$ under the valuation σ'. In particular, if there is only one free variable x in $fform$, then feval $\mathfrak{M}$ σ *fform* just means that *fform* is satisfied at the point $\sigma\ x$ in the model $\mathfrak{M}$. The advantage of using the σ is that it can simutinously cope with any number of free variables.

Definition 57 (Term Valuation)**.**

$$\begin{aligned}
&\mathsf{termval}\ (\mathfrak{M} : \alpha\ \texttt{model})\ (v : \texttt{num} \to \alpha)\ (\mathsf{fV}\ (x : \texttt{num})) \overset{\text{def}}{=} v\ x\\
&\mathsf{termval}\ (\mathfrak{M} : \alpha\ \texttt{model})\ (v : \texttt{num} \to \alpha)\ (\mathsf{Fn}\ (f : \texttt{num})\ (l : \texttt{term list})) \overset{\text{def}}{=}\\
&\ \mathfrak{M}.\mathit{Fun}\, f\ (\mathsf{MAP}\ (\lambda\, (a : \texttt{term}).\ \mathsf{termval}\ \mathfrak{M}\ v\ a)\ l)
\end{aligned}$$

Definition 58 (Satisfication of first order formula)**.**

$$\mathsf{fsatis}\ \mathfrak{M}\ \sigma\ \mathit{fform} \overset{\text{def}}{=} \mathsf{valuation}\ \mathfrak{M}\ \sigma \land \mathsf{feval}\ \mathfrak{M}\ \sigma\ \mathit{fform}$$

$$\begin{aligned}
&\mathsf{feval}\ \mathfrak{M}\ v\ \mathsf{fFALSE} \overset{\text{def}}{=} \mathsf{F}\\
&\mathsf{feval}\ \mathfrak{M}\ v\ (\mathsf{Pred}\ a\ l) \overset{\text{def}}{=} \mathfrak{M}.\mathit{Pred}\ a\ (\mathsf{MAP}\ (\mathsf{termval}\ \mathfrak{M}\ v)\ l)\\
&\mathsf{feval}\ \mathfrak{M}\ v\ (f_1 \to f_2) \overset{\text{def}}{=} \mathsf{feval}\ \mathfrak{M}\ v\ f_1 \Rightarrow \mathsf{feval}\ \mathfrak{M}\ v\ f_2\\
&\mathsf{feval}\ \mathfrak{M}\ v\ (\mathsf{FALL}\ x\ f) \overset{\text{def}}{=} \forall\, a.\ a \in \mathfrak{M}.\mathit{Dom} \Rightarrow \mathsf{feval}\ \mathfrak{M}\ v(\!|x \mapsto a|\!)\ f
\end{aligned}$$

The equivalence of first order formulas are given as:

Definition 59 (Equivalence of first order formulas)**.**

$$\begin{aligned}
&\mathsf{fequiv}\ \mu\ \mathit{ff}_1\ \mathit{ff}_2 \overset{\text{def}}{=}\\
&\ \forall\, \mathfrak{M}\ \sigma.\\
&\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\texttt{num}) \subseteq \mathfrak{M}.\mathit{frame}.\mathit{world} \Rightarrow\\
&\quad (\mathsf{fsatis}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \sigma\ \mathit{ff}_1 \iff \mathsf{fsatis}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \sigma\ \mathit{ff}_2)
\end{aligned}$$

By induction on first order formula, we show that the truth value of a first order formula only depends on where the valuation sends the free variables to:

Proposition 53 (`holds_valuation`)**.**

$$\vdash (\forall\, x.\ x \in \mathsf{FV}\ p \Rightarrow v_1\ x = v_2\ x) \Rightarrow (\mathsf{feval}\ \mathfrak{M}\ v_1\ p \iff \mathsf{feval}\ \mathfrak{M}\ v_2\ p)$$

Models for modal language can be viewed as a first order model and hence be used to interpret some first order formula, and a first order model can be also viewed as a modal model. These conversions can be done using the following two functions:

Definition 60 (Conversion between modal and first order models).

$$
\begin{array}{l}
\mathsf{mm2folm}\ \mathfrak{M} \overset{\text{def}}{=} \\
\quad <| \mathit{Dom} := \mathfrak{M}.\mathit{frame.world};\ \mathit{Fun} := (\lambda\, n\ args.\ \mathsf{CHOICE}\ \mathfrak{M}.\mathit{frame.world}); \\
\quad \mathit{Pred} := \\
\quad\ (\lambda\, p\ zs. \\
\qquad\quad \mathtt{case}\ zs\ \mathtt{of} \\
\qquad\qquad [] \Rightarrow \mathsf{F} \\
\qquad\quad |\ [w] \Rightarrow w \in \mathfrak{M}.\mathit{frame.world} \wedge \mathfrak{M}.\mathit{valt}\ p\ w \\
\qquad\quad |\ [w;\ w_2] \Rightarrow \\
\qquad\qquad p = 0 \wedge \mathfrak{M}.\mathit{frame.rel}\ w\ w_2 \wedge w \in \mathfrak{M}.\mathit{frame.world} \wedge w_2 \in \mathfrak{M}.\mathit{frame.world} \\
\qquad\quad |\ w :: w_2 :: v_{10} :: v_{11} \Rightarrow \mathsf{F}) |>
\end{array}
$$

$$
\begin{array}{l}
\mathsf{folm2mm}\ FM \overset{\text{def}}{=} \\
\quad <| \mathit{frame} := \\
\qquad <| \mathit{world} := FM.\mathit{Dom}; \\
\qquad\quad \mathit{rel} := (\lambda\, w_1\ w_2.\ FM.\mathit{Pred}\ 0\ [w_1;\ w_2] \wedge w_1 \in FM.\mathit{Dom} \wedge w_2 \in FM.\mathit{Dom}) |>; \\
\quad \mathit{valt} := (\lambda\, v\ w.\ FM.\mathit{Pred}\ v\ [w] \wedge w \in FM.\mathit{Dom}) |>
\end{array}
$$

Note a first order model from a modal model has no interesting information about function symbols and higher-ary predicate symbols. And by viewing a first order model as a modal model, we lose all the function symbols and predicate symbols except for the unary ones, and the binary one denoted by 0. So the above constructions are not inverses in general. Also observe the range of formulas which makes sense to first order model obtained by converting a modal model is quite limited, we can only use these model to interpret formulas with only one binary predicate symbol corresponds to the relation in the model, denoted by 0, and no function symbol. A first order model which do not have 'superfluous' symbols contains the same amount of information as a modal model. For such a first order model, converting it to a modal model and then to a first order model again get the original model back, in the sense of the resultant model satisfied exactly the same first order formulas without function symbols. The fact that we that we need the assumption $\mathsf{form\_functions}\ f = \emptyset$ in the following proposition is not a real constrain, but merely an assumption for well-formedness, since any formulas with function symbol does not make sense to both the original model and the resultant model we get by conversion:

Proposition 54 (`mm2folm_folm2mm_feval`).

$$
\begin{array}{l}
\vdash \mathsf{form\_functions}\ f = \emptyset \Rightarrow \\
\quad \forall\, \sigma. \\
\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathfrak{M}.\mathit{Dom} \Rightarrow \\
\qquad (\forall\, n.\ \mathfrak{M}.\mathit{Pred}\ n\ [] \iff \mathsf{F}) \Rightarrow \\
\qquad (\forall\, a\ b\ n.\ \mathfrak{M}.\mathit{Pred}\ n\ [a;\ b] \Rightarrow n = 0) \Rightarrow \\
\qquad (\forall\, a\ b\ c\ d\ n.\ \mathfrak{M}.\mathit{Pred}\ n\ (a :: b :: c :: d) \iff \mathsf{F}) \Rightarrow \\
\qquad (\mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{folm2mm}\ \mathfrak{M}))\ \sigma\ f \iff \mathsf{feval}\ \mathfrak{M}\ \sigma\ f)
\end{array}
$$

With the setup on basics about first order logic and how does it interact with modal logic, let us start building intuition about how does modal formulas corresponds to first order formulas. Observe that unlike modal formulas which atomic formulas are propositional letters standing alone, even the atomic first order formulas (except fFALSE) have variable symbols involved. Hence to translate a modal formula into a first order formula, we must get variables involved as well. This variables is used to mark the state we are looking at. And moreover, once we see a $\Diamond$ in a modal formula, we start talking about some other state which is related to the current state, so we will need another variable symbol to mark the new state of interest. Hence in mathematical language, the natural correspondence of modal and first order formulas is given by:

$ST_x(p) = Px$

$ST_x(\bot) = \bot$

$ST_x(\neg\phi) = \neg ST_x(\phi)$

$ST_x(\phi \vee \psi) = ST_x(\phi) \vee ST_x(\psi)$

$ST_x(\Diamond\phi) = \exists y(Rxy \wedge ST_y(\phi))$

We can read $ST_x\phi$ as 'the standard translation of ϕ at x'. In the definition in the HOL, the x, which is a variable symbol in our construction, is a natural number. Each time we see a $\Diamond$, we come with a fresh variable denotes by $x + 1$:

Definition 61 (Standard Translation).

$$\begin{aligned}
&\mathsf{ST}\ x\ (\mathsf{VAR}\ p) \overset{\text{def}}{=} \mathsf{fP}\ p\ (\mathsf{fV}\ x)\\
&\mathsf{ST}\ x\ \bot \overset{\text{def}}{=} \mathsf{fFALSE}\\
&\mathsf{ST}\ x\ (\neg phi) \overset{\text{def}}{=} \mathsf{fNOT}\ (\mathsf{ST}\ x\ phi)\\
&\mathsf{ST}\ x\ (\mathsf{DISJ}\ phi\ psi) \overset{\text{def}}{=} \mathsf{fDISJ}\ (\mathsf{ST}\ x\ phi)\ (\mathsf{ST}\ x\ psi)\\
&\mathsf{ST}\ x\ (\Diamond\ phi) \overset{\text{def}}{=} \mathsf{fEXISTS}\ (x\ +\ 1)\ (\mathsf{fAND}\ (\mathsf{fR}\ (\mathsf{fV}\ x)\ (\mathsf{fV}\ (x\ +\ 1)))\ (\mathsf{ST}\ (x\ +\ 1)\ phi))
\end{aligned}$$

Here $\lambda\, p\ \ t.\ \mathsf{fP}\ \ p\ \ t$ is the abbreviation of $\lambda\, p\ \ t.\ \mathsf{fP}\ \ p\ \ t$, and $\lambda\, w_1\ \ w_2.\ \mathsf{fR}\ \ w_1\ \ w_2$ is the abbreviation of $\lambda\, w_1\ \ w_2.\ \mathsf{fR}\ \ w_1\ \ w_2$. Any formula obtained by standard translation only has one free variable x, and a standard translation can never have function symbols:

Proposition 55 (`ST_FV_singleton`).

$$\vdash \mathsf{FV}\ (\mathsf{ST}\ x\ f)\ \subseteq\ \{\ x\ \}$$

Proposition 56 (`ST_form_functions_EMPTY`).

$$\vdash \mathsf{form\_functions}\ (\mathsf{ST}\ x\ f)\ =\ \emptyset$$

We can conjunct standard tranalations to get a standard translation. And the negation of standard translation is again a standard translation. With the idea of how we deal with big conjunction when we prove Hennessy-Milner theorem, we do not need an explicit definition of big conjunction, and keep things simple by just prove the existence of the formula with desired properties.

Proposition 57 (`ST_BIGCONJ`).

$$\begin{aligned}
&\vdash \mathsf{FINITE}\ s \Rightarrow \\
&\quad \forall x. \\
&\qquad (\forall f.\, f \in s \Rightarrow \exists\, phi.\, f = \mathsf{ST}\ x\ phi) \Rightarrow \\
&\qquad \exists\, cf. \\
&\qquad\quad (\forall \mathfrak{M}\ \sigma. \\
&\qquad\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathfrak{M}.\mathit{Dom} \Rightarrow \\
&\qquad\qquad (\mathsf{feval}\ \mathfrak{M}\ \sigma\ cf \iff \forall f.\, f \in s \Rightarrow \mathsf{feval}\ \mathfrak{M}\ \sigma\ f)) \land \\
&\qquad \exists\, psi.\ cf = \mathsf{ST}\ x\ psi
\end{aligned}$$

Proposition 58 (`ST_fNOT`).

$$\vdash \mathsf{ST}\ x\ (\neg f) = \mathsf{fNOT}\ (\mathsf{ST}\ x\ f)$$

Standard translation defined like this can be regarded as a first-order reformulation of modal satisfication, since we have the precise correspondence of modal satisfication and first-order satisfication for standard translations. A modal formula is satisfied a point w in a modal model iff its standrad translation at x is satisfied at the same modal viewed as a first order model when x is valuated to w:

Proposition 59 (`prop_2_47_i`).

$$\begin{aligned}
&\vdash \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathfrak{M}.\mathit{frame.world} \Rightarrow \\
&\quad (\mathsf{satis}\ \mathfrak{M}\ (\sigma\ x)\ phi \iff \mathsf{fsatis}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \sigma\ (\mathsf{ST}\ x\ phi))
\end{aligned}$$

Also if a first order model $\mathfrak{M}$ satisfies a standard translation phi, then if we view $\mathfrak{M}$ as a modal model, it satisfies the modal formula phi.

Proposition 60 (`prop_2_47_i'`).

$$\begin{aligned}
&\vdash \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathfrak{M}.\mathit{Dom} \Rightarrow \\
&\quad (\mathsf{satis}\ (\mathsf{folm2mm}\ \mathfrak{M})\ (\sigma\ x)\ phi \iff \mathsf{feval}\ \mathfrak{M}\ \sigma\ (\mathsf{ST}\ x\ phi))
\end{aligned}$$

Hence we say the current definition of standard translation makes good semantical sense, but syntactically, it has space for improvement: The formula $\Diamond\ (\Diamond\ f)$ with a_1 marking its state is translated to $\exists a_2(Ra_1a_2 \land \exists a_3(Ra_2a_3 \land ST_{a_3}(f)))$, it uses three variables a_1, a_2 and a_3, which is unnecessary: the formula above is equivalent to $\exists a_2(Ra_1a_2 \land \exists a_1(Ra_2a_1 \land ST_{a_1}(f)))$. With this observation, we conclude that we do not need to always come up with a variable symbol for each $\Diamond$, instead, we can use only two variable symbols alternating in each layer. As a consequence, we can redefine standard translation as follows, where x is the variable 0 or 1:

Definition 62 (Alternative definition of standard translation).

$$\begin{aligned}
&\mathsf{ST\_alt}\ x\ (\mathsf{VAR}\ p) \stackrel{\mathsf{def}}{=} \mathsf{fP}\ p\ (\mathsf{fV}\ x) \\
&\mathsf{ST\_alt}\ x\ \bot \stackrel{\mathsf{def}}{=} \mathsf{fFALSE} \\
&\mathsf{ST\_alt}\ x\ (\neg phi) \stackrel{\mathsf{def}}{=} \mathsf{fNOT}\ (\mathsf{ST\_alt}\ x\ phi) \\
&\mathsf{ST\_alt}\ x\ (\mathsf{DISJ}\ phi\ psi) \stackrel{\mathsf{def}}{=} \mathsf{fDISJ}\ (\mathsf{ST\_alt}\ x\ phi)\ (\mathsf{ST\_alt}\ x\ psi) \\
&\mathsf{ST\_alt}\ x\ (\Diamond\ phi) \stackrel{\mathsf{def}}{=} \\
&\quad \mathsf{fEXISTS}\ (1 - x)\ (\mathsf{fAND}\ (\mathsf{fR}\ (\mathsf{fV}\ x)\ (\mathsf{fV}\ (1 - x)))\ (\mathsf{ST\_alt}\ (1 - x)\ phi))
\end{aligned}$$

The above definition ensures that there are only two variables alternating in the formula we get.

Proposition 61 (`ST_alt_two_var`)**.**

$$\begin{aligned}&\vdash \mathsf{FV}\ (\mathsf{ST\_alt}\ 0\ \mathit{phi})\ \cup\ \mathsf{BV}\ (\mathsf{ST\_alt}\ 0\ \mathit{phi})\ \subseteq\ \{\,0;\ 1\,\}\ \wedge\\ &\quad \mathsf{FV}\ (\mathsf{ST\_alt}\ 1\ \mathit{phi})\ \cup\ \mathsf{BV}\ (\mathsf{ST\_alt}\ 1\ \mathit{phi})\ \subseteq\ \{\,0;\ 1\,\}\end{aligned}$$

It is evident from the follow proposition that our new definition of translation is equally nice from the semantical aspect as the former one:

Definition 63 (`prop_2_47_i_alt`)**.**

$$\begin{aligned}&\vdash \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \Rightarrow\\ &\quad (\mathsf{satis}\ \mathfrak{M}\ (\sigma\ 1)\ \mathit{phi}\ \iff\ \mathsf{fsatis}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \sigma\ (\mathsf{ST\_alt}\ 1\ \mathit{phi}))\ \wedge\\ &\quad (\mathsf{satis}\ \mathfrak{M}\ (\sigma\ 0)\ \mathit{phi}\ \iff\ \mathsf{fsatis}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \sigma\ (\mathsf{ST\_alt}\ 0\ \mathit{phi}))\end{aligned}$$

By conclusion, every modal formula is equivalent to a first order formula containing only two variables.

Proposition 62 (`prop_2_49_i`)**.**

$$\vdash \exists\,\mathit{fphi}.\ \mathsf{FV}\ \mathit{fphi}\ \cup\ \mathsf{BV}\ \mathit{fphi}\ \subseteq\ \{\,0;\ 1\,\}\ \wedge\ \mathsf{fequiv}\ \mu\ (\mathsf{ST}\ 0\ \mathit{phi})\ \mathit{fphi}$$

4.2 Modal Saturation via ultrafilter extensions

In the second chapter, we have seen bisimilarity implies modal equivalence, but only proved the converse for image finite models. In this section, we are interested in another particular class of models which modal equivalence implies bisimilarity, which is the class of m-saturated models.

A set of formulas Σ is called satisfiable in a set of worlds X of a model $\mathfrak{M}$ if there exists a world in X such that all the formulas in Σ are satisfied, and is called finitely satisfiable if any finite subset of Σ is satisfiable:

Definition 64 (Satisfiable)**.**

$$\mathsf{satisfiable\_in}\ \Sigma\ X\ \mathfrak{M}\ \stackrel{\mathsf{def}}{=}\ X\ \subseteq\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \wedge\ \exists\,x.\ x\ \in\ X\ \wedge\ \forall\,\phi.\ \phi\ \in\ \Sigma\ \Rightarrow\ \mathsf{satis}\ \mathfrak{M}\ x\ \phi$$

Definition 65 (Finitely satisfiable)**.**

$$\mathsf{fin\_satisfiable\_in}\ \Sigma\ X\ \mathfrak{M}\ \stackrel{\mathsf{def}}{=}\ \forall\,S.\ S\ \subseteq\ \Sigma\ \wedge\ \mathsf{FINITE}\ S\ \Rightarrow\ \mathsf{satisfiable\_in}\ S\ X\ \mathfrak{M}$$

A model $\mathfrak{M}$ is called m-saturated if for each $w\ \in\ \mathfrak{M}.\mathsf{frame}.\mathsf{world}$ and any set Σ of formulas, if Σ is finitely satisfiable in the set of successors of w, then it is satisfiable in the set of successors of w.

Definition 66 (m-saturated)**.**

$$\begin{aligned}&\mathsf{M\_sat}\ \mathfrak{M}\ \stackrel{\mathsf{def}}{=}\\ &\ \forall\,w\ \Sigma.\\ &\quad w\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \Rightarrow\\ &\quad \mathsf{fin\_satisfiable\_in}\ \Sigma\ \{\ v\ \mid\ v\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \wedge\ \mathfrak{M}.\mathit{frame}.\mathit{rel}\ w\ v\ \}\ \mathfrak{M}\ \Rightarrow\\ &\quad \mathsf{satisfiable\_in}\ \Sigma\ \{\ v\ \mid\ v\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \wedge\ \mathfrak{M}.\mathit{frame}.\mathit{rel}\ w\ v\ \}\ \mathfrak{M}\end{aligned}$$

A class of model where modal equivalence implies bisimilarity has a name, it is called a Hennessy-Milner class. If we want to formalise the definition of Hennessy-Milner class in the HOL, we could write:

Definition 67 (Hennessy-Milner class)**.**

$$\begin{aligned}
&\vdash \mathsf{HM\_class}\ K \iff \\
&\quad \forall \mathfrak{M}\ \mathfrak{M}'\ w\ w'. \\
&\qquad \mathfrak{M} \in K \land \mathfrak{M}' \in K \land w \in \mathfrak{M}.\mathit{frame.world} \land w' \in \mathfrak{M}'.\mathit{frame.world} \Rightarrow \\
&\qquad \mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w' \Rightarrow \\
&\qquad \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'
\end{aligned}$$

But in fact, such a definition is useless, since we are not allowed to have a 'class' in the HOL, we can only have set, and any set is only allowed to have elements of the same type. Hence if we use this definition, we will only be allowed to talk about bisimulations between models with the same type. As a consequence, we do not make usage of this definition in the following formalisation of the proposition which says the class of m-saturation has the Hennessy-Milner property, instead, we state it as:

Proposition 63 (`prop_2_54_DIST_TYPE`)**.**

$$\begin{aligned}
&\vdash \mathsf{M\_sat}\ \mathfrak{M} \land \mathsf{M\_sat}\ \mathfrak{M}' \land w \in \mathfrak{M}.\mathit{frame.world} \land w' \in \mathfrak{M}'.\mathit{frame.world} \Rightarrow \\
&\quad \mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w' \Rightarrow \\
&\quad \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'
\end{aligned}$$

Proof. Let $\mathfrak{M}$ and $\mathfrak{M}'$ be $(\alpha, \beta), (\alpha, \gamma)$ models respectively. Under the assumptions, the bisimulation relation is given by $\lambda\, n_1\ n_2.\, \forall \phi.\ \mathsf{satis}\ \mathfrak{M}\ n_1\ \phi \iff \mathsf{satis}\ \mathfrak{M}'\ n_2\ \phi$. The only non-trivial clause of being a bisimulation to check is that for worlds w, v of $\mathfrak{M}$ and world w' of $\mathfrak{M}'$ such that $\mathfrak{M}.\mathsf{frame.rel}\ w\ v$ and w and w' are modal equivalent, we can find a world v' of $\mathfrak{M}'$ such that $\mathfrak{M}'.\mathsf{frame.rel}\ w'\ v'$ and v and v' are modal equivalent.

Let $\mathit{\Sigma}$ denote the set of formulas satisfied by v, it suffices to find a successor of w' that realises $\mathit{\Sigma}$. As $\mathfrak{M}'$ is m-saturated, it suffices to prove each finite subset $\Delta \subseteq \Sigma$ is satisfied in some successor of w'. Take such a Δ, then it is satisfied at v. As Δ is finite, we can prove that there exists a formula ff such that for all (α, β)-models, ff is satisfied at a world if and only if all elements in Δ are satisfied:

$$\begin{aligned}
&\vdash \mathsf{FINITE}\ s \Rightarrow \\
&\quad \exists \mathit{ff}.\, \forall w\ \mathfrak{M}.\ w \in \mathfrak{M}.\mathsf{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{ff} \iff \forall f.\ f \in s \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f)
\end{aligned}$$

We have $\mathsf{satis}\ \mathfrak{M}\ v\ \mathit{ff}$, and therefore $\mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \mathit{ff})$. By modal equivalence of w and w', we then get $\mathsf{satis}\ \mathfrak{M}\ w'\ (\Diamond\ \mathit{ff})$, so there exists a successor of w' that satisfies ff.

But it is not what we want! Instead, we need a successor of w' which satisfies all elements in Σ, but we cannot get it if we just use the lemma above, since the equivalence of the set of formulas and its big conjunction we proved only works for (α, β) models, but we are in $\mathfrak{M}'$ which is an (α, γ)-model. By embedding any arbitary model into a common infinite type, it may possible to prove that under some condition on universe, if a big conjunction works for a model of one type, then it works for any other types. But it is way too complicated and will involve an ugly assumption. The key observation is that what we need is no more than a big conjunction formula that simutinously works for two distinct types. Therefore, it suffices to prove:

$$\begin{aligned}
&\vdash \mathsf{FINITE}\ s \ \Rightarrow \\
&\quad \exists \mathit{ff}. \\
&\qquad (\forall w\ \mathfrak{M}.\ w \in \mathfrak{M}.\mathsf{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{ff} \iff \forall f.\ f \in s \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f)) \ \wedge \\
&\qquad \forall w\ \mathfrak{M}.\ w \in \mathfrak{M}.\mathsf{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{ff} \iff \forall f.\ f \in s \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f)
\end{aligned}$$

This lemma works perfectly, so we will not get stuck at the former point and we are done with the proof. □

Since m-saturated models are nice, here is a natural question: How can we get such models? It turns out that every models can give arise to a m-saturated model, by taking its ultrafilter extension. In order to talk about ultrafilter extesnions, we need to build a theory of ultrafilers in HOL.

4.3 Interlude II: Ultrafilters

As its name suggests, an ultrafilter is a special kind of filter. Given a non-empty set W, a set F which is a subset of the power set of W, written as $POW(W)$ in the HOL, is called a filter if it contains W itself, closed under binary intersection, and closed upward.

Definition 68 (Filter)**.**

$$\begin{aligned}
&\mathsf{filter}\ \mathit{FLT}\ W \overset{\mathsf{def}}{=} \\
&\quad W \neq \emptyset \wedge \mathit{FLT} \subseteq \mathsf{POW}\ W \wedge W \in \mathit{FLT} \wedge (\forall X\ Y.\ X \in \mathit{FLT} \wedge Y \in \mathit{FLT} \Rightarrow X \cap Y \in \mathit{FLT}) \ \wedge \\
&\quad \forall X\ Z.\ X \in \mathit{FLT} \wedge X \subseteq Z \wedge Z \subseteq W \Rightarrow Z \in \mathit{FLT}
\end{aligned}$$

By induction, closure under binary intersection implies that a filter is closed under finite intersection.

Proposition 64 (`BIGINTER_IN_filter`)**.**

$$\vdash \mathsf{FINITE}\ s \Rightarrow s \neq \emptyset \Rightarrow \forall U\ W.\ \mathsf{filter}\ U\ W \Rightarrow s \subseteq U \Rightarrow \mathsf{BIGINTER}\ s \in U$$

Obviously for any set W, $POW(W)$ is a filter on W. By upward closure, any filter which contains the empty set must be $POW(W)$.

Proposition 65 (`POW_filter`)**.**

$$\vdash W \neq \emptyset \Rightarrow \mathsf{filter}\ (\mathsf{POW}\ W)\ W$$

Proposition 66 (`empty_improper_filter`)**.**

$$\vdash \mathsf{filter}\ U\ W \wedge \emptyset \in U \Rightarrow U = \mathsf{POW}\ W$$

But $POW(W)$ is very boring as a filter, we are mainly interested in filters which are not the whole power set, these filters are called proper filer:

Definition 69 (Proper filter).

$$\mathsf{proper\_filter}\ FLT\ W \overset{\text{def}}{=} \mathsf{filter}\ FLT\ W\ \wedge\ FLT\ \neq\ \mathsf{POW}\ W$$

If we have a set proper filters such that for any two members A, B of it, either $A \subset B$ or $B \subseteq A$, the union of this set is a proper filter.

Proposition 67 (`UNION_proper_proper`).

$$\begin{aligned}\vdash\ & W\ \neq\ \emptyset\ \wedge\ U\ \neq\ \emptyset\ \wedge\ (\forall A.\ A\ \in\ U\ \Rightarrow\ \mathsf{proper\_filter}\ A\ W)\ \wedge \\ & (\forall A\ B.\ A\ \in\ U\ \wedge\ B\ \in\ U\ \Rightarrow\ A\ \subseteq\ B\ \vee\ B\ \subseteq\ A)\ \Rightarrow \\ & \mathsf{proper\_filter}\ (\mathsf{BIGUNION}\ U)\ W\end{aligned}$$

Another important kind of filter is the generated filter. There are three versions of definitions:

- The filter generated by $E \subseteq POW(W)$ is the smallest filter on $POW(W)$ that contains E.
- The filter generated by E is the big intersection $\bigcup\{F \mid E \subseteq F \wedge F$ is a filter on $W\}$.
- The filter generated by E is given by the set $\{X \mid X \subseteq W \wedge X = W \vee ((\exists S \subseteq E)(S \neq \emptyset \wedge$ S is finite$\wedge) \bigcap S \subseteq X)\}$

We prove all these three are equivalent. We use the second definition, and prove that it has the property as in the first and third definition.

Definition 70 (Generated filter).

$$\mathsf{generated\_filter}\ E\ W \overset{\text{def}}{=} \mathsf{BIGINTER}\ \{\ G\ /\ E\ \subseteq\ G\ \wedge\ \mathsf{filter}\ G\ W\ \}$$

Use the big intersection in the definition, we derive the induction principle of a generated filter:

Proposition 68 (`generated_filter_ind`).

$$\begin{aligned}\vdash\ & E\ \subseteq\ P\ \wedge\ A\ \neq\ \emptyset\ \wedge\ P\ \subseteq\ \mathsf{POW}\ A\ \wedge\ A\ \in\ P\ \wedge\ (\forall X\ Y.\ X\ \in\ P\ \wedge\ Y\ \in\ P\ \Rightarrow\ X\ \cap\ Y\ \in\ P)\ \wedge \\ & (\forall X\ Z.\ X\ \in\ P\ \wedge\ X\ \subseteq\ Z\ \wedge\ Z\ \subseteq\ A\ \Rightarrow\ Z\ \in\ P)\ \Rightarrow \\ & \forall e.\ e\ \in\ \mathsf{generated\_filter}\ E\ A\ \Rightarrow\ e\ \in\ P\end{aligned}$$

Under this definition, the fact that generated filter is a filter does not come out for free, but easy enough to prove using induction principle:

Proposition 69 (`generated_FT_FT`).

$$\vdash E\ \subseteq\ \mathsf{POW}\ W\ \wedge\ W\ \neq\ \emptyset\ \Rightarrow\ \mathsf{filter}\ (\mathsf{generated\_filter}\ E\ W)\ W$$

Such a filter is the minimal filter that contains E directly by definition:

Proposition 70 (`generated_filter_minimal`).

$$\vdash E\ \subseteq\ \mathsf{POW}\ W\ \wedge\ W\ \neq\ \emptyset\ \Rightarrow\ \forall F_0.\ \mathsf{filter}\ F_0\ W\ \wedge\ E\ \subseteq\ F_0\ \Rightarrow\ \mathsf{generated\_filter}\ E\ W\ \subseteq\ F_0$$

It is easy to see what subsets are in the generated filter: We put any subset W that contains a finite intersection of elements in F in the generated filter of F, such subsets are precisely the ones that are required by the definition. We also use the induction principle of generated filter to prove this concrete construction is equivalent to our definition

Proposition 71 (`generated_filter_elements`)**.**

$$\begin{aligned}&\vdash E \subseteq \mathsf{POW}\ W \wedge W \neq \emptyset \Rightarrow \\ &\quad \mathsf{generated\_filter}\ E\ W = \\ &\qquad \{\, X \mid X \subseteq W \wedge (X = W \vee \exists S.\, S \subseteq E \wedge \mathsf{FINITE}\ S \wedge S \neq \emptyset \wedge \mathsf{BIGINTER}\ S \subseteq X) \,\}\end{aligned}$$

For any element $w \in W$, the set of all the subset containing w forms a filter, it is called the principle filter generated by w. In fact, principle filters are filter generated by singletons.

Definition 71 (Principle filter)**.**

$$\mathsf{principle\_UF}\ w\ W \overset{\text{def}}{=} \{\, X \mid X \subseteq W \wedge w \in X \,\}$$

An ultrafilter on a set W is a proper filter F such that for any $S \subseteq W$, either S or W/S is in F, but not both.

Definition 72 (Ultrafilter)**.**

$$\mathsf{ultrafilter}\ U\ W \overset{\text{def}}{=} \mathsf{proper\_filter}\ U\ W \wedge \forall X.\, X \in \mathsf{POW}\ W \Rightarrow (X \in U \iff W\ \mathsf{DIFF}\ X \notin U)$$

Principle filters are examples of the type of filters of our main interest, that is, the ultrafilters.

Proposition 72 (`principle_UF_UF`)**.**

$$\vdash W \neq \emptyset \wedge w \in W \Rightarrow \mathsf{ultrafilter}\ (\mathsf{principle\_UF}\ w\ W)\ W$$

Since ultrafilter are important, we may ask when can we get an ultrafilter on W from a subset of $POW(W)$. We will prove later that the answer is for any subset of $POW(W)$ with finite intersection property, we can extend it into an ultrafilter. A subset of $POW(W)$ has finite intersection property if it is closed under finite intersection:

Definition 73 (Finite intersection property)**.**

$$\vdash \mathsf{FIP}\ S\ W \iff S \subseteq \mathsf{POW}\ W \wedge \forall S'.\, S' \subseteq S \wedge \mathsf{FINITE}\ S' \wedge S' \neq \emptyset \Rightarrow \mathsf{BIGINTER}\ S' \neq \emptyset$$

Any proper filter U has the finite intersection property. Moreover, for $B \subseteq W$ such that neither B nor W/B is in U, then inserting B to U does not destory the finite intersection property of U.

Proposition 73 (`proper_filter_FIP`)**.**

$$\vdash \mathsf{proper\_filter}\ U\ W \Rightarrow \mathsf{FIP}\ U\ W$$

Proposition 74 (`proper_filter_INSERT_FIP`).

$$\vdash \textsf{proper\_filter}\ U\ W\ \wedge\ B\ \in\ \textsf{POW}\ W\ \wedge\ B\ \notin\ U\ \wedge\ W\ \textsf{DIFF}\ B\ \notin\ U\ \Rightarrow\ \textsf{FIP}\ (\{\ B\ \}\ \cup\ U)\ W$$

Proof. `proper_filter_FIP` is immediate from closure under finite intersection. We prove `proper_filter_-INSERT_FIP`. Take a finite subset $S \subseteq \{B\} \cup U$, if $B \notin S$, then we are done by `proper_filter_FIP`. Otherwise, if we have $\bigcap S = B \cap (\bigcap U')$ for some finite $U' \subseteq U$, which implies $\bigcap U' \subseteq W/B$, hence $W/B \in U$ by upward closure, contradicting the assumption that $W/B \notin U$. □

Towards the goal of extending a set with finite intersection property to an ultrafilter, we firstly prove that such sets can be extended into a proper filter.

Proposition 75 (`FIP_PSUBSET_proper_filter`).

$$\vdash W\ \neq\ \emptyset\ \wedge\ \textsf{FIP}\ S\ W\ \Rightarrow\ \exists\, V.\ \textsf{proper\_filter}\ V\ W\ \wedge\ S\ \subseteq\ V$$

Proof. The desired proper filter is given by the generated filter of S, checking its property is straightfoward. □

A proper filter which is not properly contained by any proper filter is called a maximal filter. We can readily check ultrafilters are maximal. With the last two lemmas about finite intersection property, we can prove maximal filters are ultrafilters, so maximal filters and ultrafilters turns out to be the same thing.

Proposition 76 (`maximal_ultrafilter`).

$$\vdash \textsf{proper\_filter}\ U\ W\ \wedge\ (\forall\, S.\ \textsf{filter}\ S\ W\ \wedge\ U\ \subset\ S\ \Rightarrow\ S\ =\ \textsf{POW}\ W)\ \Rightarrow\ \textsf{ultrafilter}\ U\ W$$

Proof. If a maximal filter U on W is not an ultrafilter, then for some A, either $A \in U \wedge W/A \in U$ or $A \notin U \wedge W/A \notin U$. In the first case we have $\emptyset \in U$, contradicts the properness of maximal filter. In the second case, by `proper_filter_INSERT_FIP`, the set $U \cup \{A\}$ has the finite intersection property, hence extends to a proper filter U' by `FIP_PSUBSET_proper_filter`. But then U is properly contained in U', contradicts the maximality of U. This completes the proof. □

Together with the Zorn's lemma which is already in the HOL library, now we have all ingredients of the proof of ultrafilter theorem.

Theorem 9 (`ultrafilter_theorem`).

$$\vdash \textsf{proper\_filter}\ f\ w\ \Rightarrow\ \exists\, U.\ \textsf{ultrafilter}\ U\ w\ \wedge\ f\ \subseteq\ U$$

Proof. Given a property filter F, we will find out a ultrafilter containing it. Consider the set S of all the proper filters containing F, ordered by inclusion, we will apply Zorn's lemma on S. The Zorn's lemma in the HOL looks like:

$$\begin{aligned}&\vdash s\ \neq\ \emptyset\ \wedge\ \textsf{partial\_order}\ r\ s\ \wedge\ (\forall\, t.\ \textsf{chain}\ t\ r\ \Rightarrow\ \textsf{upper\_bounds}\ t\ r\ \neq\ \emptyset)\ \Rightarrow\\ &\quad \exists\, x.\ x\ \in\ \textsf{maximal\_elements}\ s\ r\end{aligned}$$

Here s is a set, r is a relation that takes a pair of elements (a,b) with $a,b \in s$ and return the truth value whether a and b are related. For a set t with the same type as s, we have $\mathsf{chain}\ t\ r$ if for any $a,b \in t$, we have either $(a,b) \in r$ or $(b,a) \in r$. Here our s is the set S defined above. r is defined by inclusion. For $A,B \in S$, $(A,B) \in r$ iff $A \subseteq B$, so a chain T on S is a subset of S such that $A \subseteq B$ or $B \subseteq A$ for any $A,B \in T$.

The thing to check is that any chain T on S has an upper bound. If the chain is empty, then the upper bound is clearly F. Otherwise, we claim the union of the chain is the upper bound. To prove the claim, it amounts to check:

- $\bigcup T$ is a proper filer containing F.
- Any element in T is a subset of $\bigcup T$.

The second item is trivial, the first one is by `UNION_proper_proper`.

Applying the Zorn's lemma gives a proper filter $X \in S$ which is an maximal element of S, it suffices to prove X is an ultrafilter. The fact that X is the maximal element of S proves X is a maximal filter, hence we are done by `maximal_ultrafilter`. $\square$

As a corollary, we can extend any set with finite intersection into an ultrafilter:

Proposition 77 (`ultrafilter_theorem_corollary`)**.**

$$\vdash \mathsf{FIP}\ s\ W\ \wedge\ W \neq \emptyset\ \Rightarrow\ \exists\, u.\ \mathsf{ultrafilter}\ u\ W\ \wedge\ s\ \subseteq\ u$$

Proof. For $S \subseteq POW(W)$ with finite intersection property, by `FIP_PSUBSET_proper_filter`, we have a proper filter P containing S, and P extends to an ultrafiler $S \subseteq P \subseteq U$ by `ultrafilter_theorem`. $\square$

As an application of `ultrafilter_theorem_corollary`, we prove the existence of countably incomplete ultrafilters. A countably incomplete ultrafilter is an ultrafilter which is not closed under countably infinite intersection.

Definition 74 (Countably incomplete ultrafilter)**.**

$$\begin{aligned}&\mathsf{countably\_incomplete}\ U\ W\ \stackrel{\text{def}}{=}\\ &\quad \mathsf{ultrafilter}\ U\ W\ \wedge\ \exists\, \mathit{IFS}\ f.\ \mathit{IFS}\ \subseteq\ U\ \wedge\ \mathsf{BIJ}\ f\ \mathcal{U}(:\mathit{num})\ \mathit{IFS}\ \wedge\ \mathsf{BIGINTER}\ \mathit{IFS}\ =\ \emptyset\end{aligned}$$

It is not hard to see any ultrafilter U on the natural numbers $\mathbb{N}$ which does not contain any singleton is countably incomplete, since the countable intersection of the sets $\mathbb{N}/\{n\}$ which are members of U yields the empty set.

Proposition 78 (`example_2_72`)**.**

$$\vdash \mathsf{ultrafilter}\ U\ \mathcal{U}(:\mathit{num})\ \wedge\ (\forall\, n.\ \{\ n\ \}\ \notin\ U)\ \Rightarrow\ \mathsf{countably\_incomplete}\ U\ \mathcal{U}(:\mathit{num})$$

We find out an ultrafilter on $\mathbb{N}$ which does not contain any singleton, we will done if we can prove the existence of an ultrafilter on $\mathbb{N}$ which does not contain any finite set. By `ultrafilter_theorem_corollary`, it suffices to prove:

Proposition 79 (`exercise_2_5_4_a`).

$$\vdash \mathsf{FIP}\ \{\ \mathcal{U}(:\mathit{num})\ \mathsf{DIFF}\ X\ \ /\ \mathsf{FINITE}\ X\ \}\ \mathcal{U}(:\mathit{num})$$

Proof. By unfolding the definition of finite intersection property and induct on finiteness of the set we are intersecting. □

A countably incomplete ultrafilter has useful features, one of them is that such an ultrafilter U has a chain $I = I_0 \supseteq I_1 \supseteq I_2 \supseteq \cdots$ such that each I_i is in U, and $\bigcap_{n\in\mathbb{N}} I_n = \emptyset$.

Proposition 80 (`countably_incomplete_chain`).

$$\begin{aligned}&\vdash \mathsf{countably\_incomplete}\ U\ I \ \Rightarrow\\ &\quad \exists\, In.\\ &\qquad In\ 0\ =\ I\ \wedge\ (\forall\, n.\ In\ n\ \in\ U\ \wedge\ In\ (n\ +\ 1)\ \subseteq\ In\ n)\ \wedge\\ &\qquad \mathsf{BIGINTER}\ \{\ In\ n\ \ /\ n\ \in\ \mathcal{U}(:\mathit{num})\ \}\ =\ \emptyset\end{aligned}$$

Proof. By definition of countably incompleteness, there exists a family X_n in U indexed by natural numbers such that $\bigcap_{n\in\mathbb{N}} X_n = \emptyset$. Define $J_n := \bigcap_{m\leq n} X_n$, so $J_{n+1} \supseteq J_n$ for all $n \in \mathbb{N}$, and moreover $\bigcap_{n\in\mathbb{N}} I_n = \emptyset$. In the HOL, the family J_n is defined using a primitive recursive function, defined by $\mathsf{PRIM\_REC}\ (X\ 0)\ (\lambda\, Xn\ n.\ Xn\ \cap\ X\ (n\ +\ 1))$. We get the desired chain I_n by inserting I at the begining of J_n. □

Our theory of ultrafilters built above is to enable us to talk about ultrafilter extensions. Ultrafilter extension is defined as a function that takes a model $\mathfrak{M}$, and give the extended model. The world set of $\mathsf{UE}\ \mathfrak{M}$ is the set of all the ultrafilters on the world set of $\mathfrak{M}$. And the relation of these worlds requires some explaination. For $X \subseteq \mathfrak{M}.\mathsf{frame.world}$, there comes two interesting sets of worlds, one is the set of worlds that is linked to some world in X, and the other one is the set of worlds that are only linked to worlds in X. These two sets are dual to each other:

Definition 75 ('Can see').

$$\mathsf{can\_see}\ \mathfrak{M}\ X\ \stackrel{\mathsf{def}}{=}\ \{\ w\ \ /\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \exists\, x.\ x\ \in\ X\ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ w\ x\ \}$$

Definition 76 ('Only see').

$$\begin{aligned}&\mathsf{only\_see}\ \mathfrak{M}\ X\ \stackrel{\mathsf{def}}{=}\\ &\quad \{\ w\ \ /\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \forall\, x.\ x\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathfrak{M}.\mathit{frame.rel}\ w\ x\ \Rightarrow\ x\ \in\ X\ \}\end{aligned}$$

Proposition 81 (`can_only_dual`).

$$\begin{aligned}&\vdash X\ \subseteq\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\\ &\quad \mathsf{can\_see}\ \mathfrak{M}\ X\ =\ \mathfrak{M}.\mathit{frame.world}\ \mathsf{DIFF}\ \mathsf{only\_see}\ \mathfrak{M}\ (\mathfrak{M}.\mathit{frame.world}\ \mathsf{DIFF}\ X)\end{aligned}$$

Proposition 82 (`only_can_dual`).

$$\begin{aligned}&\vdash X\ \subseteq\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\\ &\quad \mathsf{only\_see}\ \mathfrak{M}\ X\ =\ \mathfrak{M}.\mathit{frame.world}\ \mathsf{DIFF}\ \mathsf{can\_see}\ \mathfrak{M}\ (\mathfrak{M}.\mathit{frame.world}\ \mathsf{DIFF}\ X)\end{aligned}$$

Directly from their definitions, the worlds satisfying $\Diamond$ *phi* are the worlds that can see a world satisfying ϕ. And the worlds satisfying $\Box$ *phi* are the worlds that can only see worlds that satisfies ϕ, and the set of worlds which only see $X \cap Y$ is the intersection of worlds only see X and worlds only see Y:

Proposition 83 (`valt_can_see`).

$$\begin{aligned}&\vdash \{\ w\ /\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \phi)\ \}\ =\\ &\quad \mathsf{can\_see}\ \mathfrak{M}\ \{\ v\ /\ v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathsf{satis}\ \mathfrak{M}\ v\ \phi\ \}\end{aligned}$$

Proposition 84 (`valt_only_see`).

$$\begin{aligned}&\vdash \{\ w\ /\ w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Box\ \phi)\ \}\ =\\ &\quad \mathsf{only\_see}\ \mathfrak{M}\ \{\ v\ /\ v\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathsf{satis}\ \mathfrak{M}\ v\ \phi\ \}\end{aligned}$$

Proposition 85 (`only_see_INTER`).

$$\vdash \mathsf{only\_see}\ \mathfrak{M}\ (X\ \cap\ Y)\ =\ \mathsf{only\_see}\ \mathfrak{M}\ X\ \cap\ \mathsf{only\_see}\ \mathfrak{M}\ Y$$

We define two ultrafilter u, v to be related if any world set of $\mathfrak{M}$ in v can only see world sets in u. Such a definition has a reformulation: for any world set of $\mathfrak{M}$, if the set of worlds that it can only see is in u, then this set is in v:

Definition 77 (Relation used in ultrafilter extension).

$$\begin{aligned}&\mathsf{UE\_rel}\ \mathfrak{M}\ u\ v\ \overset{\text{def}}{=}\\ &\ \mathsf{ultrafilter}\ u\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathsf{ultrafilter}\ v\ \mathfrak{M}.\mathit{frame.world}\ \wedge\\ &\ \forall X.\ X\ \in\ v\ \Rightarrow\ \mathsf{can\_see}\ \mathfrak{M}\ X\ \in\ u\end{aligned}$$

Proposition 86 (`exercise_2_5_5`).

$$\begin{aligned}&\vdash \mathsf{ultrafilter}\ u\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathsf{ultrafilter}\ v\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\\ &\quad (\mathsf{UE\_rel}\ \mathfrak{M}\ u\ v\ \iff\ \{\ Y\ /\ \mathsf{only\_see}\ \mathfrak{M}\ Y\ \in\ u\ \wedge\ Y\ \subseteq\ \mathfrak{M}.\mathit{frame.world}\,\}\ \subseteq\ v)\end{aligned}$$

The ultrafilter extension is defined as follows:

Definition 78 (Ultrafilter extension).

$$\begin{aligned}&\mathsf{UE}\ \mathfrak{M}\ \overset{\text{def}}{=}\\ &\ \mathit{<|frame}\ :=\ \mathit{<|world}\ :=\ \{\ u\ /\ \mathsf{ultrafilter}\ u\ \mathfrak{M}.\mathit{frame.world}\,\}\,;\ \mathit{rel}\ :=\ \mathsf{UE\_rel}\ \mathfrak{M}\,\mathit{|>};\\ &\ \mathit{valt}\ :=\ (\lambda\,p\ v.\ \mathsf{ultrafilter}\ v\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ \mathfrak{M}.\mathit{valt}\ p\ \cap\ \mathfrak{M}.\mathit{frame.world}\ \in\ v)\,\mathit{|>}\end{aligned}$$

The ultrafilter extension also changes the type of model, namely, it change the type of world set from β to $(\beta \to \mathit{bool}) \to \mathit{bool}$, it is indeed an extension, in the sense that $\mathfrak{M}$ is embedded in $\mathsf{UE}\ \mathfrak{M}$ by the function sending $w\ \in\ \mathfrak{M}.\mathsf{frame.world}$ to the principle ultrafilter $\mathsf{principle\_UF}\ w\ \mathfrak{M}.\mathsf{frame.world}$ generated by w. In general, this embedding does not necessarily give a generated submodel, nevertheless, we have an invariance result for this embedding:

Proposition 87 (`prop_2_59_ii`).

$$\vdash w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\ \mathsf{modal\_eq}\ \mathfrak{M}\ (\mathsf{UE}\ \mathfrak{M})\ w\ (\mathsf{principle\_UF}\ w\ \mathfrak{M}.\mathit{frame.world})$$

This is actually a special case of the following proposition, where u is taken as $\mathsf{principle\_UF}\ w\ \mathfrak{M}.\mathsf{frame.world}$. This proposition captures the idea that ultrafilters are used to describe the sense of 'most of', where the sets in the ultrafilters can be regarded as 'sets which are large enough'. From this aspect, the proposition says that a formula ϕ is satisfied in an ultrafilter u iff ϕ is satisfied at most of worlds in $\mathfrak{M}$, where the sense of 'most of' is measured by u.

Proposition 88 (`prop_2_59_i`).

$$\begin{array}{l} \vdash \mathsf{ultrafilter}\ u\ \mathfrak{M}.\mathit{frame.world} \Rightarrow \\ \quad (\{\ w\ /\ w\ \in\ \mathfrak{M}.\mathit{frame.world} \land \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}\ \in\ u \iff \mathsf{satis}\ (\mathsf{UE}\ \mathfrak{M})\ u\ \mathit{phi}) \end{array}$$

Proof. By induction on phi. Everything except for the diamond case is by unwinding the definitions of $\mathsf{UE}\ \mathfrak{M}$ and ultrafilter. We only give the proof of the diamond case.

From right to left: Suppose $\mathsf{satis}\ (\mathsf{UE}\ \mathfrak{M})\ u\ (\Diamond\ \mathit{phi})$, then $\mathsf{satis}\ (\mathsf{UE}\ \mathfrak{M})\ u'\ \mathit{psi}$ for some $\mathsf{UE\_rel}\ \mathfrak{M}\ u\ u'$. By inductive hypothesis, $\mathsf{satis}\ (\mathsf{UE}\ \mathfrak{M})\ u'\ \mathit{phi}$ implies $\{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}\ \in\ u'$. And then by definition of $\mathsf{UE\_rel}\ \mathfrak{M}$, this implies $\mathsf{can\_see}\ \mathfrak{M}\ \{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}\ \in\ u$. But this set is exactly the set $\{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \mathit{phi})\ \}$ by the observation in `valt_can_see`.

From left to right: Suppose $\{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \mathit{phi})\ \}\ \in\ u$, we need to find out an ultrafilter u' such that $\mathsf{UE\_rel}\ \mathfrak{M}\ u\ u'$ and $\mathsf{satis}\ (\mathsf{UE}\ \mathfrak{M})\ u'\ \mathit{phi}$. Using `exercise_2_5_-5`, $\mathsf{UE\_rel}\ \mathfrak{M}\ u\ u'$ is equivalent to $s = \{\ Y\ \mid\ \mathsf{only\_see}\ \mathfrak{M}\ Y\ \in\ u\ \land\ Y\ \subseteq\ \mathfrak{M}.\mathsf{frame.world}\ \}\ \subseteq\ u'$. We prove $s\ \cup\ \{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}$ has finite intersection property. It suffices to check (1) s is closed under intersection, and (2) The interesection of any element in s with $\{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}$ is non-empty.

For (1), if $a, b \in s$, then both $\mathsf{only\_see}\ \mathfrak{M}\ a$ and $\mathsf{only\_see}\ \mathfrak{M}\ b$ are in u, and hence there intersection. By `only_see_INTER`, $\mathsf{only\_see}\ \mathfrak{M}\ a\ \cap\ \mathsf{only\_see}\ \mathfrak{M}\ b\ =\ \mathsf{only\_see}\ \mathfrak{M}\ (a\ \cap\ b)$, hence $a\ \cap\ b\ \in\ u$.

For (2), let $Y \in s$, then $\mathsf{only\_see}\ \mathfrak{M}\ Y\ \in\ u$, as also $\{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \mathit{phi})\ \}\ \in\ u$, by closure under intersection for an ultrafilter, $\mathsf{only\_see}\ \mathfrak{M}\ Y\ \cap\ \{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \mathit{phi})\ \}\ \neq\ \emptyset$, we obtain an element of $Y\ \cap\ \{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}$ from any element x of it. As $x\ \in\ \{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ (\Diamond\ \mathit{phi})\ \}$, there exists $y\ \in\ \mathfrak{M}.\mathsf{frame.world}$ such that $\mathsf{satis}\ \mathfrak{M}\ y\ \mathit{phi}$, as $x\ \in\ \mathsf{only\_see}\ \mathfrak{M}\ Y$, we have $y \in Y$.

The rest of the proof of finite intersection property is done by induction. Hence by `ultrafilter_-theorem_corollary`, there exists an ultrafilter u' such that $s \cup \{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world} \land \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}\ \subseteq\ u'$. This is the u' we want: We have $\mathsf{UE\_rel}\ \mathfrak{M}\ u\ u'$ since $\{\ Y\ \mid\ \mathsf{only\_see}\ \mathfrak{M}\ Y\ \in\ u\ \} = s \subseteq u'$ and $\{\ w\ \mid\ w\ \in\ \mathfrak{M}.\mathsf{frame.world}\ \land\ \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}\ \}\ \in\ u'$, we get $\mathsf{satis}\ (\mathsf{UE}\ \mathfrak{M})\ u'\ \mathit{phi}$ by inductive hypothesis. ☐

The above proposition leads to a proof of m-saturatedness of ultrafilter extensions.

Proposition 89 (`prop_2_61`).

$$\vdash \mathsf{M\_sat}\ (\mathsf{UE}\ \mathfrak{M})$$

Proof. Suppose Σ is a set of formulas which is finitely satisfiable in the set of successors of a world $u\ \in\ (\mathsf{UE}\ \mathfrak{M}).\mathsf{frame.world}$, we need to find a world $u'\ \in\ (\mathsf{UE}\ \mathfrak{M}).\mathsf{frame.world}$ such that $\mathsf{UE\_rel}\ \mathfrak{M}\ u\ u'$ and satis (UE $\mathfrak{M}$) u' phi for all $phi \in \Sigma$. By `exercise_2_5_5` and `prop_2_59_i`, it amounts to find an ultrafilter u' on $\mathfrak{M}$.frame.world such that $\{ Y \mid$ only_see $\mathfrak{M}$ $Y \in u \} \subseteq u'$ and $\{ w \mid w \in \mathfrak{M}$.frame.world $\wedge$ satis $\mathfrak{M}$ w phi $\} \in u'$ for all $phi \in \Sigma$.

Consider the set $\Delta =\{ \{ w \mid w \in \mathfrak{M}$.frame.world $\wedge \forall phi.\ phi \in s \Rightarrow$ satis $\mathfrak{M}$ w $phi \} \mid$ FINITE $s \wedge s \subseteq \Sigma \} \cup \{ Y \mid$ only_see $\mathfrak{M}$ $Y \in w \wedge Y \subseteq \mathfrak{M}$.frame.world $\}$. Similar as in the proof of `prop_2_59_i`, we check Δ has the finite intersection property. The only nontrivial thing to check is that for a in the first set of union and b in the second set of union, we have $a \cap b \neq \emptyset$.

Suppose $s \subseteq \Sigma$ is finite, and b is a set of worlds in $\mathfrak{M}$ such that only_see $\mathfrak{M}$ $b \in u$, we show $\{ w \mid w \in \mathfrak{M}$.frame.world $\wedge \forall phi.\ phi \in s \Rightarrow$ satis $\mathfrak{M}$ w $phi \} \cap b \neq \emptyset$. Recall Σ is finitely satisfiable in the set of successors of u, we have a world u'' such that UE_rel $\mathfrak{M}$ u u'' and satis (UE $\mathfrak{M}$) u'' phi for all $\phi \in s$, in other worlds, $\{ w \mid w \in \mathfrak{M}$.frame.world $\wedge$ satis $\mathfrak{M}$ w $phi \} \in u''$ for all $\phi \in s$. Then $\{ w \mid w \in \mathfrak{M}$.frame.world $\wedge \forall phi.\ phi \in s \Rightarrow$ satis $\mathfrak{M}$ w $phi \}$ is a big intersection of sets in u'', and hence is in u''. By `exercise_2_5_5` again, UE_rel $\mathfrak{M}$ u u'' gives $\{ Y \mid$ only_see $\mathfrak{M}$ $Y \in u \wedge Y \subseteq \mathfrak{M}$.frame.world $\} \subseteq u''$, so $b \in u''$ as well. As two elements in u'' has nonempty intersection, we are done.

Hence by `ultrafilter_theorem_corollary`, there exists an ultrafilter u' contains Δ, it is trivial to check u' is what we want. □

Finally, as claimed at the begining of this chapter, we arrive at the characterisation of modal equivalence as bisimilarity somewhere else:

Theorem 10 (`thm_2_62`)**.**

$$\begin{aligned}
&\vdash w \in \mathfrak{M}.\textit{frame.world} \wedge w' \in \mathfrak{M}'.\textit{frame.world} \Rightarrow \\
&\quad (\text{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w' \iff \\
&\qquad \text{bisim\_world}\ (\text{UE}\ \mathfrak{M})\ (\text{UE}\ \mathfrak{M}')\ (\text{principle\_UF}\ w\ \mathfrak{M}.\textit{frame.world}) \\
&\qquad\quad (\text{principle\_UF}\ w'\ \mathfrak{M}'.\textit{frame.world}))
\end{aligned}$$

Proof. Bisimulation implies modal equivalence by `thm_2_20`. For the reverse direction, if $w \in \mathfrak{M}$.frame.world and $w' \in \mathfrak{M}'$.frame.world are modal equivalence, then principle_UF w $\mathfrak{M}$.frame.world $\in$ (UE $\mathfrak{M}$).frame.world is modal equivalent to principle_UF w' $\mathfrak{M}'$.frame.world $\in$ (UE $\mathfrak{M}'$).frame.world. As UE $\mathfrak{M}$ and UE $\mathfrak{M}'$ are m-saturated by `prop_2_61`, the result follows by `prop_2_54_DIST_TYPE`. □

5 Invariant under bisimulations and simulations

In this chapter we put everthing we have developed so far together, and use them to give syntactical characterisations of semantical properties 'invariant for bisimulation' and 'preserved under simulation'. This whole chapter realies highly on first order logic, and we will swap between modal models and first order models very frequently. Hence when we talk about first order models, we restrict our scope to the ones which contains the same amount of information as a modal model. And only talk about first order formulas which makes sense to such models. So when we say 'a first order formula' in this chapter, we secretly refer to a first order formula with no function symbols and only have unary predicate symbols and only one binary predicate symbol.

5.1 Standard translations vs invariant under bisimulations

In the last chapter, we see the correspondence between modal formulas and first order formulas, and get known to the fact that every modal formula is equivalent to a first order formula. But actually, standard translation is really a small fragment of modal formulas. Even in our restricted scope, there are many first order formulas which is not equivalent to the standard translation of any formulas. Here we ask: What are the first order formulas which are equivalent to some standard translation? This section is to answer this question.

The short answer to our question is that a first order formula is equivalent to a modal formula iff it is invariant under bisimulations. A first order formula $\alpha(x)$ with at most one free variable x is invariant for bisimulations if for all models $\mathfrak{M}$ and N, with $w \in \mathfrak{M}.\mathsf{frame.world}$ and $v \in N.\mathsf{frame.world}$, if w and v are bisimilar, then $\alpha(x)$ holds in $\mathfrak{M}$ with x sends to w iff it holds in N with x sends to v.

When we see a formula is invariant under bisimulation, we allow $\mathfrak{M}$ and N to be models of any type, and the type of $\mathfrak{M}$ and N do not need to be identical. However, just like the issue we have seen when we defined equiv0, we are not allowed to have new type variables in only the right hand side of a definition. Thus in the HOL, we are not allowed to talk about invariance for bisimulation for any type of models, and have to refer to the type of models which are linked by the bisimulation, the definition is stated like this:

Definition 79 (Invariant for bisimulation)**.**

$$\begin{array}{l}
\mathsf{invar4bisim}\ x\ t_1\ t_2\ phi \overset{\text{def}}{=} \\
\quad \mathsf{FV}\ phi \subseteq \{\ x\ \} \ \wedge \\
\quad \forall \mathfrak{M}\ N\ v\ w. \\
\qquad \mathsf{bisim\_world}\ \mathfrak{M}\ N\ w\ v \ \Rightarrow \\
\qquad \forall \sigma m\ \sigma n. \\
\qquad\quad \mathsf{fsatis}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \sigma m (\!| x \mapsto w |\!)\ phi \iff \\
\qquad\quad\; \mathsf{fsatis}\ (\mathsf{mm2folm}\ N)\ \sigma n (\!| x \mapsto v |\!)\ phi
\end{array}$$

Modal formulas is preserved under bisimulation by `thm_2_20`, and hence their standard translations by `prop_2_47_i`. To justify the above answer to the question, the rest of this section devotes to prove any formulas which is invariant under bisimulation is equivalent to a standard translation. The tools that this proof will use centered on saturated models, which we are introducing now:

Given a first order model $\mathfrak{M}$, we can add some new constants to it to equip the model with more information. By adding new constant, we mean add nullary function symbols, each corresponds to some point in its domain. A model $\mathfrak{M}'$ is an expansion of $\mathfrak{M}$ if it is same as $\mathfrak{M}$ except it has more constants. In the HOL, expansion $\mathfrak{M}\ A\ \mathfrak{M}'\ f$ means '$\mathfrak{M}'$ is the expansion of $\mathfrak{M}$ obtained by adding a bunch of constant, each corresponds an element in A'.

Definition 80 (Expansion).

$$\begin{aligned}
&\textsf{expansion}\ \mathfrak{M}\ A\ \mathfrak{M}'\ f \ \stackrel{\textsf{def}}{=}\\
&\ \ \mathfrak{M}'.\textit{Dom} \ = \ \mathfrak{M}.\textit{Dom} \ \wedge\ \textsf{BIJ}\, f\ (\textsf{count}\ (\textsf{CARD}\ A))\ A\ \wedge\\
&\ \ \mathfrak{M}'.\textit{Fun} \ = \ (\lambda\, n\ args.\ \texttt{if}\, n\ <\ \textsf{CARD}\ A\ \wedge\ args\ =\ []\ \texttt{then}\, f\ n\ \texttt{else}\,\textsf{CHOICE}\ \mathfrak{M}.\textit{Dom})\ \wedge\\
&\ \ \mathfrak{M}'.\textit{Pred} \ = \ \mathfrak{M}.\textit{Pred}
\end{aligned}$$

We are interested in the case that we are adding finitely many constants to a first order model without any function. So when we use this definition, $\mathfrak{M}$ should be a model with no interesting information about constants, and A is a finite set of worlds in $\mathfrak{M}$. The $\textsf{count}\ n$ is the set of the n natural numbers from 0 to $n-1$. For $\mathfrak{M}$ an α-model, f is of type $num \to \alpha$, its role is to assign each natural number an element in A. Hence $\textsf{expansion}\ \mathfrak{M}\ A\ \mathfrak{M}'\ f$ means $\mathfrak{M}'$ is a model that obtained by adding constants $\textsf{Fn}\ n\ []$ with $n\ <\ \textsf{CARD}\ A$ which will evaluate to $f\ n$. For fixed $\mathfrak{M}$ and A, the model $\mathfrak{M}'$ such that $\textsf{expansion}\ \mathfrak{M}\ A\ \mathfrak{M}'\ f$ is not unique, since the f may varies. But it does not matter since all those models will be equivalent, so we can actually refer to 'the' expansion of $\mathfrak{M}'$.

A first order model $\mathfrak{M}$ is called n-saturated if for any set $A\ \subseteq\ \mathfrak{M}.\textsf{frame.world}$ where $\textsf{CARD}\ A\ <\ n$, the any $\mathfrak{M}'$ such that $\textsf{expansion}\ \mathfrak{M}\ A\ \mathfrak{M}'\ f$ for valid f realise every set G of formulas where each formula in G only have one free variable x and is only possible to involve nullary function symbols corresponds to the elements in A. Here the definition of consistence is borrowed from first order logic. A set G is consistent to the theory of a first order model $\mathfrak{M}$ iff any finite set of G is realisable in $\mathfrak{M}$. A model is countably saturated if it is n-saturated for all n.

Definition 81 (Consistent).

$$\begin{aligned}
&\textsf{consistent}\ \mathfrak{M}\ G\ \stackrel{\textsf{def}}{=}\\
&\ \ \forall\, G_0.\\
&\ \ \ \ \textsf{FINITE}\ G_0\ \wedge\ G_0\ \subseteq\ G\ \Rightarrow\\
&\ \ \ \ \exists\,\sigma.\ \textsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \textit{num})\ \subseteq\ \mathfrak{M}.\textit{Dom}\ \wedge\ \forall\, phi.\ phi\ \in\ G_0\ \Rightarrow\ \textsf{fsatis}\ \mathfrak{M}\ \sigma\ phi
\end{aligned}$$

Definition 82 (n-saturated).

$$\begin{aligned}
&\textsf{n\_saturated}\ \mathfrak{M}\ n\ \stackrel{\textsf{def}}{=}\\
&\ \ \forall\, A\ \mathfrak{M}'\ G\ x\ f.\\
&\ \ \ \ \textsf{IMAGE}\, f\ \mathcal{U}(:\ \textit{num})\ \subseteq\ \mathfrak{M}.\textit{Dom}\ \wedge\ \textsf{FINITE}\ A\ \wedge\ \textsf{CARD}\ A\ \leq\ n\ \wedge\ A\ \subseteq\ \mathfrak{M}.\textit{Dom}\ \wedge\\
&\ \ \ \ \textsf{expansion}\ \mathfrak{M}\ A\ \mathfrak{M}'\ f\ \wedge\\
&\ \ \ \ (\forall\, phi.\\
&\ \ \ \ \ \ phi\ \in\ G\ \Rightarrow\\
&\ \ \ \ \ \ \forall\, c.\ c\ \in\ \textsf{form\_functions}\ phi\ \Rightarrow\ \textsf{FST}\ c\ \in\ \textsf{count}\ (\textsf{CARD}\ A)\ \wedge\ \textsf{SND}\ c\ =\ 0)\ \wedge\\
&\ \ \ \ \textsf{ftype}\ x\ G\ \wedge\ \textsf{consistent}\ \mathfrak{M}'\ G\ \Rightarrow\\
&\ \ \ \ \textsf{frealizes}\ \mathfrak{M}'\ x\ G
\end{aligned}$$

Definition 83 (Countably saturated).

$$\textsf{countably\_saturated}\ \mathfrak{M}\ \stackrel{\textsf{def}}{=}\ \forall\, n.\ \textsf{n\_saturated}\ \mathfrak{M}\ n$$

The following result explains how can countably saturated models be related to bisimulations, this is another version of 'modal equivalence implies bisimulation somewhere-else':

Theorem 11 (`thm_2_65_corollary`).

$$\begin{aligned}&\vdash \mathsf{countably\_saturated}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \wedge\ \mathsf{countably\_saturated}\ (\mathsf{mm2folm}\ \mathfrak{M}')\ \wedge\\ &\quad w\ \in\ \mathfrak{M}.\mathit{frame.world}\ \wedge\ w'\ \in\ \mathfrak{M}'.\mathit{frame.world}\ \Rightarrow\\ &\quad \mathsf{modal\_eq}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'\ \Rightarrow\\ &\quad \mathsf{bisim\_world}\ \mathfrak{M}\ \mathfrak{M}'\ w\ w'\end{aligned}$$

By `prop_2_54`, to prove the above, it suffices to prove:

Theorem 12 (`thm_2_65`).

$$\vdash \mathsf{countably\_saturated}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \Rightarrow\ \mathsf{M\_sat}\ \mathfrak{M}$$

Proof. Suppose $\mathsf{countably\_saturated}\ (\mathsf{mm2folm}\ \mathfrak{M})$. Let $a\ \in\ \mathfrak{M}.\mathsf{frame.world}$ and Σ a set of modal formulas which is finitely satisfiable in the set of successors of a. We find a successor of a in $\mathfrak{M}$ realising all the formulas in Σ.

Define $\Sigma'\ =\ \{\ \mathsf{fR}\ (\mathsf{Fn}\ 0\ [])\ (\mathsf{fV}\ x)\ \}\ \cup\ \mathsf{IMAGE}\ (\mathsf{ST}\ x)\ \Sigma$ and $MA =$
$<\!|\,\mathsf{Dom} := (\mathsf{mm2folm}\ \mathfrak{M}).\mathsf{Dom};$
$\mathsf{Fun} := (\lambda\, n\ \mathit{args}.\ \texttt{if}\ n = 0 \wedge \mathit{args} = []\ \texttt{then}\ w\ \texttt{else}\ \mathsf{CHOICE}\ (\mathsf{mm2folm}\ \mathfrak{M}).\mathsf{Dom});$
$\mathsf{Pred} := (\mathsf{mm2folm}\ \mathfrak{M}).\mathsf{Pred}\,|\!>$, then $\mathsf{expansion}\ (\mathsf{mm2folm}\ \mathfrak{M})\ \{\ w\ \}\ MA\ (\lambda\, n.\ w)$. We claim $\mathsf{consistent}\ MA\ \Sigma'$. Take a finite set $\Sigma_0\ \subseteq\ \Sigma'$, we should find an element in MA realising it. For each element in Σ_0 which is a standard translation, use CHOICE to choose a modal formula $p \in \Sigma$ that is translated to it, note that we do need the choice function since standard translation is not injective under our construction. Collect the formulas we choose into a set ps, then ps is a finite subset of Σ. Recall we have assumed Σ is finitely satisfiable in the set of successors of a, hence there exists $b\ \in\ \mathfrak{M}.\mathsf{frame.world}$ and $\mathfrak{M}.\mathsf{frame.rel}\ a\ b$ such that $\mathsf{satis}\ \mathfrak{M}\ b\ p$ for any $p\ \in\ \Sigma$. It follows by `prop_2_47_i` that no matter $\mathsf{fR}\ (\mathsf{Fn}\ 0\ [])\ (\mathsf{fV}\ x)$ is in Σ or not, we have Σ_0 is realised at b in MA.

This proves $\mathsf{consistent}\ MA\ \Sigma'$. Since $\mathsf{mm2folm}\ \mathfrak{M}$ is countably saturated, Σ' itself is realised in some b in MA. The fact that $\mathsf{fR}\ (\mathsf{Fn}\ 0\ [])\ (\mathsf{fV}\ x)$ is realised at b implies b is a successor of a in $\mathfrak{M}$, and $\mathsf{IMAGE}\ (\mathsf{ST}\ x)\ \Sigma$ is realised at b implies that $\mathsf{satis}\ \mathfrak{M}\ b\ \mathit{phi}$ for any $\mathit{phi}\ \in\ \Sigma$ by `prop_2_47_i`. □

There remains two steps from our main characterisation result, which are discussed in the following two interludes.

5.1.1 Interlude III: Countably saturated models via ultraproducts

The procceding discussion about countably saturated models left us an important question: How do we get some countably saturated models? The fundamental construction underlying the theory of countably saturated models is that of ultraproduct. We firstly construct ultraproduct about sets, then about models.

Suppose $I \neq \emptyset$ and U is an ultrafilter on I. And for each $i \in I$, A_i is a non-empty set. The Cartesian product is the set of functions with domain I such that for all $i \in I$, $f(i) \in A_i$.

Definition 84 (Cartesian product).

$$\mathsf{Cart\_prod}\ I\ A \overset{\text{def}}{=} \{ f \mid \forall i.\ i \in I \Rightarrow f\ i \in A\ i \}$$

For two functions f, g in the Cartesian product, we say f and g are U-equivalent if f and g agrees on some set in U. This defined an equivalent relation on the Cartestian product of the A_i's:

Definition 85 (U-equivalence).

$$\begin{aligned}&\mathsf{Uequiv}\ U\ I\ A\ f\ g \overset{\text{def}}{=}\\ &\quad \mathsf{ultrafilter}\ U\ I \wedge (\forall i.\ i \in I \Rightarrow A\ i \neq \emptyset) \wedge f \in \mathsf{Cart\_prod}\ I\ A \wedge\\ &\quad g \in \mathsf{Cart\_prod}\ I\ A \wedge \{ i \mid i \in I \wedge f\ i = g\ i \} \in U\end{aligned}$$

Definition 86 (`prop_A_16`).

$$\vdash \mathsf{ultrafilter}\ U\ I \Rightarrow \mathsf{Uequiv}\ U\ I\ A\ \mathsf{equiv\_on}\ \mathsf{Cart\_prod}\ I\ A$$

The ultraproduct of A_i modulo U is the set of equivalence classes partitioned by the relation $\mathsf{Uequiv}\ U\ \mathsf{I}\ A$:

Definition 87.

$$\mathsf{ultraproduct}\ U\ I\ A \overset{\text{def}}{=} \mathsf{partition}\ (\mathsf{Uequiv}\ U\ I\ A)\ (\mathsf{Cart\_prod}\ I\ A)$$

In the case where $A_i = A$ for all $i \in I$, the ultraproduct is called ultrapower of W modulo U.

We deal with ultraproduct of modal models first. Given a family MS of modal models indexed by I and an ultrafilter U on I, the ultraproduct model of MS modulo U is the model described as follows:

- The world set is the ultraproduct of world sets of Mis modulo U.
- For two equivalence classes f_U, g_U of functions in the ultraproduct, they are related iff there exists $f \in f_U, g \in g_U$ such that $\{ i \in \mathsf{I} \mid (MS\ i).\mathsf{frame.rel}\ (f\ i)\ (g\ i) \}$ is in U.
- For a propositional letter p and an equivalence class f_U, we have p is satisfied at f_U iff there exists $g \in f_U$ such that $\{ i \mid i \in \mathsf{I} \wedge f\ i \in (MS\ i).\mathsf{valt}\ p \}$ is in U.

The HOL definition looks like this:

Definition 88 (Ultraproduct of modal models).

$$\begin{aligned}&\mathsf{ultraproduct\_model}\ U\ I\ MS \overset{\text{def}}{=}\\ &\ \langle|\mathit{frame} :=\\ &\quad \langle|\mathit{world} := \mathsf{ultraproduct}\ U\ I\ (\mathsf{models2worlds}\ MS);\\ &\quad\ \mathit{rel} :=\\ &\qquad (\lambda fu\ gu.\\ &\qquad\quad \exists f\ g.\\ &\qquad\qquad f \in fu \wedge g \in gu \wedge\\ &\qquad\qquad \{ i \mid i \in I \wedge (MS\ i).\mathit{frame.rel}\ (f\ i)\ (g\ i) \} \in U)|\rangle;\\ &\ \mathit{valt} := (\lambda p\ fu.\ \exists f.\ f \in fu \wedge \{ i \mid i \in I \wedge f\ i \in (MS\ i).\mathit{valt}\ p \} \in U)|\rangle\end{aligned}$$

Note that since the choice of representatives does not make any difference since $\mathsf{Uequiv}\ U\ \mathsf{I}\ A$ is an equivalence relation, we get an equivalent definition of ultraproduct models if we replace all the existential quantifers with universal quantifiers.

where the function $\mathsf{models2worlds}$ is used here to extract the underlying sets of models.

$$\mathsf{models2worlds}\ \mathit{MS} \overset{\text{def}}{=} (\lambda\, i.\ (\mathit{MS}\ i).\mathsf{frame.world})$$

Our central result about ultraproduct on modal models is a 'modal verion' of the Los theorem. The standard version of Los theorem will be discussed in a moment.

Theorem 13 (`Los_modal_thm`)**.**

$$\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ J\ \Rightarrow \\
\quad \forall\, \mathit{phi}\ \mathit{fc}. \\
\qquad \mathit{fc}\ \in\ (\mathsf{ultraproduct\_model}\ U\ J\ \mathit{Ms}).\mathit{frame.world}\ \Rightarrow \\
\qquad (\mathsf{satis}\ (\mathsf{ultraproduct\_model}\ U\ J\ \mathit{Ms})\ \mathit{fc}\ \mathit{phi}\ \iff \\
\qquad\quad \exists\, f.\ f\ \in\ \mathit{fc}\ \wedge\ \{\ i\ /\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (f\ i)\ \mathit{phi}\ \}\ \in\ U)
\end{array}$$

Proof. Given an ultrafilter U over J and a family Ms of models. We proceed by induction on phi, the base case for $\mathit{phi}\ =\ \mathsf{VAR}\ p$ is directly by definition, and the case for $\mathit{phi}\ =\ \bot$ is by the fact that the empty set is not in the ultrafilter. The boolean cases are by basic property of ultrafilters. We only spell out the proof for diamond case. Suppose for any equivalence fc in the ultraproduct, we have
$\mathsf{satis}\ (\mathsf{ultraproduct\_model}\ U\ J\ \mathit{Ms})\ \mathit{fc}\ \mathit{phi}\ \iff$
$\exists\, f.\ f\ \in\ \mathit{fc}\ \wedge\ \{\ i\ \mid\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (f\ i)\ \mathit{phi}\ \}\ \in\ U$

Given a world in $\mathsf{ultraproduct\_model}\ U\ J\ \mathit{Ms}$, we prove $\mathsf{satis}\ (\mathsf{ultraproduct\_model}\ U\ J\ \mathit{Ms})\ \mathit{fc}\ (\Diamond\ \mathit{phi})\ \iff$
$\exists\, f.\ f\ \in\ \mathit{fc}\ \wedge\ \{\ i\ \mid\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (f\ i)\ (\Diamond\ \mathit{phi})\ \}\ \in\ U$

Left to right: The assumption says that there is an equivalence class that is related to fc and satisfied phi. Suppose the equivalence class gc is represented by a function g, and fc is represented by the function f. We check the f can be taken as our f as above. By defination of satisfication, our task is to check the set $A\ =$
$\{\ i\ \mid$
$i\ \in\ J\ \wedge\ f\ i\ \in\ (\mathit{Ms}\ i).\mathsf{frame.world}\ \wedge$
$\exists\, v.\ (\mathit{Ms}\ i).\mathsf{frame.rel}\ (f\ i)\ v\ \wedge\ v\ \in\ (\mathit{Ms}\ i).\mathsf{frame.world}\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ v\ \mathit{phi}\ \}$ is in U.

By inductive hypothesis, the fact that phi is satisfied at gc implies the existence of an element in x gc such that $\{\ i\ \mid\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (x\ i)\ \mathit{phi}\ \}$ is in U, but as Uequiv is an equivalence relation, this implies $\{\ i\ \mid\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (g\ i)\ \mathit{phi}\ \}$ is in U. As we can check: Since x and g lives in the same equivalence class, $\{\ i\ \mid\ i\ \in\ J\ \wedge\ g\ i\ =\ x\ i\ \}$ is in U. As ultrafilters are closed under finite intersection, $\{i\ \mid\ i \in J \wedge g\ i = x\ i\} \cap \{i\ \mid\ i \in J \wedge \mathsf{satis}\ (\mathit{Ms}\ i)\ (x\ i)\ \mathit{phi}\}$ is in U, this is a subset of $\{\ i\ \mid\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (g\ i)\ \mathit{phi}\ \}$, hence by upward closure, the result follows. As fc and gc is related, by a same procedure of checking independence of representatives, the set $\{\ i\ \mid\ i\ \in\ J\ \wedge\ (\mathit{Ms}\ i).\mathsf{frame.rel}\ (f\ i)\ (g\ i)\ \}$ is in U. Hence the intersection $\{\ i\ \mid\ i\ \in\ J\ \wedge\ (\mathit{Ms}\ i).\mathsf{frame.rel}\ (f\ i)\ (g\ i)\ \}\ \cap$
$\{\ i\ \mid\ i\ \in\ J\ \wedge\ \mathsf{satis}\ (\mathit{Ms}\ i)\ (g\ i)\ \mathit{phi}\ \}$ is in U. As our A is a supset of this set, A is in U as well.

Right to left: Suppose there is an $f \in fc$ such that $\{ i \mid$
$i \in J \land f\ i \in (Ms\ i)$.frame.world $\land$
$\exists v.\ (Ms\ i)$.frame.rel $(f\ i)\ v \land v \in (Ms\ i)$.frame.world $\land$ satis $(Ms\ i)\ v\ phi\}$ is in U, we need to find an equivalence class which is related to fc and satisfies ϕ, which by definition of relation in ultraproduct model, amounts to find a representative of such an equivalence class. The representative is given by : $\lambda\, i.$
`if` $\exists v.\ (Ms\ i)$.frame.rel $(f\ i)\ v \land v \in (Ms\ i)$.frame.world $\land$ satis $(Ms\ i)\ v\ phi$ `then`
CHOICE $\{ v \mid (Ms\ i)$.frame.rel $(f\ i)\ v \land v \in (Ms\ i)$.frame.world $\land$ satis $(Ms\ i)\ v\ phi\}$
`else` CHOICE $(Ms\ i)$.frame.world

checking this is the correct representative is tedious, but just routine, using representative independence and the property of CHOICE function.

□

In the case that we are taking the ultraproduct of a constant family of models with $MS\ i = \mathfrak{M}$ for all $i \in I$, we get an ultrapower of $\mathfrak{M}$. Specialing `Los_modal_thm` to the case of ultrapowers yields the following proposition.

Corollary 1 (`prop_2_71`)**.**

$$\begin{array}{l}
\vdash (\forall i.\ i \in J \Rightarrow Ms\ i = \mathfrak{M}) \land \textsf{ultrafilter}\ U\ J \Rightarrow \\
\quad \forall phi\ w. \\
\qquad \textsf{satis}\ (\textsf{ultraproduct\_model}\ U\ J\ Ms) \\
\qquad\ \ \{ fw \mid \textsf{Uequiv}\ U\ J\ (\textsf{models2worlds}\ Ms)\ (\lambda\, i.\ w)\ fw \}\ phi \iff \\
\qquad\ \ \textsf{satis}\ \mathfrak{M}\ w\ phi
\end{array}$$

The construction of ultraproduct of first order models is similar to the construction for modal models. Restricting our scope only to first order models with 'well-formed' conditions in the sense of it contains the same information as a modal model does not save much work, so we will just construct it for any first order models. This requires us to deal with function symbols and predicate symbols with any arity.

- A function with its symbol denoted by natural number n will send a list of equivalence class to the equivalence class represented by the function that sending $i \in I$ to $(FMS\ i)$.Fun n (MAP $(\lambda fc.$ CHOICE $fc\ i)\ fs)$. That is, to see where does $i \in I$ goes to, evaluate each representative in the list fs at i, collect them into a list, and use this list as the input of the function denoted by n in the model $FMS\ i$.

- A predicate with its symbol denoted by p will hold for a list zs of equivalence classes iff when we pick representatives in each member of zs and evaluate these representatives at i, the predicate p holds in $FMS\ i$ when we evaluate it with the list MAP $(\lambda fc.$ CHOICE $fc\ i)\ zs$ of the outputs.

To avoid having quantifiers everywhere, we fix the representative for each equivalence class fc to be CHOICE fc.

The definition looks like this:

Definition 89 (Ultraproduct of first order models).

$$\begin{array}{l}
\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS} \overset{\mathsf{def}}{=} \\
\quad \mathsf{<|}\mathit{Dom} := \mathsf{ultraproduct}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS}); \\
\quad \mathit{Fun} := \\
\quad\ (\lambda\, n\, fs. \\
\qquad \{\ y\ \mathsf{|} \\
\qquad (\forall\, i.\ i \in I \Rightarrow y\ i \in (\mathit{FMS}\ i).\mathit{Dom}) \wedge \\
\qquad \{\ i\ \mathsf{|}\ i \in I \wedge y\ i = (\mathit{FMS}\ i).\mathit{Fun}\ n\ (\mathsf{MAP}\ (\lambda\, fc.\ \mathsf{CHOICE}\ fc\ i)\ fs)\ \} \in U\ \}); \\
\quad \mathit{Pred} := (\lambda\, p\ zs.\ \{\ i\ \mathsf{|}\ i \in I \wedge (\mathit{FMS}\ i).\mathit{Pred}\ p\ (\mathsf{MAP}\ (\lambda\, fc.\ \mathsf{CHOICE}\ fc\ i)\ zs)\ \} \in U)\mathsf{|>}
\end{array}$$

We now prove this construction has nice semantical behaviour.

For an ultraproduct model of the family FMS of first order models, an evaluation σ assigns each variable symbol an equivalence class in the ultraproduct of the world sets of the family. A term t will be sent to the equivalence class represented by the function obtained as follows. For $i \in I$, the function that assigns a variable symbol n to the representative of $\sigma\ n$ evaluated at i will be an assignment of variable symbols to elements in $\mathit{FMS}\ i$, and i is sent to the element in $(\mathit{FMS}\ i).\mathsf{Dom}$ we will get by evaluating t in $\mathit{FMS}\ i$ using this assignment of variable symbols.

Theorem 14 (`thm_A_19_i`).

$$\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I \Rightarrow \\
\quad \forall\, \sigma\ \mathit{FMS}. \\
\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathsf{ultraproduct}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS}) \Rightarrow \\
\qquad (\forall\, i\ \mathit{ff}\ ll.\ i \in I \Rightarrow (\mathit{FMS}\ i).\mathit{Fun}\ \mathit{ff}\ ll \in (\mathit{FMS}\ i).\mathit{Dom}) \Rightarrow \\
\qquad \mathsf{termval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS})\ \sigma\ t = \\
\qquad\ \{\ f\ \mathsf{|} \\
\qquad\quad \mathsf{Uequiv}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS})\ f \\
\qquad\qquad (\lambda\, i.\ \mathsf{termval}\ (\mathit{FMS}\ i)\ (\lambda\, n.\ \mathsf{CHOICE}\ (\sigma\ n)\ i)\ t)\ \}
\end{array}$$

Proof. By complete induction on `term_size t`. □

The nice semantical behaviour of ultraproduct is evident from the Los theorem:

Theorem 15.

$$\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I \Rightarrow \\
\quad \forall\, \sigma\ \mathit{FMS}. \\
\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathsf{ultraproduct}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS}) \Rightarrow \\
\qquad (\forall\, i\ \mathit{ff}\ ll.\ i \in I \Rightarrow (\mathit{FMS}\ i).\mathit{Fun}\ \mathit{ff}\ ll \in (\mathit{FMS}\ i).\mathit{Dom}) \Rightarrow \\
\qquad (\mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS})\ \sigma\ \mathit{phi} \iff \\
\qquad\ \{\ i\ \mathsf{|}\ i \in I \wedge \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ \mathit{phi}\ \} \in U)
\end{array}$$

Proof. By induction on *phi*. The base case for fFALSE comes from the fact that the empty set is not in an ultrafilter. And the atomic case reduces to check representative independence by unwinding the definitions. We should prove $S_1 = \{\, i \mid i \in \mathsf{I} \wedge (\mathit{FMS}\ i).\mathsf{Pred}\ n\ (\mathsf{MAP}\ (\lambda\, x.\ \mathsf{CHOICE}\ (\mathsf{termval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ \sigma\ x)\ i)\ l)\,\}$ is in U iff $S_2 = \{\, i \mid i \in \mathsf{I} \wedge (\mathit{FMS}\ i).\mathsf{Pred}\ n\ (\mathsf{MAP}\ (\mathsf{termval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i))\ l)\,\}$ is in U.

Let $I_0 =$
$\{\, i \mid$
$i \in \mathsf{I} \wedge$
$\mathsf{MAP}\ (\lambda\, x.\ \mathsf{CHOICE}\ (\mathsf{termval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ \sigma\ x)\ i)\ l = \mathsf{MAP}\ (\mathsf{termval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i))\ l\,\}$, then $I_0 \cap S_1 = I_0 \cap S_2$. If $I_0 \in U$, suppose in addition that S_1 is in U, then S_2 is a supset of the set $I_0 \cap S_2$, which is in U, and vice versa. So it suffices to prove $I_0 \in U$. Obviously, if the two maps we are interested in agree on each member of l, then the resultant list we get will be equal. That gives $A =$
$\mathsf{BIGINTER}$
$\{\, \{\, i \mid i \in \mathsf{I} \wedge \mathsf{CHOICE}\ (\mathsf{termval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ \sigma\ a)\ i = \mathsf{termval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ a\,\} \mid$
$\mathsf{MEM}\ a\ l\,\}$ is a subset of I_0, reduces our task to prove A is in U. But by `thm_A_19_i`, for each member a of l, $\mathsf{termval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ \sigma\ a$ is the equivalence class represents by $\lambda\, i.\ \mathsf{termval}\ (\mathit{FMS}\ i)\ (\lambda\, n.\ \mathsf{CHOICE}\ (\sigma\ n)\ i)\ a$. Hence $\{\, i \mid$
$i \in \mathsf{I} \wedge$
$\mathsf{CHOICE}\ (\mathsf{termval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ \sigma\ a)\ i = \mathsf{termval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ a\,\}$ is in U for each a. So A is in U as a finite intersection of sets in U.

The implication case is trivial from inductive hypothesis. Finally, we prove the case for universal quantifier. From left to right, suppose *phi* is satisfied at any equivalence class in the ultraproduct model for FMS. We need $\{\, i \mid i \in \mathsf{I} \wedge \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ (\mathsf{FALL}\ n\ \mathit{phi})\,\} \in U$. Suppose not, then as U is an ultrafilter, $B = \{\, i \mid i \in \mathsf{I} \wedge \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ (\mathsf{fEXISTS}\ n\ (\mathsf{fNOT}\ \mathit{phi}))\,\} \in U$. Use choice, define the function f to send $i \in I$ to a chosen point in $(\mathit{FMS}\ i).\mathsf{Dom}$ where *phi* is not satisfied. Such a function in HOL looks like: $f =$
$(\lambda\, i.$
`if` $\exists\, a.\ a \in (\mathit{FMS}\ i).\mathsf{Dom} \wedge \neg\mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)(\!|n \mapsto a|\!)\ \mathit{phi}$ `then`
$\mathsf{CHOICE}\ \{\, a \mid a \in (\mathit{FMS}\ i).\mathsf{Dom} \wedge \neg\mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)(\!|n \mapsto a|\!)\ \mathit{phi}\,\}$
`else` $\mathsf{CHOICE}\ (\mathit{FMS}\ i).\mathsf{Dom})$. Then $\{\ i \mid i \in \mathsf{I} \wedge \neg\mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)(\!|n \mapsto f\ i|\!)\ \mathit{phi}\ \} = B$, and hence is in U. Then by the inductive hypothesis, we can show the equivalence class represented by f does not satisfy *phi*, contradiction.

From right to left. It is straightforward to check $\{\ i \mid i \in \mathsf{I} \wedge \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ (\mathsf{FALL}\ n\ \mathit{phi})\ \}$ is a subset of $\{\ i \mid i \in \mathsf{I} \wedge \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma(\!|n \mapsto a|\!)\ x)\ i)\ \mathit{phi}\ \}$, for any equivalence class a. So the former one is in U implies the later one is in U.

$\square$

The way we stated the Los theorem will looks not such satisfactory, since it seems to restrict us for the choice of representatives, but it is not. We can be very free with our choice of representatives, as indicated in the following result:

Lemma 11 (`ultraproduct_rep_independence_lemma`).

$$\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I\ \Rightarrow \\
\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num})\ \subseteq\ \mathsf{ultraproduct}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS})\ \Rightarrow \\
\quad \forall\, \mathit{phi}\ \mathit{rv}. \\
\qquad (\forall\, v.\ v\ \in\ \mathsf{FV}\ \mathit{phi}\ \Rightarrow\ \mathit{rv}\ v\ \in\ \sigma\ v)\ \Rightarrow \\
\qquad (\{\ i\ /\ i\ \in\ I\ \wedge\ \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, x.\ \mathsf{CHOICE}\ (\sigma\ x)\ i)\ \mathit{phi}\ \}\ \in\ U\ \iff \\
\qquad\ \ \{\ i\ /\ i\ \in\ I\ \wedge\ \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, v.\ \mathit{rv}\ v\ i)\ \mathit{phi}\ \}\ \in\ U)
\end{array}$$

If the world of models in FMS are of type β, then rv here is of type $\mathit{num} \to \beta$. The σ here is an assignment of variable symbols to worlds of the ultraproduct model, which are equivalence classes. And rv here assigns each variable symbol v an element $\mathit{rv}\ v$ in the equivalence class assigned to v by σ.For its proof, it actually amounts to check representative independency, which is of the same flavor of the second base case in the proof of Los theorem.

The independence of representative provide us convenience of actually putting our Los theorem into usage. Since then if we want to find an assignments of elements in a ultraproduct model satisfying a first order formula ϕ, instead of assigning equivalence classes directly, it suffices to use the rv below to assign each v a suitable representative functions in the Cartesian product :

Proposition 90 (`ultraproduct_suffices_rep`).

$$\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I\ \Rightarrow \\
\quad (\forall\, i\ \mathit{ff}\ \mathit{ll}.\ i\ \in\ I\ \Rightarrow\ (\mathit{FMS}\ i).\mathsf{Fun}\ \mathit{ff}\ \mathit{ll}\ \in\ (\mathit{FMS}\ i).\mathsf{Dom})\ \Rightarrow \\
\quad \forall\, \mathit{rv}. \\
\qquad (\forall\, v\ i.\ i\ \in\ I\ \Rightarrow\ \mathit{rv}\ v\ i\ \in\ (\mathit{FMS}\ i).\mathsf{Dom})\ \Rightarrow \\
\qquad \forall\, \mathit{phi}. \\
\qquad\quad \{\ i\ /\ i\ \in\ I\ \wedge\ \mathsf{feval}\ (\mathit{FMS}\ i)\ (\lambda\, v.\ \mathit{rv}\ v\ i)\ \mathit{phi}\ \}\ \in\ U\ \Rightarrow \\
\qquad\quad \mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS}) \\
\qquad\quad\ \ (\lambda\, v.\ \{\ g\ /\ \mathsf{Uequiv}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS})\ g\ (\mathit{rv}\ v)\ \}\,)\ \mathit{phi}
\end{array}$$

With the help of `ultraproduct_rep_independence_lemma`, we can prove a classical corollary of Los theorem, which says any first order model $\mathfrak{M}$ is embedded in any of its ultraproduct model by sending a world to the equivalence class represented by the constant function on that world:

Corollary 2 (`corollary_A_21`).

$$
\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I\ \Rightarrow \\
\quad (\forall\, i.\ i\ \in\ I\ \Rightarrow\ \mathit{FMS}\ i\ =\ \mathit{FM})\ \Rightarrow \\
\quad (\forall\, i\ \mathit{ff}\ \mathit{ll}.\ i\ \in\ I\ \Rightarrow\ \mathit{FM}.\mathsf{Fun}\ \mathit{ff}\ \mathit{ll}\ \in\ \mathit{FM}.\mathsf{Dom})\ \Rightarrow \\
\quad \forall\, \sigma. \\
\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathsf{num})\ \subseteq\ \mathit{FM}.\mathsf{Dom}\ \Rightarrow \\
\qquad \forall\, \mathit{phi}. \\
\qquad\quad \mathsf{feval}\ \mathit{FM}\ \sigma\ \mathit{phi}\ \iff \\
\qquad\quad\ \ \mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS}) \\
\qquad\qquad\quad (\lambda\, x.\ \{\ g\ \mid\ \mathsf{Uequiv}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS})\ g\ (\lambda\, i.\ \sigma\ x)\ \}\,)\ \mathit{phi}
\end{array}
$$

All the construction we done above serves to paving a way to getting a countably saturated model:

Lemma 12 (`lemma_2_73`).

$$
\begin{array}{l}
\vdash (\forall\, i.\ i\ \in\ I\ \Rightarrow\ (\mathit{MS}\ i).\mathsf{frame}.\mathsf{world}\ \neq\ \emptyset)\ \Rightarrow \\
\quad \mathsf{countably\_incomplete}\ U\ I\ \Rightarrow \\
\quad \mathsf{countably\_saturated}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ \mathit{MS}))
\end{array}
$$

With all the setups about ultraproduct models, we may feel confident that this will be a consequence of Los theorem. But if we take a closer look of it, we will find out the Los theorem cannot be directly applied here. The definition of countably saturated asks us to prove the realisation of a set of first order formulas in an expanded first order model, and the first order model itself is from turning a modal model into a first order model. The obstrucles here will become clear when we compare what we want to prove to the statement of Los theorem: The Los theorem tells us the result for an ultraproduct of first order models, and says nothing about expansion. But we are proving a statement for an expanded model obtained from viewing an ultraproduct of modal models as a first order model. However, as we shell see now, it cannot stop us from applying the Los theorem.

The first issue is to remove the expansion on the outmost layer of expansion. The key observation is that we have an alternative approach to capture the idea of 'constants'. Constant are nothing more than forcing some symbols to be sent to some point in a model under any valuation, hence rather then use nullary function symbols, we fixed a set of variable letters, each corresponds to a function symbol, and only consider the valuations that sends these variable letters to fixed certain points. With this idea, we can remove all the constants in a formula, and hence swap from an expanded model back to the unexpanded model. For our aim for proving `lemma_2_73`, we only care about the case that we only have finitely many function symbols. To get rid of n-function symbols, we need to come up with n fresh variable symbols to be sent to certain places. The easiest way to ensure that the variable symbols we use for constants do not clash with the variable symbols we use for usual aim is to force those n-variable symbols to be represented by $0, \cdots, n-1$, and then add n to all the variable symbols which originally appear in a formula. The construction of discarding constants in a formula can be done by a function:

Definition 90 (Shifting on terms and formulas).

$$\begin{aligned}
&\mathsf{shift\_term}\ n\ (\mathsf{fV}\ m) \overset{\text{def}}{=} \mathsf{fV}\ (m\ +\ n)\\
&\mathsf{shift\_term}\ n\ (\mathsf{Fn}\ m\ l) \overset{\text{def}}{=}\\
&\quad \textit{if}\ l\ =\ []\ \textit{then}\ \mathsf{fV}\ m\ \textit{else}\ \mathsf{Fn}\ m\ (\mathsf{MAP}\ (\lambda\, a.\ \mathsf{shift\_term}\ n\ a)\ l)
\end{aligned}$$

$$\begin{aligned}
&\mathsf{shift\_form}\ n\ \mathsf{fFALSE} \overset{\text{def}}{=} \mathsf{fFALSE}\\
&\mathsf{shift\_form}\ n\ (\mathsf{Pred}\ m\ l) \overset{\text{def}}{=} \mathsf{Pred}\ m\ (\mathsf{MAP}\ (\mathsf{shift\_term}\ n)\ l)\\
&\mathsf{shift\_form}\ n\ (f_1\ \to\ f_2) \overset{\text{def}}{=} \mathsf{shift\_form}\ n\ f_1\ \to\ \mathsf{shift\_form}\ n\ f_2\\
&\mathsf{shift\_form}\ n\ (\mathsf{FALL}\ x\ f) \overset{\text{def}}{=} \mathsf{FALL}\ (x\ +\ n)\ (\mathsf{shift\_form}\ n\ f)
\end{aligned}$$

Here we will need to prove the termination for the function `shift_term`. The termination relation is given by $\mathsf{measure}\ (\mathsf{term\_size}\ \circ\ \mathsf{SND})$.

We get the syntactical result we want: All the function symbols are removed by applying the shifting construction, and the set of free variables is also under control.

Proposition 91 (Useful consequences of shifting).

$$\begin{aligned}
&\vdash (\forall\, c.\ c\ \in\ \mathsf{form\_functions}\ phi\ \Rightarrow\ \mathsf{FST}\ c\ \in\ \mathsf{count}\ (\mathsf{CARD}\ A)\ \wedge\ \mathsf{SND}\ c\ =\ 0)\ \Rightarrow\\
&\quad \mathsf{form\_functions}\ (\mathsf{shift\_form}\ (\mathsf{CARD}\ A)\ phi)\ =\ \emptyset
\end{aligned}$$

$$\begin{aligned}
&\vdash \mathsf{FV}\ phi\ \subseteq\ s\ \wedge\\
&\quad (\forall\, c.\ c\ \in\ \mathsf{form\_functions}\ phi\ \Rightarrow\ \mathsf{FST}\ c\ \in\ \mathsf{count}\ (\mathsf{CARD}\ A)\ \wedge\ \mathsf{SND}\ c\ =\ 0)\ \Rightarrow\\
&\quad \mathsf{FV}\ (\mathsf{shift\_form}\ (\mathsf{CARD}\ A)\ phi)\ \mathsf{DIFF}\ \mathsf{count}\ (\mathsf{CARD}\ A)\ \subseteq\ \{\ x\ +\ \mathsf{CARD}\ A\ /\ x\ \in\ s\ \}
\end{aligned}$$

Now if we want to use an arbitary valuation to evaluate a shifted formula, something will go wrong, because $0, \cdots, n-1$ are now designed to be sent to fixed place $f\ 0, \cdots, f\ (n-1)$, it does not make sense to assign it to anywhere else. Hence to talk about valuations, the first thing is to make sure that they sends the variables which acturally denotes constants to the right place, therefore, we need to shift them accordingly:

Definition 91 (Shifting on valuations).

$$\mathsf{shift\_valuation}\ n\ \sigma\ f \overset{\text{def}}{=} (\lambda\, m.\ \textit{if}\ m\ <\ n\ \textit{then}\ f\ m\ \textit{else}\ \sigma\ (m\ -\ n))$$

Recall our aim is to get rid of constants and hence avoid talking about expansion of models, the following proposition proves our construction does a good job:

Proposition 92 (`expansion_shift_feval`).

$$\begin{aligned}
&\vdash \mathsf{expansion}\ (\mathsf{mm2folm}\ \mathfrak{M})\ A\ \mathfrak{M}'\ f\ \Rightarrow\\
&\quad \forall\, phi.\\
&\qquad (\forall\, c.\ c\ \in\ \mathsf{FC}\ phi\ \Rightarrow\ c\ <\ \mathsf{CARD}\ A)\ \Rightarrow\\
&\qquad \forall\, \sigma.\\
&\qquad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ \mathfrak{M}.\mathit{frame.world}\ \Rightarrow\\
&\qquad\quad (\mathsf{feval}\ \mathfrak{M}'\ \sigma\ phi\ \iff\\
&\qquad\qquad \mathsf{feval}\ (\mathsf{mm2folm}\ \mathfrak{M})\ (\mathsf{shift\_valuation}\ (\mathsf{CARD}\ A)\ \sigma\ f)\\
&\qquad\qquad\quad (\mathsf{shift\_form}\ (\mathsf{CARD}\ A)\ phi))
\end{aligned}$$

The shifting construction get us out of the expansion, leaving us in a model obtained by coverting a ultraproduct modal model to a first model. How can we apply Los theorem to it? We do have a modal version of Los theorem, but we are not restricted to talking about standard translations, so `Los_modal_thm` is unhelpful. The correct way to apply first order version Los theorem on modal ultraproduct model is actually prove the ultraproduct construction on modal and first order models are capatible, in the sense that they are equivalent when we talk about formulas which makes sense to both of them. The capitability results are also witnesses of our success on construction of ultraproduct models.

Firstly, if we start with a family of modal models and take their ultraproduct in modal sense, then turn the resultant model into a first order model, then this model satisfies the same well-formed formula as the model we get by firstly turning the family of modal models into a family of first order models, then apply the ultraproduct construction in first order sense.

Proposition 93 (`ultraproduct_comm_feval`).

$$
\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I\ \Rightarrow \\
\quad \mathsf{form\_functions}\ \mathit{phi}\ =\ \emptyset\ \Rightarrow \\
\quad \forall\, \sigma. \\
\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ \mathsf{ultraproduct}\ U\ I\ (\mathsf{models2worlds}\ \mathit{MS})\ \Rightarrow \\
\qquad (\mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ \mathit{MS}))\ \sigma\ \mathit{phi}\ \iff \\
\qquad\quad \mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ (\lambda\, i.\ \mathsf{mm2folm}\ (\mathit{MS}\ i)))\ \sigma\ \mathit{phi})
\end{array}
$$

Moreover, for a family of well-formed first order models, if we take their ultraproducts of them in first order sense, it satisfies the same well-formed formulas as the model we get by firstly regard this family as a family of modal models, take their ultraproduct in modal sense, and then turn the resultant ultraproduct modal model into a first order model.

Proposition 94 (`ultraproduct_comm_feval'`).

$$
\begin{array}{l}
\vdash \mathsf{ultrafilter}\ U\ I\ \Rightarrow \\
\quad \mathsf{form\_functions}\ \mathit{phi}\ =\ \emptyset\ \Rightarrow \\
\quad (\forall\, i\ n.\ i\ \in\ I\ \Rightarrow\ ((\mathit{MS}\ i).\mathit{Pred}\ n\ []\ \iff\ \mathsf{F}))\ \Rightarrow \\
\quad (\forall\, i\ a\ b\ n.\ i\ \in\ I\ \Rightarrow\ (\mathit{MS}\ i).\mathit{Pred}\ n\ [a;\ b]\ \Rightarrow\ n\ =\ 0)\ \Rightarrow \\
\quad (\forall\, i\ a\ b\ c\ d\ n.\ i\ \in\ I\ \Rightarrow\ ((\mathit{MS}\ i).\mathit{Pred}\ n\ (a\ ::\ b\ ::\ c\ ::\ d)\ \iff\ \mathsf{F}))\ \Rightarrow \\
\quad (\forall\, n_0\ l_0\ i.\ i\ \in\ I\ \Rightarrow\ (\mathit{MS}\ i).\mathit{Fun}\ n_0\ l_0\ \in\ (\mathit{MS}\ i).\mathit{Dom})\ \Rightarrow \\
\quad \forall\, \sigma. \\
\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ \mathsf{ultraproduct}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{MS})\ \Rightarrow \\
\qquad (\mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{MS})\ \sigma\ \mathit{phi}\ \iff \\
\qquad\quad \mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ (\lambda\, i.\ \mathsf{folm2mm}\ (\mathit{MS}\ i))))\ \sigma\ \mathit{phi})
\end{array}
$$

Further more, with the capitability lemmas and `corollary_A_21`, here is another version of the embedding lemma to ultraproduct models:

Proposition 95 (`ultraproduct_mm2folm_folm2mm_comm_feval`).

$$\begin{array}{l}
\vdash \mathsf{FV}\ a \subseteq \{\, x \,\} \wedge \mathsf{form\_functions}\ a = \emptyset \wedge \mathsf{ultrafilter}\ U\ I \wedge (\forall n.\ \neg\mathfrak{M}.\mathit{Pred}\ n\ []) \wedge \\
\quad (\forall a\ b\ n.\ \mathfrak{M}.\mathit{Pred}\ n\ [a;\ b] \Rightarrow n = 0) \wedge (\forall a\ b\ c\ d\ n.\ \neg\mathfrak{M}.\mathit{Pred}\ n\ (a :: b :: c :: d)) \wedge \\
\quad (\forall \mathit{ff}\ \mathit{ll}.\ \mathfrak{M}.\mathit{Fun}\ \mathit{ff}\ \mathit{ll} \in \mathfrak{M}.\mathit{Dom}) \wedge \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathfrak{M}.\mathit{Dom} \Rightarrow \\
\quad (\mathsf{feval}\ \mathfrak{M}\ \sigma\ a \iff \\
\quad\quad \mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ (\lambda\, i.\ \mathsf{folm2mm}\ \mathfrak{M}))) \\
\quad\quad\quad (\lambda\, x.\ \{\, \mathit{fw} \mid \mathsf{Uequiv}\ U\ I\ (\mathsf{models2worlds}\ (\lambda\, i.\ \mathsf{folm2mm}\ \mathfrak{M}))\ (\lambda\, i.\ \sigma\ x)\ \mathit{fw} \,\}) \\
\quad\quad\quad a)
\end{array}$$

Actually, the `ultraproduct_comm_feval` and `expansion_shift_feval` reduces our task into prove:

Lemma 13 (`ultraproduct_sat`).

$$\begin{array}{l}
\vdash \mathsf{countably\_incomplete}\ U\ I \Rightarrow \\
\quad \forall f. \\
\quad\quad \mathsf{IMAGE}\ f\ \mathcal{U}(:\mathit{num}) \subseteq \mathsf{ultraproduct}\ U\ I\ (\mathsf{folmodels2Doms}\ \mathit{FMS}) \Rightarrow \\
\quad\quad (\forall i\ \mathit{ff}\ \mathit{ll}.\ i \in I \Rightarrow (\mathit{FMS}\ i).\mathit{Fun}\ \mathit{ff}\ \mathit{ll} \in (\mathit{FMS}\ i).\mathit{Dom}) \Rightarrow \\
\quad\quad \forall s. \\
\quad\quad\quad (\forall \mathit{phi}.\ \mathit{phi} \in s \Rightarrow \mathsf{form\_functions}\ \mathit{phi} = \emptyset \wedge \mathsf{FV}\ \mathit{phi}\ \mathsf{DIFF}\ N \subseteq \{\, x \,\}) \Rightarrow \\
\quad\quad\quad (\forall \mathit{ss}. \\
\quad\quad\quad\quad \mathsf{FINITE}\ \mathit{ss} \wedge \mathit{ss} \subseteq s \Rightarrow \\
\quad\quad\quad\quad \exists \sigma. \\
\quad\quad\quad\quad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS}).\mathit{Dom} \wedge \\
\quad\quad\quad\quad\quad (\forall n.\ n \in N \Rightarrow \sigma\ n = f\ n) \wedge \\
\quad\quad\quad\quad\quad \forall \mathit{phi}. \\
\quad\quad\quad\quad\quad\quad \mathit{phi} \in \mathit{ss} \Rightarrow \\
\quad\quad\quad\quad\quad\quad \mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS})\ \sigma\ \mathit{phi}) \Rightarrow \\
\quad\quad\quad \exists \sigma. \\
\quad\quad\quad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS}).\mathit{Dom} \wedge \\
\quad\quad\quad\quad (\forall n.\ n \in N \Rightarrow \sigma\ n = f\ n) \wedge \\
\quad\quad\quad\quad \forall \mathit{phi}.\ \mathit{phi} \in s \Rightarrow \mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ I\ \mathit{FMS})\ \sigma\ \mathit{phi}
\end{array}$$

Proof. Suppose s is a set of formulas such that each element has no function symbol and only one free varible x other then the ones in N which are actually used to capture constants. Suppose in addition that for all finite $ss \subseteq s$, exists a valuation σ such that $\sigma\ n = f\ n$ for all $n \in N$, and $\mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ \mathit{phi}$ for each $\mathit{phi} \in s$. If s is finite, there is nothing to prove, so we assume s is infinite. As we are using a countable first order language, there exists a bijection enum from the natural number to the set s. We prove there exists a valuation σ such that agree with f on N and moreover, $\mathsf{feval}\ (\mathsf{ultraproduct\_folmodel}\ U\ \mathsf{I}\ \mathit{FMS})\ (\mathit{enum}\ n)$ for all natural number n. Such a valuation is an assignment of variables to equivalence class, by `ultraproduct_-suffices_rep` and the Los theorem, it suffices to find out a function rv that assigning each natural number a representative of some equivalence class, such that it satisfies:

- $\forall\, v\ i.\ i\ \in\ \mathsf{I}\ \Rightarrow\ rv\ v\ i\ \in\ (FMS\ i).\mathsf{Dom}$
- $\forall\, n.\ n\ \in\ N\ \Rightarrow\ \{\ g\ \mid\ \mathsf{Uequiv}\ U\ \mathsf{I}\ (\mathsf{folmodels2Doms}\ FMS)\ g\ (rv\ n)\ \}\ =\ f\ n$
- $\forall\, k.\ \{\ i\ \mid\ i\ \in\ \mathsf{I}\ \wedge\ \mathsf{feval}\ (FMS\ i)\ (\lambda\, v.\ rv\ v\ i)\ (conj\ k)\ \}\ \in\ U$

The first item says what rv assign to each natural number must be an element in the Cartesion product. And the second item says that the equivalence class assigned to free variables in N has already been fixed by f. We devote to satisfy the third condition.

By `countbly_incomplete_chain` proved in interlude II, we have a chain In where $In\ n\ \in\ U$ and $In\ (n\ +\ 1)\ \subseteq\ In\ n$ for each n, and moreover, the intersection of this chain is empty. Let $conj\ =$ $\mathsf{PRIM\_REC}\ \mathsf{True}\ (\lambda\, conjn\ n.\ \mathsf{fAND}\ conjn\ (enum\ n))$, hence $conj\ 0\ =\ \mathsf{True}$, and $conj\ n$ is the conjunction from $enum\ 0$ to $enum\ (n\ -\ 1)$. Define $Jn =$
$(\lambda\, n.$
$\{\, i\ \mid$
$i \in \mathsf{I} \wedge \forall\, \sigma.\ (\forall\, k.\ k \in N \Rightarrow \sigma\ k = \mathsf{CHOICE}\ (f\ k)\ i) \Rightarrow \mathsf{feval}\ (FMS\ i)\ \sigma\ (\mathsf{fEXISTS}\ x\ (conj\ n))\,\})$, then Jn is also a descending chain. And moreover, the Los theorem implies that $Jn\ n\ \in\ U$ for any n. Define $Xn\ =$ $(\lambda\, n.\ In\ n\ \cap\ Jn\ n)$, then Xn is a descending chain in U starting with I and intersects to the empty set. For such a chain, each element $i\ \in\ \mathsf{I}$ can only belong to finitely many of Xns. Hence there exists a function Ni that send an element i to smallest set in the chain that i belongs to. That is, for all $i\ \in\ \mathsf{I}\ i\ \in\ Xn\ (Ni\ i)$ and $i\ \notin\ Xn\ a$ for any $a\ >\ Ni\ i$.

The rv we are looking for can be taken as: $\lambda\, v\ i.$
$\texttt{if}\ v \in N\ \texttt{then}\ \mathsf{CHOICE}\ (f\ v)\ i$
$\texttt{else}$
CHOICE
$\{\, a\ \mid\ a \in (FMS\ i).\mathsf{Dom} \wedge \mathsf{feval}\ (FMS\ i)\ (\lambda\, n.\ \texttt{if}\ n \in N\ \texttt{then}\ \mathsf{CHOICE}\ (f\ n)\ i\ \texttt{else}\ a)\ (conj\ (Ni\ i))\,\}$ The first two conditions are immediate to check. It remains to show $\{\ i\ \mid\ i\ \in\ In\ 0 \wedge \mathsf{feval}\ (FMS\ i)\ (\lambda\, v.\ \sigma r\ v\ i)\ (conj\ k)\ \}\ \in$ U for any k. As $Xn\ k$ is in U, it suffices to check this is a supset of $Xn\ k$. For any $i\ \in\ Xn\ k$, by definition of the function Ni, we have $k\ \leq\ Ni\ i$. As $i\ \in\ Xn\ (Ni\ i)$, in particular, $i\ \in\ Jn\ (Ni\ i)$. Hence $\mathsf{feval}\ (FMS\ i)\ (\lambda\, v.\ \sigma r\ v\ i)\ (conj\ (Ni\ i))$. As $conj\ m$ implies $conj\ n$ for $n\ \leq\ m$, we are done.

□

We can step on the two stairs we built above to get `lemma_2_73`, using the capatibility lemma, we migrant the above lemma to ultraproduct of modal models:

Lemma 14 (`ultraproduct_sat'`).

$$
\begin{array}{l}
\vdash \mathsf{countably\_incomplete}\ U\ I\ \Rightarrow \\
\quad (\forall\, i.\ i\ \in\ I\ \Rightarrow\ (MS\ i).\mathit{frame}.\mathit{world}\ \neq\ \emptyset)\ \Rightarrow \\
\quad \mathsf{IMAGE}\ f\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ \mathsf{ultraproduct}\ U\ I\ (\mathsf{models2worlds}\ MS)\ \Rightarrow \\
\quad \forall\, s. \\
\quad\quad (\forall\, phi.\ phi\ \in\ s\ \Rightarrow\ \mathsf{form\_functions}\ phi\ =\ \emptyset\ \wedge\ \mathsf{FV}\ phi\ \mathsf{DIFF}\ N\ \subseteq\ \{\ x\ \})\ \Rightarrow \\
\quad\quad (\forall\, ss. \\
\quad\quad\quad \mathsf{FINITE}\ ss\ \wedge\ ss\ \subseteq\ s\ \Rightarrow \\
\quad\quad\quad \exists\, \sigma. \\
\quad\quad\quad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS)).\mathit{Dom}\ \wedge \\
\quad\quad\quad\quad (\forall\, n.\ n\ \in\ N\ \Rightarrow\ \sigma\ n\ =\ f\ n)\ \wedge \\
\quad\quad\quad\quad \forall\, phi. \\
\quad\quad\quad\quad\quad phi\ \in\ ss\ \Rightarrow \\
\quad\quad\quad\quad\quad \mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS))\ \sigma\ phi)\ \Rightarrow \\
\quad\quad \exists\, \sigma. \\
\quad\quad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS)).\mathit{Dom}\ \wedge \\
\quad\quad\quad (\forall\, n.\ n\ \in\ N\ \Rightarrow\ \sigma\ n\ =\ f\ n)\ \wedge \\
\quad\quad\quad \forall\, phi.\ phi\ \in\ s\ \Rightarrow\ \mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS))\ \sigma\ phi
\end{array}
$$

And use the shifting, we then migrant to expanded models:

Lemma 15 (`ultraproduct_sat''`).

$$
\begin{array}{l}
\vdash \mathsf{countably\_incomplete}\ U\ I\ \Rightarrow \\
\quad (\forall i.\ i\ \in\ I\ \Rightarrow\ (MS\ i).\mathit{frame}.\mathit{world}\ \neq\ \emptyset)\ \Rightarrow \\
\quad \forall A\ \mathfrak{M}'\ f. \\
\qquad \mathsf{expansion}\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS))\ A\ \mathfrak{M}'\ f\ \Rightarrow \\
\qquad \mathsf{IMAGE}\ f\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ \mathsf{ultraproduct}\ U\ I\ (\mathsf{models2worlds}\ MS)\ \Rightarrow \\
\qquad \forall s. \\
\qquad\quad (\forall phi. \\
\qquad\qquad phi\ \in\ s\ \Rightarrow \\
\qquad\qquad (\forall c. \\
\qquad\qquad\quad c\ \in\ \mathsf{form\_functions}\ phi\ \Rightarrow \\
\qquad\qquad\quad \mathsf{FST}\ c\ \in\ \mathsf{count}\ (\mathsf{CARD}\ A)\ \wedge\ \mathsf{SND}\ c\ =\ 0)\ \wedge\ \mathsf{FV}\ phi\ \subseteq\ \{\ x\ \})\ \Rightarrow \\
\qquad\quad (\forall ss. \\
\qquad\qquad \mathsf{FINITE}\ ss\ \wedge\ ss\ \subseteq\ s\ \Rightarrow \\
\qquad\qquad \exists \sigma. \\
\qquad\qquad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq \\
\qquad\qquad\quad (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS)).\mathit{Dom}\ \wedge \\
\qquad\qquad\quad \forall phi.\ phi\ \in\ ss\ \Rightarrow\ \mathsf{feval}\ \mathfrak{M}'\ \sigma\ phi)\ \Rightarrow \\
\qquad\quad \exists \sigma. \\
\qquad\qquad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\ \mathit{num})\ \subseteq\ (\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ I\ MS)).\mathit{Dom}\ \wedge \\
\qquad\qquad \forall phi.\ phi\ \in\ s\ \Rightarrow\ \mathsf{feval}\ \mathfrak{M}'\ \sigma\ phi
\end{array}
$$

we can deduce `lemma_2_73` trivially from here.

With the help of `lemma_2_73`, we prove the critical theorem:

Theorem 16 (`thm_2_74_half_2`).

$$
\begin{array}{l}
\vdash w\ \in\ \mathfrak{M}.\mathit{frame}.\mathit{world}\ \wedge\ v\ \in\ N.\mathit{frame}.\mathit{world}\ \Rightarrow \\
\quad (\forall phi.\ \mathsf{satis}\ \mathfrak{M}\ w\ phi\ \iff\ \mathsf{satis}\ N\ v\ phi)\ \Rightarrow \\
\quad \exists U\ I. \\
\qquad \mathsf{ultrafilter}\ U\ I\ \wedge \\
\qquad \mathsf{bisim\_world}\ (\mathsf{ultraproduct\_model}\ U\ I\ (\lambda i.\ \mathfrak{M})) \\
\qquad\ (\mathsf{ultraproduct\_model}\ U\ I\ (\lambda i.\ N)) \\
\qquad\ \{\ fw\ \mid\ \mathsf{Uequiv}\ U\ I\ (\mathsf{models2worlds}\ (\lambda i.\ \mathfrak{M}))\ (\lambda i.\ w)\ fw\ \} \\
\qquad\ \{\ fv\ \mid\ \mathsf{Uequiv}\ U\ I\ (\mathsf{models2worlds}\ (\lambda i.\ N))\ (\lambda i.\ v)\ fv\ \}
\end{array}
$$

Proof. The U we require here is any countably incomplete ultrafilter, we have proved such ultrafilter exists in Interlude II. Then by `lemma_2_73`, the models $\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ \mathsf{I}\ (\lambda i.\ \mathfrak{M}))$, $\mathsf{mm2folm}\ (\mathsf{ultraproduct\_model}\ U\ \mathsf{I}\ (\lambda i.$ are countably incomplete. Hence we are done by `thm_2_65_corollary`. □

The last step towards the main theorem is used to pass from infinite set to finite set.

5.1.2 Interlude IV: Compactness theorem

The compactness theorem is formlised in (reference), the statement in HOL is:

Theorem 17 (Compactness of first order logic)**.**

$$\begin{array}{l} \vdash (\forall\, t. \\ \quad \mathsf{FINITE}\ t \ \wedge\ t \subseteq s \ \Rightarrow \\ \quad \exists\, \mathfrak{M}.\ \mathsf{interpretation}\ (\mathsf{language}\ s)\ \mathfrak{M} \ \wedge\ \mathfrak{M}.\mathit{Dom} \neq \emptyset \ \wedge\ \mathfrak{M}\ \mathsf{satisfies}\ t) \ \Rightarrow \\ \ \exists\, C.\ \mathsf{interpretation}\ (\mathsf{language}\ s)\ C \ \wedge\ C.\mathit{Dom} \neq \emptyset \ \wedge\ C\ \mathsf{satisfies}\ s \end{array}$$

For more about its proof, see (reference). Here we only spell out what did else have we done.

The compactness theorem has a useful corollary:

Corollary 3 (`compactness_corollary`)**.**

$$\begin{array}{l} \vdash (\forall\, \mathfrak{M}\ \sigma.\ \mathsf{valuation}\ \mathfrak{M}\ \sigma \ \Rightarrow\ (\forall\, f.\ f \in s \Rightarrow \mathit{feval}\ \mathfrak{M}\ \sigma\ f) \Rightarrow \mathit{feval}\ \mathfrak{M}\ \sigma\ \mathit{phi}) \Rightarrow \\ \quad \forall\, ss. \\ \qquad \mathsf{FINITE}\ ss \ \wedge\ ss \subseteq s \ \wedge \\ \qquad \forall\, \mathfrak{M}\ \sigma.\ \mathsf{valuation}\ \mathfrak{M}\ \sigma \ \Rightarrow\ (\forall\, f.\ f \in s \Rightarrow \mathit{feval}\ \mathfrak{M}\ \sigma\ f) \Rightarrow \mathit{feval}\ \mathfrak{M}\ \sigma\ \mathit{phi} \end{array}$$

Proof. Talk about its proof once I can understand it... □

From the compactness for first order models, we can prove a modal version of compactness theorem using the relation between modal and first order logic, stating that if any finite subset of a set of modal formulas is realised by a modal model, then the whole set is realised by a modal model.

Theorem 18 (Compactness of modal logic)**.**

$$\begin{array}{l} \vdash (\forall\, ss. \\ \quad \mathsf{FINITE}\ ss \ \wedge\ ss \subseteq s \ \Rightarrow \\ \quad \exists\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \wedge \forall\, f.\ f \in ss \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f) \Rightarrow \\ \ \exists\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \wedge \forall\, f.\ f \in s \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f \end{array}$$

Proof. Under the assumptions, let $A = \{\ \mathsf{ST}\ x\ f \mid f \in s\ \}$. By `prop_2_24_i`, each finite subset of A is realised by some first order model. Hence A is realised by a first order model $\mathfrak{M}$. By `prop_2_47_i'`, $\mathsf{folm2mm}\ \mathfrak{M}$ realises all the modal formulas in s.

□

As in the first order case, the modal version of compactness theorem has a similar corollary:

Corollary 4 (`modal_compactness_corollary`)**.**

$$\begin{array}{l} \vdash (\forall\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\forall\, f.\ f \in s \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f) \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ a) \Rightarrow \\ \quad \exists\, ss. \\ \qquad \mathsf{FINITE}\ ss \ \wedge\ ss \subseteq s \ \wedge \\ \qquad \forall\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\forall\, f.\ f \in ss \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f) \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ a \end{array}$$

Proof. Analogue of `compactness_corollary`. □

With all these set-ups, we give a fully justified answer to the question in the begining of the section.

We have a similar corollary for the modal version compactness theorem, with a proof of same flavor as the proof of `compactness_corollary` from `compactness`.

Corollary 5 (`modal_compactness_corollary`).

$$\begin{aligned}&\vdash (\forall \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\forall f.\ f \in s \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f) \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ a) \Rightarrow\\ &\quad \exists\, ss.\\ &\qquad \mathsf{FINITE}\ ss \wedge ss \subseteq s \wedge\\ &\qquad \forall \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\forall f.\ f \in ss \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f) \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ a\end{aligned}$$

Theorem 19 (`thm_2_68_half1`, Van Benthem Characterization Theorem).

$$\begin{aligned}&\vdash \mathsf{FV}\ a \subseteq \{\ x\ \} \wedge \mathsf{form\_functions}\ a = \emptyset \Rightarrow\\ &\quad \mathsf{invar4bisim}\ x\ t_1\ t_2\ a \Rightarrow\\ &\quad \exists\, phi.\\ &\qquad \forall \mathfrak{M}\ \sigma.\\ &\qquad\quad (\forall n.\ \mathfrak{M}.\mathit{Pred}\ n\ [] \iff \mathsf{F}) \wedge (\forall a\ b\ n.\ \mathfrak{M}.\mathit{Pred}\ n\ [a;\ b] \Rightarrow n = 0) \wedge\\ &\qquad\quad (\forall a\ b\ c\ d\ n.\ \mathfrak{M}.\mathit{Pred}\ n\ (a :: b :: c :: d) \iff \mathsf{F}) \wedge\\ &\qquad\quad (\forall n_0\ l_0.\ \mathfrak{M}.\mathit{Fun}\ n_0\ l_0 \in \mathfrak{M}.\mathit{Dom}) \Rightarrow\\ &\qquad\quad \mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathit{num}) \subseteq \mathfrak{M}.\mathit{Dom} \Rightarrow\\ &\qquad\quad (\mathsf{feval}\ \mathfrak{M}\ \sigma\ (\mathsf{ST}\ x\ phi) \iff \mathsf{feval}\ \mathfrak{M}\ \sigma\ a)\end{aligned}$$

Proof. Suppose a is a first order formula invariant for bisimulation with only one free variable x. Consider the modal consequence of a, which is the set of standard translations implied by a on well-formed first order models, defined in the HOL as $MOC =$
$\{\mathsf{ST}\ x\ phi\ |$
$phi\ |$
$\forall \mathfrak{M}\ \sigma.$
$(\forall n.\ \mathfrak{M}.\mathsf{Pred}\ n\ [] \iff \mathsf{F}) \wedge (\forall a\ b\ n.\ \mathfrak{M}.\mathsf{Pred}\ n\ [a;\ b] \Rightarrow n = 0) \wedge$
$(\forall a\ b\ c\ d\ n.\ \mathfrak{M}.\mathsf{Pred}\ n\ (a::b::c::d) \iff \mathsf{F}) \wedge (\forall n_0\ l_0.\ \mathfrak{M}.\mathsf{Fun}\ n_0\ l_0 \in \mathfrak{M}.\mathsf{Dom}) \Rightarrow$
$\mathsf{IMAGE}\ \sigma\ \mathcal{U}(:\mathtt{num}) \subseteq \mathfrak{M}.\mathsf{Dom} \Rightarrow$
$\mathsf{feval}\ \mathfrak{M}\ \sigma\ a \Rightarrow$
$\mathsf{feval}\ \mathfrak{M}\ \sigma\ (\mathsf{ST}\ x\ phi)\}$ According to the discussion about compactness theorem proved as in interlude IV, it suffices to prove a is implies by MOC. Now, fix a model $\mathfrak{M}$ and suppose $\mathsf{feval}\ \mathfrak{M}\ \sigma\ f$ for any $f \in MOC$, we prove $\mathsf{feval}\ \mathfrak{M}\ \sigma\ a$.

Consider of the set Tx of formulas $\mathsf{ST}\ x\ phi$ such that $\mathsf{feval}\ \mathfrak{M}\ \sigma\ (\mathsf{ST}\ x\ phi)$. We prove there is a well-formed model N with an evaluation σn such that for all $f \in Tx \cup \{\ a\ \}$, we have $\mathsf{feval}\ N\ \sigma n\ f$. Suppose, in order to get a contradiction, that such a model does not exists, then for any well-formed model, once all the formulas in Tx are satisfied, then a is not satisfied. Then by compactness, there exists a finite subset of

Tx implies $\neg a$. Taking its contrapositive, then a implies the negation of the big conjunctionpsi of finitely many elements in Tx. By `ST_BIGCONJ` and ST_fNOT, a negated big conjunction of standard translations is again a standard translation. Hence $\mathsf{fNOT}\ psi$ is in MOC. Recall we have assumed $\mathsf{feval}\ \mathfrak{M}\ \sigma\ f$ for any $f \in MOC$, so $\mathsf{feval}\ \mathfrak{M}\ \sigma\ (\mathsf{fNOT}\ psi)$, but also $\mathsf{feval}\ \mathfrak{M}\ \sigma\ psi$ by definition of Tx, we have a contradiction.

Hence we obtain a model N and a valuation σn with desired property. Now let w denote $\sigma\ x$ and v denote $\sigma n\ x$, we claim that if we regard both $\mathfrak{M}$ and N as modal models, then w and v are modal equivalent. To prove this, suppose $\mathsf{satis}\ (fol2mm\ \mathfrak{M})\ w\ phi$, then $\mathsf{ST}\ x\ phi$ is in Tx by definition of Tx, hence $\mathsf{feval}\ N\ \sigma n\ (\mathsf{ST}\ x\ phi)$. By `mm2folm_folm2mm_feval`, this is equivalent to $\mathsf{feval}\ (\mathsf{mm2folm}\ (\mathsf{folm2mm}\ N))\ \sigma n\ (\mathsf{ST}\ x\ phi)$, and if follows from `prop_2_47_i` that $\mathsf{satis}\ (\mathsf{folm2mm}\ N)\ v\ phi$. This proves $\forall\, phi.\ \mathsf{satis}\ (\mathsf{folm2mm}\ \mathfrak{M})\ w\ phi \iff \mathsf{satis}\ (\mathsf{folm2mm}\ N)\ v\ phi$. The other direction is by a symmetry argument.

If modal equivalence implies bisimularity, then we are done: Suppose modal equivalence implies bisimularity, then as $w \in (\mathsf{folm2mm}\ \mathfrak{M}).\mathsf{frame.world}$ and $v \in (\mathsf{folm2mm}\ N).\mathsf{frame.world}$ are modal equivalent, there exists a bisimulation between them. And as a is invariant for bisimulation and is satisfied at v, then it is also satisfied at w.

But actually it is not always the case. Fortunately, we can take a detour with the help of `thm_2_47_half2`. We can embed the worlds w, v into these models $Mst = \mathsf{ultraproduct\_model}\ U\ \mathsf{I}\ (\lambda\, i.\ \mathsf{folm2mm}\ \mathfrak{M})$, $Nst = \mathsf{ultraproduct\_model}\ U\ \mathsf{I}\ (\lambda\, i.\ \mathsf{folm2mm}\ N)$ by $wst = \{\ fw \mid \mathsf{Uequiv}\ U\ \mathsf{I}\ (\mathsf{models2worlds}\ (\lambda\, i.\ \mathsf{folm2mm}\ \mathfrak{M}))\ (\lambda\, i.\ w)\ fw\ \}$, $vst = \{\ fv \mid \mathsf{Uequiv}\ U\ \mathsf{I}\ (\mathsf{models2worlds}\ (\lambda\, i.\ \mathsf{folm2mm}\ N))\ (\lambda\, i.\ v)\ fv\ \}$ respectively, where these two worlds are bisimilar. As a is preserved under simulation, a holds as wst in $\mathsf{mm2folm}\ Mst$ iff it holds at vst in $\mathsf{mm2folm}\ Nst$. We are going to carry the a from the model N where a is satisfied at v, to the point vst in $\mathsf{mm2folm}\ Nst$, then to the point wst in $\mathsf{mm2folm}\ Mst$, and finally to w in $\mathfrak{M}$.

To carry a arount, it suffices to prove $\mathsf{feval}\ \mathfrak{M}\ \sigma\ a \iff \mathsf{fsatis}\ (\mathsf{mm2folm}\ Mst)\ (\lambda\, x.\ wst)\ a$ and $\mathsf{feval}\ N\ \sigma n\ a \iff \mathsf{fsatis}\ (\mathsf{mm2folm}\ Nst)\ (\lambda\, x.\ vst)\ a$. These two are of the same pattern, they holds by `ultraproduct_mm2folm_folm2mm_comm_feval`.

□

5.2 Positive existential formula vs preserved under simulations

In this section, we use a similar proof strategy as we did in `thm_2_68_half1` to present a result about the concept 'half of a bisimulation', which is called 'simulation'.

Definition 92 (Simulation)**.**

$$
\begin{array}{l}
\mathsf{sim}\ Z\ \mathfrak{M}\ \mathfrak{M}' \stackrel{\text{def}}{=} \\
\quad \forall\, w\ w'. \\
\qquad w \in \mathfrak{M}.\mathit{frame.world} \land w' \in \mathfrak{M}'.\mathit{frame.world} \land Z\ w\ w' \Rightarrow \\
\qquad (\forall\, p.\ w \in \mathfrak{M}.\mathit{valt}\ p \Rightarrow w' \in \mathfrak{M}'.\mathit{valt}\ p) \land \\
\qquad \forall\, v. \\
\qquad\quad v \in \mathfrak{M}.\mathit{frame.world} \land \mathfrak{M}.\mathit{frame.rel}\ w\ v \Rightarrow \\
\qquad\quad \exists\, v'.\ v' \in \mathfrak{M}'.\mathit{frame.world} \land Z\ v\ v' \land \mathfrak{M}'.\mathit{frame.rel}\ w'\ v'
\end{array}
$$

The concept corresponds to 'invariant for bisimulation' is called 'preserved by simulation', it takes the types of models as parameters by exactly the same reason why we need those parameters for the definition `invar4bisim`:

Definition 93 (Preserved under simulation)**.**

$$\begin{array}{l} \mathsf{preserved\_under\_sim}\ \mu\ \nu\ phi \overset{\text{def}}{=} \\ \quad \forall \mathfrak{M}\ \mathfrak{M}'\ Z\ w\ w'. \\ \quad\quad w \in \mathfrak{M}.\mathit{frame.world} \wedge w' \in \mathfrak{M}'.\mathit{frame.world} \wedge \mathsf{sim}\ Z\ \mathfrak{M}\ \mathfrak{M}' \wedge Z\ w\ w' \Rightarrow \\ \quad\quad \mathsf{satis}\ \mathfrak{M}\ w\ phi \Rightarrow \\ \quad\quad \mathsf{satis}\ \mathfrak{M}'\ w'\ phi \end{array}$$

The aim of the rest of the section is to characterise formulas preserved under bisimulation as positively existential formulas. A positive existential formula is a modal formula which it either equal to $\top$ or $\bot$, or is built up using only contains positive connectives, namely '$\wedge$','$\vee$' or '$\Diamond$'. Its definition is formalised as an inductive relation.

Definition 94 (Rules of positive existential formulas)**.**

$$\frac{}{\mathsf{PE}\ \bot} \qquad \frac{}{\mathsf{PE\ TRUE}} \qquad \frac{}{\mathsf{PE\ (VAR}\ p)} \qquad \frac{\mathsf{PE}\ f_1 \quad \mathsf{PE}\ f_2}{\mathsf{PE\ (AND}\ f_1\ f_2)} \qquad \frac{\mathsf{PE}\ f_1 \quad \mathsf{PE}\ f_2}{\mathsf{PE\ (DISJ}\ f_1\ f_2)} \qquad \frac{\mathsf{PE}\ f}{\mathsf{PE}\ (\Diamond\ f)}$$

A useful syntactial feature of such formulas is that any finite conjunction or disjunction of positive existential formulas is again a positive existential formula. Again, instead of explicitly define big conjunction and big disjunction, we use the following propositions to captures this idea:

Proposition 96 (`PE_BIGCONJ`)**.**

$$\begin{array}{l} \vdash \mathsf{FINITE}\ ss \Rightarrow \\ \quad (\forall f.\ f \in ss \Rightarrow \mathsf{PE}\ f) \Rightarrow \\ \quad \exists ff. \\ \quad\quad \mathsf{PE}\ ff \wedge \\ \quad\quad \forall \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ ff \iff \forall f.\ f \in ss \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ f) \end{array}$$

Proposition 97 (`PE_BIGDISJ`)**.**

$$\begin{array}{l} \vdash \mathsf{FINITE}\ ss \Rightarrow \\ \quad (\forall f.\ f \in ss \Rightarrow \mathsf{PE}\ f) \Rightarrow \\ \quad \exists ff. \\ \quad\quad \mathsf{PE}\ ff \wedge \\ \quad\quad \forall \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathit{frame.world} \Rightarrow (\mathsf{satis}\ \mathfrak{M}\ w\ ff \iff \exists f.\ f \in ss \wedge \mathsf{satis}\ \mathfrak{M}\ w\ f) \end{array}$$

By induction using `PE_ind`, it is easy to proof half of our main theorem:

Theorem 20 (`thm_2_78_half1`)**.**

$$\begin{array}{l} \vdash \mathsf{PE}\ phi_0 \wedge \mathsf{equiv0}\ \mu\ phi\ phi_0 \wedge \mathsf{equiv0}\ \nu\ phi\ phi_0 \Rightarrow \\ \quad \mathsf{preserved\_under\_sim}\ \mu\ \nu\ phi \end{array}$$

The reverse direction of `thm_2_78_half1_lemma` will get m-saturated to be involved, since m-saturated models are not only good for bisimulations, but also give nice result for simulations.

Proposition 98 (`exercise_2_7_1`).

$$\begin{aligned}&\vdash \mathsf{M\_sat}\,\mathfrak{M} \;\wedge\; \mathsf{M\_sat}\,\mathfrak{M}' \;\wedge\; w \;\in\; \mathfrak{M}.\mathit{frame.world} \;\wedge\; w' \;\in\; \mathfrak{M}'.\mathit{frame.world} \;\Rightarrow\\ &\quad (\forall\, \mathit{phi}.\ \mathsf{PE}\ \mathit{phi} \;\Rightarrow\; \mathsf{satis}\,\mathfrak{M}\ w\ \mathit{phi} \;\Rightarrow\; \mathsf{satis}\,\mathfrak{M}'\ w'\ \mathit{phi}) \;\Rightarrow\\ &\quad \exists\, Z.\ \mathsf{sim}\ Z\ \mathfrak{M}\,\mathfrak{M}' \;\wedge\; Z\ w\ w'\end{aligned}$$

Proof. Under the assumptions, $\lambda\, w_1\ w_2.\ \forall\, \mathit{phi}.\ \mathsf{PE}\ \mathit{phi} \;\Rightarrow\; \mathsf{satis}\ \mathfrak{M}\ w_1\ \mathit{phi} \;\Rightarrow\; \mathsf{satis}\ \mathfrak{M}'\ w_2\ \mathit{phi}$ gives a simulation relation. Checking it is indeed a simulation is completely analogue to the proof of `prop_2_-54`. □

We will take a detour through m-saturated models again, but since this time we are working within a modal language instead of first order logic, we only need to use ultrafilter extensions rather than ultrapowers.

Theorem 21 (`thm_2_78_half2`).

$$\vdash \mathsf{preserved\_under\_sim}\ \nu\ \nu\ \mathit{phi} \;\Rightarrow\; \exists\, \mathit{phi}_0.\ \mathsf{equiv0}\ \mu\ \mathit{phi}\ \mathit{phi}_0 \;\wedge\; \mathsf{PE}\ \mathit{phi}_0$$

At first glance, the statement may looks awkward because the assumption is about preservsion under simulation for models of type $(\beta \to \mathit{bool}) \to \mathit{bool}$, but the conclusion is about equivalent to a formula on models of type β. This is because our detour through ultrafilter extensions, which embeds a model with β-worlds into a model with $(\beta \to \mathit{bool}) \to \mathit{bool}$-worlds. It means that for the usual language statement, although the hypothesis is talking about preserved under simulation for any types, what we actually require in the proof is only the fact that the formula is preserved under simulation for a certain type.

Proof. Suppose ϕ is preserved under simulation. Consider the set of positive existential consequence of ϕ, defined as $\mathit{PEC} = \{\, \mathit{psi} \mid \mathsf{PE}\ \mathit{psi} \wedge \forall\, \mathfrak{M}\ w.\ w \in \mathfrak{M}.\mathsf{frame.world} \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi} \Rightarrow \mathsf{satis}\ \mathfrak{M}\ w\ \mathit{psi} \,\}$ By `modal_-compactness_corollary` proved by the last interlude, if we can prove for any model $\mathfrak{M}$ and $w \;\in\; \mathfrak{M}.\mathsf{frame.world}$, $\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{psi}$ for all $\mathit{psi} \;\in\; \mathit{PEC}$ implies $\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}$, then there exists a finite subset S of PEC that entails ϕ. This will prove ϕ is equivalent to the conjunction of all the formulas in S, which is again a positive existential formula.

So we prove the entailment from PEC to ϕ. Suppose $\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{psi}$ for all $\mathit{psi} \;\in\; \mathit{PEC}$, we prove $\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{phi}$. Define $\mathit{\Gamma} \;=\; \{\ \neg\mathit{psi} \mid \mathsf{PE}\ \mathit{psi} \;\wedge\; \mathsf{satis}\ \mathfrak{M}\ w\ (\neg\mathit{psi})\ \}$. We claim that there exists a model that realises the set $\mathit{\Gamma} \;\cup\; \{\ \mathit{phi}\ \}$. By `modal_compactness_thm`, it suffices to prove each finite subset of $\mathit{\Gamma} \;\cup\; \{\ \mathit{phi}\ \}$ is realised by some model. Suppose there exists a finite subset of $\mathit{\Gamma} \;\cup\; \{\ \mathit{phi}\ \}$ which cannot be realised by any model, then there exists $\neg\psi_0, \cdots, \neg\psi_n \in \Gamma$ such that for any model N and any world v of it, $\mathsf{satis}\ N\ v\ \mathit{psii}$ for some i. By `PE_BIGDISJ`, as all these ψ's are positive existential, there exists a positive existential formula ψ which is satisfied precisely when some ψ_i is satisfied. Hence $\mathit{psi} \;\in\; \mathit{PEC}$. As $\mathfrak{M}$ entails PEC, we have $\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{psi}$. But then it means $\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{psii}$ for some i, but from the definition of the ψ's, $\mathsf{satis}\ \mathfrak{M}\ w\ (\neg\mathit{psii})$ for any ψ_i, this is a contradiction.

Hence we obtain a model N realising $\mathit{\Gamma} \;\cup\; \{\ \mathit{phi}\ \}$ at point v. For any positive existential formula ψ such that $\neg\mathsf{satis}\ \mathfrak{M}\ w\ \mathit{psi}$, we have $\neg\mathit{psi} \;\in\; \mathit{\Gamma}$, so $\mathsf{satis}\ N\ v\ (\neg\mathit{psi})$. Hence for any positive existential

ψ, if satis N v psi, then satis $\mathfrak{M}$ w psi. We migrant this implication onto ultrafilter extensions: Now take the ultrafilter extension of $\mathfrak{M}$ and N respectively. By `prop_2_59_ii`, for any positive existential ψ, satis (UE N) (principle_UF v N.frame.world) psi implies satis (UE $\mathfrak{M}$) (principle_UF w $\mathfrak{M}$.frame.world) psi. As UE $\mathfrak{M}$ and UE N are m-saturated by `prop_2_61`, by `exercise_2_7_1`, we have a simulation linking principle_UF v N.frame.world to principle_UF w $\mathfrak{M}$.frame.world.

As satis N v phi, `prop_2_59_ii` gives satis (UE N) (principle_UF v N.frame.world) phi, as ϕ is preserved under simulation, we have satis (UE $\mathfrak{M}$) (principle_UF w $\mathfrak{M}$.frame.world) phi. Again by `prop_2_59_ii` , it implies satis $\mathfrak{M}$ w phi. This completes the proof. □

6 Conclusion

6.1 Choice we made

- Restriction to basic modal language: In general, we allow more than one modal operator in a modal language and allow them to correspond to relations of any arity. We made this choice because of limitation of time, and the fact that the proofs for the basic modal language is enlightening enough.
- Restrict to the same type to prove the characterisation theorems in the last chapter.
- Use natural number to denote variables for first order logic: This is a choice made in (reference), not by us. But it actually gives us convenience when we want to come up with a fresh variable-just add one, whereas if we allow variables in any type, we may need CHOICE to choose a frsh variable.

6.2 Problems we meet

- There are some issue because of the limitation of plain type theory. (1) Some definitions must take types as its parameter, namely definition of equiv0 fequiv, invar4bisim and *preserve_under_sim*. (2) We cannot talk about the collection of all models at once, and hence Require assumption on universe.
- Well-formedness of first order model.
- We cannot talk about class of models. So the definition we made on Hennessy-Milner class is useless. Also this issue block use from formalising section 2.6, which talks about class of model that 'closed under ultraproduct'.

6.3 What we have done

- The entire construction is an attempt of formalising the book 'Modal Logic' by Blackburn et al. The progress is up to section 2.7. Anything up to section 2.7 that can be captured by the basic modal language and the HOL are formalised. And the proofs are taken as in the book. For the proofs which are omitted or left as exercise, we worked out the exercise and mechanised it in the HOL. And we fill the omitted details. In particular, we give a fully formalised proof of proposition 2.29, which is claimed to be obvious by the textbook.

Due to the type issue, we cannot talk about class of models that 'closed under ultraproduct', so cannot formalise theorem 2.75 and 2.76. Due to the fact that we restricted to the basic modal language, we cannot talk about propositional dynamic logic(PDL) and hence cannot prove theorem 2.84 in the book characterising formulas which are safe under bisimulation as PDL formulas in special pattern.

- The Interludes II can be taken as independent with our theory of modal logic, so does Interlude III which only relies on John Harrison's work on first order logic. These two pieces can be taken as a ground to do more work about ultrafilters, ultraproducts, and their relation to model theory.

  Interlude I does rely on our construction of modal formula, but it is enlightening and with mere effort one can use the same idea to prove any propositional formula is equivalent to a disjunction normal form, which is a 'common sense' that we cannot find a proof anywhere online.

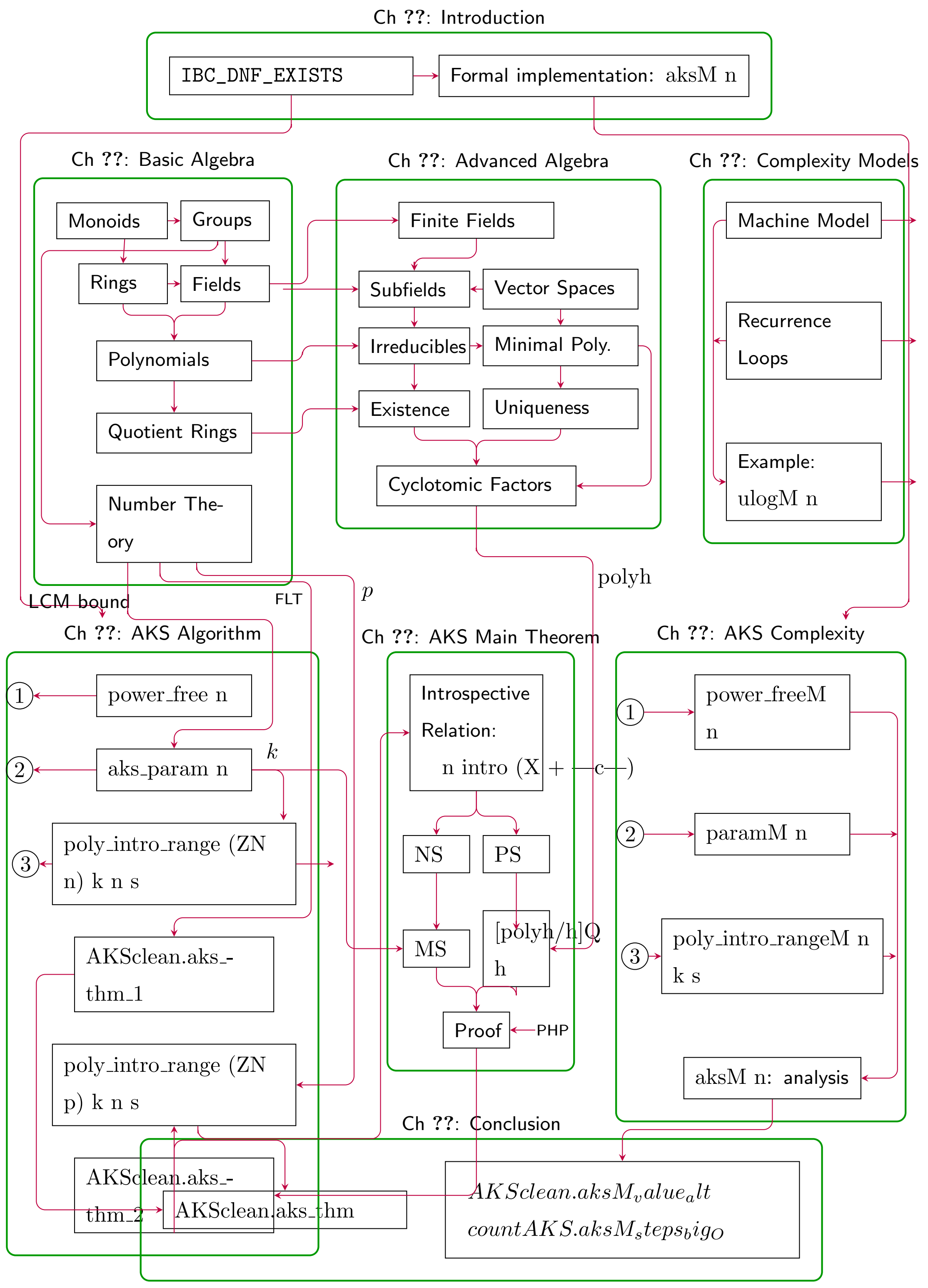
- Initially, before we encounter the ultrafilters, we use natural numbers to encode worlds of any models. Hence we need to use natural numbers to represent lists of natural numbers. In order to be able to do this, we developed `nlistTheory`, which is eventually not used because we cannot restrict to countable world set anymore since ultrafiler models can have uncountable world set. But `nlistTheory` is proved to be useful since it is currently used for formalisation of computability theory to encode list of states in a machine, where we do not need to bother with uncountability.

6.4 Future work

- Try to capture the idea of the decidability part in section 2.6 and formalise it.

- Proceed withe the book. A interesting point is that the formalisation of chapter 3 requires a theory on second-order logic. We can try build a theory on second order logic which interact nicely with the current first order theory and modal model theory.

- Use Interlude II and III as a little library of ultrafilters and ultraproduct to mechanise more about model theory. For instance, the fact that the theory we build can be quoted for proving `lemma_2_73` suggests that we can try formalising more about saturation property of models. A plausible next step is to construct the Keisler's order using the existing theories.


Ch ??: Introduction
IBC_DNF_EXISTS
Formal implementation: aksM n
Ch ??: Basic Algebra
Monoids
Groups
Rings
Fields
Polynomials
Quotient Rings
Number Theory
Ch ??: Advanced Algebra
Finite Fields
Subfields
Vector Spaces
Irreducibles
Minimal Poly.
Existence
Uniqueness
Cyclotomic Factors
Ch ??: Complexity Models
Machine Model
Recurrence Loops
Example: ulogM n
LCM bound
FLT
p
polyh
Ch ??: AKS Algorithm
power_free n
aks_param n
k
poly_intro_range (ZN n) k n s
AKSclean.aks_thm_1
poly_intro_range (ZN p) k n s
AKSclean.aks_thm_2
AKSclean.aks_thm
1
2
3
Ch ??: AKS Main Theorem
Introspective Relation: n intro (X + c)
NS
PS
MS
[polyh/h]Q h
Proof
PHP
Ch ??: AKS Complexity
power_freeM n
paramM n
poly_intro_rangeM n k s
aksM n: analysis
Ch ??: Conclusion
$AKSclean.aksM_value_alt$
$countAKS.aksM_steps_big_O$