

# Pluralities and Sets

Roger Bishop Jones

## Abstract

This is a formal exploration with ProofPower of a topic addressed by Øystein Linnebo in a paper under the same title.

Created 2011/02/24

Last Change Date: 2011/05/16 21:40:17

Id: t046.doc,v 1.8 2011/05/16 21:40:17 rbj Exp

<http://www.rbjones.com/rbjpub/pp/doc/t046.pdf>

© Roger Bishop Jones; Licenced under Gnu LGPL

# Contents

<b>1</b>	<b>Prelude</b>	<b>2</b>
<b>2</b>	<b>Changes</b>	<b>3</b>
2.1	Recent Changes . . . . .	3
2.2	Changes Under Consideration . . . . .	3
2.3	Issues . . . . .	3
<b>3</b>	<b>Introduction</b>	<b>4</b>
<b>4</b>	<b>Methods</b>	<b>4</b>
4.1	Dependencies . . . . .	4
<b>5</b>	<b>Plural Quantification</b>	<b>5</b>
5.1	A Plurals Type-Constructor . . . . .	5
<b>6</b>	<b>Modalised Set Theory</b>	<b>7</b>
6.1	Alternatives to COLLAPSE . . . . .	7
6.2	Restoring Consistency . . . . .	7
<b>7</b>	<b>Logical Properties of the Modalised Quantifiers</b>	<b>13</b>
<b>8</b>	<b>Postscript</b>	<b>15</b>
<b>A</b>	<b>Theory Listings</b>	<b>17</b>
A.1	The Theory t046a . . . . .	17
A.2	The Theory t046b . . . . .	18
	<b>Bibliography</b>	<b>21</b>
	<b>Index</b>	<b>22</b>

## 1 Prelude

This is one of a series of documents in which philosophically motivated technical issues are explored making use of an interactive proof tool for a higher order logic.

The subject matter in this case is a paper by Øystein Linnebo entitled *Pluralities and Sets*[4].

Discussion of what might become of this document in the future may be found the postscript (Section 8).

In this document, phrases in coloured text are hyperlinks, like on a web page, which will usually get you to another part of this document (the blue parts, the contents list, page numbers in the Index) but sometimes take you (the red bits) somewhere altogether different (if you happen to be online), e.g.: [the online copy of this document](#).

## 2 Changes

### 2.1 Recent Changes

**Version 1.8** Principally a reversion to 1.6.

**Version 1.7** This was a fairly radical extension to the scope of the document which has now been reversed.

**Version 1.6** Cosmetic changes, corrections to text, rationalisation of use of subscripts.

**Version 1.5** Two changes have been made to the specifications, both of which represent corrections to my reading of Linnebo's intentions, and proofs reworked as necessary.

The first is that the modal version of plural quantification has been made sensitive to the stage, so that plural quantifiers range only over plurals of sets individuated at that stage (previously they ranged over plurals of sets formed at any stage).

The second is to the requirement on stages, which was previously stronger than Linnebo's in asserting closure under union<sup>1</sup>. This is now weakened to directedness.

I have also, *en passant*, tried to ensure that all plural quantifications bind a double letter variable.

### 2.2 Changes Under Consideration

It seems to me that some further requirements should be placed on the stages, but I see nothing in Linnebo's paper to underwrite them.

Firstly that the stages should exhaust the sets. Secondly that "ThisStage" (the default stage in which judgements are to be understood) should be the first stage, and should be empty.

There is nothing to say that there are many stages, nor is there any analysis of the strength of the resulting set theory. One would presumably want to be able to say that it is possible that we have a model of ZF, or of some large cardinal axiom, and this would require that there are very many stages, which the theory does not at present require.

### 2.3 Issues

Immediately following from the above considerations about stages is the question whether of necessity there should be any sets at all (prior to assuming COLLAPSE or some other ontological principle).

Linnebo talks of plurals as being necessarily non-empty, but does not reflect this in his plural comprehension scheme. I have therefore allowed them to be empty.

If I insisted on their being non-empty, then a contradiction could be derived from plural comprehension alone (without COLLAPSE). If on the other hand, we insist on them being non-empty and restrict plural comprehension accordingly, then it looks as if we are not then able to prove that there are any sets (unless this is made a part of the conception of stages), and cannot even derive a contradiction from the combination of plural comprehension and COLLAPSE.

See also Section 8.

---

<sup>1</sup>Which was not so much a misreading of Linnebo, as a careless mistranlation.

### 3 Introduction

Three logical systems are explored in combination in this document. They are:

- The logic of pluralities.
- Set theory.
- Modal operators.

At this stage the document is little more than a partial formalisation materials from Linnebo's paper.

### 4 Methods

The following formal analysis is conducted in a single logical system, a Higher Order Logic based on Church's formulation of the Simple Theory of Types[1]. The formal treatment is prepared with the assistance of an interactive theorem proving tool. This tool assists by syntax checking and type-checking specifications, by confirming that these specifications are conservative over the initial logical system, by facilitating the construction of detailed formal proofs and mechanically checking their correctness. The tool also prepares listings of the resulting theories, which may be found in appendices A.1 and A.2 and facilitates the preparation of documents including the formal materials.

There is some additional complexity in undertaking strictly formal work in this manner, which is not entirely eliminated by the use of software support. Feasibility depends on careful choice of methods (and problems) to keep complexity within bounds. In the kind of exploratory investigation at hand, one simplification is to avoid reasoning about syntax. This may be done by constructing interpretations of the target systems and reasoning about these interpretations in HOL. In the resulting theorems the syntax of HOL (which is in some degree extendable) is used to express claims which correspond to the rules, axioms and theorems of the logical system or systems under investigation. When proven they give good grounds for belief in the soundness of the logic under consideration, even though we have avoided formal treatment of its syntax.

This approach to reasoning in some logic of interest using a tool supporting a different logical system, is sometimes called *shallow embedding* (by contrast with *deep embedding* in which both syntax and semantics and the relationship between them are formally treated, supporting full formalisation of the metatheory). An extended discussion of these methods is not within our present scope, but I will try to include a certain amount of further explanation as the document proceeds in the hope of making the technical detail as intelligible as practicable.

#### 4.1 Dependencies

For a complete understanding of the details of the formal materials in this document it would strictly be necessary to refer to the definition of the language in which the specifications are written (ProofPower-HOL) and to the listings of the various theories in the context of which these theories have been developed. I hope that the material will be intelligible to a reasonable degree without studying all this material, many readers will already be familiar with Church's formulation of the Simple Theory of Types [1] and will be familiar with the meanings of the usual logical connectives in that context.

For the full detail the following documents may be consulted.

1. Church's formulation of STT: [1].
2. The ProofPower-HOL language: informal description [6]
3. The ProofPower-HOL language: formal specification [5]
4. ProofPower-HOL theory listings: [7] or [in HTML at RBJones.com](#)<sup>2</sup>.
5. Other theories at RBJones.com: `rbjmisc` [3], `t045` [2].
6. Complete documentation for ProofPower can be obtained from [the ProofPower web pages](#)<sup>3</sup>.

## 5 Plural Quantification

Since quantification in HOL is *typed*, and the quantifiers therefore range over the elements of any chosen type, to interpret plural quantification we need a type of plurals, or better, a *type-constructor*, which, given any type, will construct the type of plurals of values of its argument type.

We therefore introduce in HOL just such a type constructor, and the polymorphic relationship '*is one of*' ( $<$ ) between arbitrary values and plurals.

SML

```
|open_theory "rbjmisc";  
|force_new_theory "t046a";
```

### 5.1 A Plurals Type-Constructor

In order to be able to quantify over plurals, we introduce a new type constructor.

This can be done axiomatically, or conservatively (using definitions rather than arbitrary axioms). A conservative extension is normally preferred, but this requires a full understanding of the requirements since once the type has been defined no further modifications can be made. In this case we adopt a kind of compromise.

We explore the logic of plurals without settling the question of which non-empty extensions of any particular type do form plurals. This will be achieved by introducing a loosely defined constant to determine this question, and then proving hypothetical results in the theory of plurals dependent on hypotheses about the properties of this constant, i.e. dependent upon various assumptions about which extensions are plurals.

I introduce a 1-ary type constructor for plurals. It is not clear to me from Linnebo's paper whether this is the correct way to model his intentions. Sometimes Linnebo talks as if plurals can be formed from absolutely everything, but if plurals exist, if we can quantify over plurals, then absolutely everything would include the plurals, and I see no sign in Linnebo that he intends this. Instead we have a new type  $S$  which may include both urelements and sets, and a type-constructor ' $a$  PL' which is the type of plurals of values of type ' $a$ '. It is not impossible that the plurals are also elements of  $S$ , but the type system prevents us from asking the question.

---

<sup>2</sup><http://rbjones.com/rbjpub/pp/pptheories.html>

<sup>3</sup><http://www.lemma-one.com/ProofPower/doc/doc.html>

We do have quite a variety of kinds of set already available, but here we must start from scratch again. It is clear that Linnebo is thinking in terms of a well-founded set theory with urelements.

The type constructors are introduced thus:

```
SML
| new_type("PL", 1);
| new_type("S", 0);
```

We then introduce some primitive constants for these types, without giving definitions.

```
SML
| new_const("<_p", "⌈!a → ('a) PL → BOOL⌋");
| declare_infix(240, "<_p");
| declare_alias("<", "⌈$_<_p⌋");
| new_const("∈_s", "⌈: S → S → BOOL⌋");
| declare_infix(230, "∈_s");
| declare_alias("∈", "⌈$_∈_s⌋");
```

We are now in a position to define various notions which Linnebo uses.

```
HOL Constant
| FORM: S PL × S → BOOL
|-----
|  $\forall xx\ y\bullet\ \mathit{FORM}(xx, y) \Leftrightarrow \forall u\bullet\ u\ <_p\ xx \Leftrightarrow u\ \in_s\ y$ 
```

```
HOL Constant
| COLLAPSE: BOOL
|-----
|  $\mathit{COLLAPSE} \Leftrightarrow \forall xx:S\ PL\bullet\ \exists y:S\bullet\ \mathit{FORM}(xx, y)$ 
```

The principle of plural comprehension is that the extension of any formula is a plurality. Linnebo does elsewhere require that pluralities be non-empty, but he does not incorporate this requirement into his principle of plural comprehension. The following definition differs from Linnebo's, as our proceeding does generally, in being formulated in HOL rather than in some first order language. So here we have a single principle which quantifies over properties, where Linnebo has a first order axiom scheme.

Note also that here *P\_Comp* is not an axiom. It is the name of a boolean constant, which we will use as a hypothesis (on the left of sequents) in some of the theorems which we prove later.

```
HOL Constant
| P_Comp: BOOL
|-----
|  $P\_Comp \Leftrightarrow \forall \phi\bullet\ \exists xx:S\ PL\bullet\ \forall u:S\bullet\ u\ <_p\ xx \Leftrightarrow \phi\ u$ 
```

Linnebo then derives a contradiction, corresponding to which we have the theorem:

```
| COLLAPSE_contradiction_thm =
| COLLAPSE, P_Comp ⊢ F
```

Proofs will not be shown, they are all straightforward.

Linnebo observes that this is normally taken as a refutation of COLLAPSE, but proposes to resolve the contradiction by some other means.

To do this he needs to scrutinise carefully all the assumptions which participate in the derivation of the contradiction. Doing the proof formally gives us a strong basis for considering this matter, for it is clearer what the logical context of the proof is.

This has involved some assumptions about the types of pluralities and sets which are not explicit in Linnebo (who is not working in a context in which these questions arise). We have also formulated a second order principle of plural abstraction, whereas Linnebo's is an axiom scheme.

Apart from considering *P\_Comp* as a suspect, Linnebo considers that he has assumed that one can quantify over absolutely everything, and questions this assumption. This is implicit in our formalisation if we interpret *S* as *absolutely everything*, though it is not apparent that the proof depends on this.

Linnebo approaches a distinction between COLLAPSE and full comprehension by referring to the concept used in comprehension as *intensional*, by contrast with the extensional nature of the plurality used to determine a set in COLLAPSE. This doesn't work very well here, because COLLAPSE is formulated using an extensional notion of concept (a propositional function in an extensional higher order logic), and still suffices to derive the contradiction. The point Linnebo is driving at is that comprehension does not give a properly iterative process of building the universe of sets, because a set introduced at some point in an imagined iterative process has one extension before we add it, but once added it (and other new sets) become candidates for membership in that same set, so the extension may therefore change. He seems here to be wanting to treat the pluralities as rigid designators of extensions, even if they are defined by plural comprehension, so that the question of whether a set introduced by COLLAPSE is a member of itself does not arise. However, this perspective is not a legitimate interpretation of the theory as we have formulated it here. If we want our universe to be the result of some iterative process, we need a new approach to the formal modelling.

## 6 Modalised Set Theory

### 6.1 Alternatives to COLLAPSE

Linnebo devotes a section of his paper to a critique of alternatives to COLLAPSE. I pass over this to concentrate on his positive proposals.

### 6.2 Restoring Consistency

SML

```
|open_theory "t046a";  
|force_new_theory "t046b";  
|new_parent "t045";
```

We now come to Linnebo's proposals for obtaining a consistent system incorporating COLLAPSE.

This is based upon an idea by Yablo that determinacy of the pool of candidates is a pre-requisite for the determinacy of a set determined by some condition  $\phi$  (this is good in the present context

because it replaces the idea that the problem is with *intensionality* of the condition, which doesn't work well in our present context).

This translates into the requirement that the condition determines a set only if it is applied at a specific stage in an iterative process of forming the universe of sets, and the resulting set appears at the next stage. The extension of the set will then correspond to those elements prior to the determination of the set which satisfy the condition. Not every element of the complete universe (not *absolutely everything*) is a candidate, and in particular the set itself is not a candidate for membership. We may say parenthetically, that what Linnebo has done here is something like adopting separation instead of comprehension. Note however, that Linnebo does not embrace the idea that the hierarchy of sets ever can be completed (and nor do I). In the present context, interpretations of our theory will assign some set to the type  $S$ , which will (when we consider further constraints below) correspond to some stage in the formation at which this interminable process is arbitrarily brought to a halt. The results we obtain relate not to the completion, nor to any particular stage short of completion, but to some unspecified stage short of completion, and hence to all unspecified stages (subject to some constraints) short of completion.

Linnebo proposes to spell out the idea using modal vocabulary to talk about stages. This is rather different to and more subtle than the initial result of our formalisation as relating to arbitrary stages, and so I now examine how such a modal notation can be used.

In this modal notion, a *possible world* is a stage in the formation of the “universe”  $S$ . Let us ignore the fact that  $S$  itself cannot be the whole, for the present, and imagine that it is. Such a stage will be a subset of the type  $S$ , and therefore will in our logical system be a value of type  $S\ SET$ . The collection of possible worlds will then be a set of such sets. We will call them *Stages*.

We introduce this as a loosely defined constant. The only constraint we apply to it is to ensure that this is a non-empty directed set (though it may be necessary to beef this up later).

HOL Constant

**Stages:** ( $S\ SET$ )  $SET$

$(\exists x \bullet x \in Stages)$

$\wedge \forall x\ y \bullet x \in Stages \wedge y \in Stages \Rightarrow \exists z \bullet z \in Stages \wedge x \subseteq z \wedge y \subseteq z$

We note here the elementary consequence of directedness that every member of a stage is reachable from any stage.

**Stages\_fc\_thm** =

$\vdash \forall x\ y \bullet x \in Stages \wedge y \in Stages \Rightarrow \exists z \bullet z \in Stages \wedge x \subseteq z \wedge y \subseteq z$

**set\_reachability\_thm** =

$\vdash \forall s1\ s2\ x \bullet s1 \in Stages \wedge s2 \in Stages \wedge x \in s2$   
 $\Rightarrow (\exists s3 \bullet s3 \in Stages \wedge x \in s3 \wedge s1 \subseteq s3)$

We could then define the relevant notion of accessibility between these stages, but since this is set inclusion (or its converse) which is already defined we will use that directly (A is accessible from B if A is a stage subsequent to B and hence  $\ulcorner B \subseteq A \urcorner$ ).

The expression of modal claims in our logic requires a recasting of all the logical vocabulary. The definitions extant in HOL treat propositions as values of type  $\ulcorner :\ BOOL \urcorner$ , but in the modal logic they will have to be considered as values of type  $\ulcorner :\ S\ SET \rightarrow\ BOOL \urcorner$ , i.e. a Boolean valued function over the relevant possibilities, which in this case are stages in the iterative cumulation of sets.



From here on in the names used for constants will usually be subscripted with ‘m’ to distinguish them from previous versions, but will be aliased back to the non-subscripted name so that the subscript may be omitted if no ambiguity arises (usually the type-checker can resolve any ambiguity). In the case of the universal and existential quantifiers we need different definitions according to whether we are quantifying over sets or plurals, and in these cases the subscripts ‘s’ and ‘p’ will be used as appropriate.

SML

```
| declare_type_abbrev("PROPm", [], ⌈:S SET → BOOL⌋);
```

The modal operators are defined as follows:

HOL Constant

$\Box_m : PROP_m \rightarrow PROP_m$	
$\forall s \bullet \Box_m s = \lambda w \bullet \forall v \bullet v \in Stages \wedge w \subseteq v \Rightarrow s v$	

HOL Constant

$\Diamond_m : PROP_m \rightarrow PROP_m$	
$\forall s \bullet \Diamond_m s = \neg \Box_m (\neg s)$	

So that we need not use the subscript  $m$  unless ambiguity might otherwise arise, we introduce the un-subscripted modal operators as aliases:

SML

```
| declare_alias("□", ⌈□m⌋);
| declare_alias("◇", ⌈◇m⌋);
```

Since modal propositions are not of type  $\lceil : BOOL \rceil$  we need a way of asserting them, for which we use the symbol  $\models$ , which should be read as asserting ‘truth’. Though ‘truth’ simpliciter in a modal logic should perhaps be construed as truth in “this world”, and since worlds are stages we use the name *ThisStage* and introduce this as a new constant. Linnebo says nothing about this, so nor do we, so we know nothing from this loose ‘definition’ about *ThisStage* other than that it is a *Stage* (and we do already know that there is one, so the definition is conservative).

HOL Constant

$\$ThisStage : S SET$	
$ThisStage \in Stages$	

HOL Constant

$\$ \models_m : PROP_m \rightarrow BOOL$	
$\forall p \bullet \models_m p \Leftrightarrow p ThisStage$	

SML

```
| declare_alias("⊨", ⌈⊨m⌋);
| declare_prefix(5, "⊨m");
```

$$\begin{array}{l} \diamond \Box_m \text{-thm} = \vdash \forall s \bullet p \bullet (\Box_m p s \Leftrightarrow (\forall s2 \bullet s \subseteq s2 \Rightarrow p s2)) \\ \wedge (\diamond_m p s \Leftrightarrow (\exists s2 \bullet s \subseteq s2 \wedge p s2)) \end{array}$$

We can now *prove* as theorems the propositions which would normally be taken as axioms for this kind of modal logic.

$$\begin{array}{l} \text{distrib}_m \text{-thm} = \vdash \models \Box (A \Rightarrow B) \Rightarrow \Box A \Rightarrow \Box B \\ D_m \text{-thm} = \vdash \models \Box A \Rightarrow \diamond A \\ M_m \text{-thm} = \vdash \models \Box A \Rightarrow A \\ A4_m \text{-thm} = \vdash \models \Box A \Rightarrow \Box (\Box A) \\ \Box M_m \text{-thm} = \vdash \models \Box (\Box A \Rightarrow A) \\ C4_m \text{-thm} = \vdash \models \Box (\Box A) \Rightarrow \Box A \\ C_m \text{-thm} = \vdash \models \diamond (\Box A) \Rightarrow \Box (\diamond A) \end{array}$$

The quantifiers, or at least those over sets, are now expected to be modally sensitive. They quantify over a particular stage in the formation of the cumulative hierarchy.

HOL Constant

$$\begin{array}{l} \forall_s : (S \rightarrow PROP_m) \rightarrow PROP_m \\ \hline \forall f \bullet \forall_s f = \lambda s \bullet \forall x \bullet x \in s \Rightarrow f x s \end{array}$$

SML

```
declare_alias ("∀", "∀s");
declare_binder "∀s";
```

HOL Constant

$$\begin{array}{l} \exists_s : (S \rightarrow PROP_m) \rightarrow PROP_m \\ \hline \forall f : S \rightarrow PROP_m \bullet \exists_s f = \lambda s : S \bullet SET \bullet \exists x : S \bullet x \in s \wedge f x s \end{array}$$

SML

```
declare_alias ("∃", "∃s");
declare_binder "∃s";
```

Declaring these as binders has the effect that when applied to a lambda-expression the lambda is to be omitted, giving the usual syntax for quantifiers. Sometimes, as in the following theorems, we want to use the quantifier without binding a variable, in which case its status as a binder can be suspended for a particular occurrence by prefixing the quantifier with a dollar sign.

$$\begin{array}{l} \exists_s \text{-thm} = \vdash \forall f s \bullet \$\exists f s \Leftrightarrow (\exists x \bullet x \in s \wedge f x s) \\ \forall_s \text{-thm} = \vdash \forall f s \bullet \$\forall f s \Leftrightarrow (\forall x \bullet x \in s \Rightarrow f x s) \end{array}$$

We need new quantifiers to quantify over pluralities in a modal context. I have made plural quantification also sensitive to the stage, but in this case the constraint is not membership of the stage but extensional inclusion in it.

HOL Constant

$$\frac{\forall_p:(S \text{ PL} \rightarrow \text{PROP}_m) \rightarrow \text{PROP}_m}{\forall f \bullet \forall_p f = \lambda s \bullet \forall xx \bullet (\forall z \bullet z <_p xx \Rightarrow z \in s) \Rightarrow f \text{ } xx \text{ } s}$$

SML

```
declare_alias ("∀", "∀p");  
declare_binder "∀p";
```

HOL Constant

$$\frac{\exists_p:(S \text{ PL} \rightarrow \text{PROP}_m) \rightarrow \text{PROP}_m}{\forall f \bullet \exists_p f = \lambda s \bullet \exists xx:S \text{ PL} \bullet (\forall z \bullet z <_p xx \Rightarrow z \in s) \wedge f \text{ } xx \text{ } s}$$

SML

```
declare_alias ("∃", "∃p");  
declare_binder "∃p";
```

$$\begin{aligned} \exists_p\text{-thm} &= \vdash \forall f \text{ } s \bullet \$\exists_p f \text{ } s \Leftrightarrow (\exists xx \bullet (\forall z \bullet z <_p xx \Rightarrow z \in s) \wedge f \text{ } xx \text{ } s) \\ \forall_p\text{-thm} &= \vdash \forall f \text{ } s \bullet \$\forall_p f \text{ } s \Leftrightarrow (\forall xx \bullet (\forall z \bullet z <_p xx \Rightarrow z \in s) \Rightarrow f \text{ } xx \text{ } s) \end{aligned}$$

We are now able to demonstrate the ‘converse Barcan’ formula. The Barcan formula itself fails, because it infers from possible existence to actual existence which would only be possible if no stage actually added any extra sets.

$$\text{CBF}_m\text{-thm} = \vdash \models_m (\exists x \bullet \diamond A x) \Rightarrow \diamond (\exists_s x \bullet A x)$$

The notion of plurality has not really changed as we moved to a modal set theory, but our notion of proposition has changed and so we need to replicate the definitions of the plural operations to deliver the right kinds of proposition.

We didn’t actually have a ‘definition’ of plural membership, just a constant about which we made assumptions as needed. Here we lift that constant to deliver a modal proposition which does the same thing at every stage. This reflects the rigidity of the values we quantify over, both sets and plurals.

We need a similar constant at a different type.

SML

```
declare_infix(240, "<m");
```

HOL Constant

$$\frac{\$<_m: 'a \rightarrow ('a) \text{ PL} \rightarrow \text{PROP}_m}{\forall x \text{ } yy \text{ } s \bullet (x <_m yy) \text{ } s \Leftrightarrow x <_p yy}$$

SML

```
declare_alias("<", "⋄$<");
```

On my first pass I defined the modal set membership taking into account of the stage parameter so that it is false if the value of the variable does not exist. Later, seeing Linnebo's appendix in which he affirms the necessity of true membership claims this was thrown into doubt. It turns out that the definition of membership in which membership is false if the set does not appear in the current stage is compatible with the necessity of true membership statements but not with the necessity of false ones, which Linnebo requires, so I have revised the definition accordingly.

SML

```
| declare_infix(240, "∈m");
```

HOL Constant

```
| $∈m: S → S → PROPm
```

---

```
| ∀x y s• (x ∈m y) s ⇔ x ∈s y
```

SML

```
| declare_alias("∈", "⌈$∈m⌋");
```

We now restate the two contradictory principles using modal language (i.e. using quantifiers suitable for use in modal contexts, but not modal operators).

HOL Constant

```
| FORMm: S PL × S → PROPm
```

---

```
| ∀xx y• FORMm(xx, y) = ∀ u• u <m xx ⇔ u ∈m y
```

```
| FORMm-thm = ⊢ ∀ xx y s• FORMm (xx, y) s ⇔ (∀ x• x ∈ s ⇒ ((x < xx) s ⇔ (x ∈ y) s))
```

HOL Constant

```
| COLLAPSEm: BOOL
```

---

```
| COLLAPSEm ⇔ ⊨m ∀p xx:S PL• ∃ y:S• FORMm(xx, y)
```

We can now state the principle of plural comprehension:

HOL Constant

```
| P_Compm: BOOL
```

---

```
| P_Compm ⇔ ∀φ• ⊨m ∃p xx:S PL• ∀ u:S• u <m xx ⇔ φ u
```

Since there are no modal operators, these principles speak only of the stage which is in context globally, of which we know nothing, except what the principles tell is (if they are assumed).

These two principles, under this new interpretation, remain contradictory:

```
| COLLAPSE_contradiction_thm2 =
  COLLAPSEm, P_Compm ⊢ F
```

It is accidental in the previous version of this that the domain of sets is a type and hence cannot be empty. In this version the domain of sets is a class (or may best be thought of as a class, it has type  $\ulcorner :S \text{ SET} \urcorner$  in which  $S$  is the type of sets for present purposes, and  $\text{SET}$  is similar to the higher type in second order logic). We can therefore establish its non-emptiness here only by using the two principles at hand, and it is fortuitous that the principle of plural comprehension does not insist that plurals are non-empty. If it did the two principles as formalised above would not be contradictory.

We now define the two properly modalised versions of plural comprehension and collapse (as given by Linnebo).

HOL Constant

$$\begin{array}{|l} \mathbf{COLLAPSE}^\diamond : \text{BOOL} \\ \hline \mathbf{COLLAPSE}^\diamond \Leftrightarrow \models_m \Box \forall_p xx \bullet \Diamond \exists_s y \bullet \text{FORM}_m(xx, y) \end{array}$$

HOL Constant

$$\begin{array}{|l} \mathbf{P\_Comp}^\diamond : \text{BOOL} \\ \hline \mathbf{P\_Comp}^\diamond \Leftrightarrow \forall \phi \bullet \models_m \Diamond \exists xx \bullet \Box \forall_s u \bullet u <_m xx \Leftrightarrow \phi u \end{array}$$

Once again the, a contradiction is derivable from these two principles.

$$\mathbf{COLLAPSE\_contradiction\_thm3} = \mathbf{COLLAPSE}^\diamond, \mathbf{P\_Comp}^\diamond \vdash F$$

Strangely, though taking this contradiction as establishing the need to weaken plural comprehension, he does not offer a weakened version which is compatible with  $\mathbf{COLLAPSE}$ .

One obvious ‘solution’ is simply to restrict plural comprehension to some stage in the formation of sets, thus in effect downgrading it to separation.

HOL Constant

$$\begin{array}{|l} \mathbf{P\_Comp}_w^\diamond : \text{BOOL} \\ \hline \mathbf{P\_Comp}_w^\diamond \Leftrightarrow \forall \phi \bullet \models_m \Diamond \exists xx \bullet \forall_s u \bullet u <_m xx \Leftrightarrow \phi u \end{array}$$

I don’t have the means in the present context to establish the consistency of this weakened comprehension with  $\mathbf{COLLAPSE}$  though that does seem intuitively plausible.

## 7 Logical Properties of the Modalised Quantifiers

This section addresses the material under the above heading in Linnebo’s Appendix. Some of this material has already emerged as a result of our method of introducing the modal vocabulary, but the main theses of this appendix are metatheoretic and explicitly syntactic in character. Since we do not have here a formalisation of the syntax of the system, these theses cannot be formalised. We could formalise the syntax in order to be able to state them, but the cost of doing so would probably not justify the return. Instead we prove a variety of purely semantic claims which are closely related to the results given by Linnebo.

I believe that Linnebo’s prohibition of instantiation of plural quantifiers with constants is unnecessary in our treatment. He seems by this means to wish to prohibit instantiation of plural quantifiers with

concepts such as *ordinal*, since it is not necessarily the case that the entire extension of a concept will correspond to a plural, once P-Comp has been weakened as necessary for it to be compatible with COLLAPSE.

In our treatment, the HOL type system supplies the necessary constraint. We could instantiate with plural constants, if we had any (and we could have some). But we cannot instantiate with a concept, because concepts do not have the same *type* as plurals.

The next matter considered by Linnebo is the correspondence of his system with the modal logic S4.2, which we have already shown and arises from the definition we have give for the *Stages*, which are our “possible worlds”.

Then we come to the necessity of equality and membership assertions and their negations, three of which Linnebo asserts as axioms.

First we must define equality, inequality and non-membership suitable for use in these modal contexts, i.e. which deliver a modal proposition rather than a BOOL value.

SML

```

| declare_infix(240, "=m");
| declare_infix(240, "≠m");
| declare_infix(240, "∉m");

```

HOL Constant

```

| $=m: S → S → PROPm
|-----
| ∀x y s • (x =m y) s ⇔ x = y

```

HOL Constant

```

| $≠m: S → S → PROPm
|-----
| ∀x y • x ≠m y = ¬ x =m y

```

HOL Constant

```

| $∉m: S → S → PROPm
|-----
| ∀x y • x ∉m y = ¬ x ∈m y

```

The four required results (of which the last three are asserted by Linnebo as axioms) may then be proven, resulting in the following theorems:

```

| eq∈m-□-thm = ⊢ ⊨ x ∈m y ⇒ □ (x ∈m y)
| ≠∈m-□-thm = ⊢ ⊨ x ≠m y ⇒ □ (x ≠m y)
| ∈m-□-thm = ⊢ ⊨ x ∈m y ⇒ □ (x ∈m y)
| ∉m-□-thm = ⊢ ⊨ x ∉m y ⇒ □ (x ∉m y)

```

Now we consider Linnebo’s *Lemma1*. This is needed by Linnebo to prove his *Theorem 1*, instead of which we will attempt to demonstrate a corresponding purely semantic claim, so we will not need *Lemma 1* as it is given by Linnebo. However, by way of checking that our definitions are consistent with Linnebo’s system the following results, corresponding to steps in Linnebo’s proof, have been proven.

$$\begin{aligned}
& \diamond \Box \mathit{eq}_m \mathit{thm} = \\
& \quad \vdash \models \diamond (x =_m y) \Rightarrow \Box (x =_m y) \\
& \diamond \Box \mathit{\in}_m \mathit{thm} = \\
& \quad \vdash \models \diamond (x \in y) \Rightarrow \Box (x \in y) \\
& \diamond \Box \neg \mathit{thm} = \models \diamond \phi \Rightarrow \Box \phi \\
& \quad \vdash \models \diamond (\neg \phi) \Rightarrow \Box (\neg \phi) \\
& \diamond \Box \wedge \mathit{thm} = \models \diamond \phi \Rightarrow \Box \phi, \models \diamond \psi \Rightarrow \Box \psi \\
& \quad \vdash \models \diamond (\phi \wedge \psi) \Rightarrow \Box (\phi \wedge \psi) \\
& \diamond \Box \diamond \exists_p \mathit{thm} = \forall xx \bullet \models \diamond (\psi xx) \Rightarrow \Box (\psi xx) \\
& \quad \vdash \models \diamond (\exists xx \bullet \psi xx) \Rightarrow \Box (\diamond (\exists xx \bullet \psi xx))
\end{aligned}$$

Linnebo's *Theorem 1* is to the effect that the modalised quantifiers are equivalent to the normal non modal quantification over sets. Since this means that they quantify over all the sets in all the stages this can be stated directly rather than as a result about provability of translations.

The following theorem comes close to that result:

$$\begin{aligned}
& \diamond \exists_p \mathit{lemma1} = \vdash \forall \phi \bullet (\models \diamond (\exists x \bullet \phi x)) \\
& \quad \Leftrightarrow (\exists s \bullet s \in \mathit{Stages} \wedge \mathit{ThisStage} \subseteq s \wedge x \in s \wedge \phi x s)
\end{aligned}$$

It falls short by requiring that  $\phi x$  be true at some stage which is reachable from *ThisStage*. This would not be an issue if we required that all stages are reachable from *ThisStage*, which is what one might reasonably expect.

This theorem is of greater scope than Linnebo's *Theorem 1*, since we have no constraint on what  $\phi$  may be, whereas *Theorem 1* is concerned only with the modal translations of formulae of non-modal first order set theory. Since for that set of propositions *Lemma 1* shows that they are necessary if true, and hence there will always be a stage which is reachable from *ThisStage* at which  $\exists x \bullet \phi x$  there is any stage at which it holds.

So our formally obtained results come very close to Linnebo's *Theorem 1* if not quite as close as we might like. To get a more exact semantic correspondent to Linnebo's *Theorem 1* it would be necessary to define the set of modal propositions which are the meanings of the set of formulae obtainable by modalisation of sentences of first order set theory.

## 8 Postscript

I have now worked through the main body of Linnebo's paper, and the second of his two appendices.

The first appendix is different in character, obtaining a result stated in terms of plural set theory for which an equivalent theorem is uncontroversial in ZFC, I shall not attempt to formalise this proof.

The second appendix gives more details of and results in the modal/plural set theory. Much of this is metatheoretic. It would be possible to treat this more fully within the bounds of the syntax-free methods I have adopted, but this would be significantly more complex and time consuming than any of the other material so far undertaken, for relatively little further benefit.

Beyond these technical details there is then the question of the philosophical significance both of Linnebo's paper and of this kind of formal treatment.

There are then some problems in relation to the semantics of set theory not addressed by Linnebo which might benefit from the modal approach, and I may possibly add further material on these matters.



## A Theory Listings

### A.1 The Theory t046a

#### Parents

*rbjmisc*

#### Children

*t046b*

#### Constants

$\$<_p$              $'a \rightarrow 'a \text{ PL} \rightarrow \text{BOOL}$   
 $\$\epsilon_s$              $S \rightarrow S \rightarrow \text{BOOL}$   
**FORM**             $S \text{ PL} \times S \rightarrow \text{BOOL}$   
**COLLAPSE**         $\text{BOOL}$   
**P\_Comp**            $\text{BOOL}$

#### Aliases

$<$                  $\$<_p : 'a \rightarrow 'a \text{ PL} \rightarrow \text{BOOL}$   
 $\epsilon$                  $\$\epsilon_s : S \rightarrow S \rightarrow \text{BOOL}$

#### Types

$'1 \text{ PL}$   
 $S$

#### Fixity

*Right Infix 230:*

$\epsilon_s$

*Right Infix 240:*

$<_p$

#### Definitions

**FORM**             $\vdash \forall xx \ y \bullet \text{FORM} (xx, y) \Leftrightarrow (\forall u \bullet u < xx \Leftrightarrow u \in y)$   
**COLLAPSE**         $\vdash \text{COLLAPSE} \Leftrightarrow (\forall xx \bullet \exists y \bullet \text{FORM} (xx, y))$   
**P\_Comp**            $\vdash \text{P\_Comp} \Leftrightarrow (\forall \phi \bullet \exists xx \bullet \forall u \bullet u < xx \Leftrightarrow \phi u)$

#### Theorems

**COLLAPSE\_contradiction\_thm**

$\text{COLLAPSE}, \text{P\_Comp} \vdash F$

## A.2 The Theory t046b

### Parents

*t045 t046a*

### Constants

<b>Stages</b>	$S \mathbb{P} \mathbb{P}$
$\square_m$	$PROP_m \rightarrow PROP_m$
$\diamond_m$	$PROP_m \rightarrow PROP_m$
<b>ThisStage</b>	$S \mathbb{P}$
$\$ \models_m$	$PROP_m \rightarrow BOOL$
$\$ \forall_s$	$(S \rightarrow PROP_m) \rightarrow PROP_m$
$\$ \exists_s$	$(S \rightarrow PROP_m) \rightarrow PROP_m$
$\$ \forall_p$	$(S \text{ PL} \rightarrow PROP_m) \rightarrow PROP_m$
$\$ \exists_p$	$(S \text{ PL} \rightarrow PROP_m) \rightarrow PROP_m$
$\$ <_m$	$'a \rightarrow 'a \text{ PL} \rightarrow PROP_m$
$\$ \in_m$	$S \rightarrow S \rightarrow PROP_m$
<b>FORM<sub>m</sub></b>	$S \text{ PL} \times S \rightarrow PROP_m$
<b>COLLAPSE<sub>m</sub></b>	$BOOL$
<b>P_Comp<sub>m</sub></b>	$BOOL$
<b>COLLAPSE<sup>◇</sup></b>	$BOOL$
<b>P_Comp<sup>◇</sup></b>	$BOOL$
<b>P_Comp<sub>w</sub><sup>◇</sup></b>	$BOOL$
$\$ =_m$	$S \rightarrow S \rightarrow PROP_m$
$\$ \neq_m$	$S \rightarrow S \rightarrow PROP_m$
$\$ \notin_m$	$S \rightarrow S \rightarrow PROP_m$

### Aliases

$\square$	$\square_m : PROP_m \rightarrow PROP_m$
$\diamond$	$\diamond_m : PROP_m \rightarrow PROP_m$
$\models$	$\$ \models_m : PROP_m \rightarrow BOOL$
$\forall$	$\$ \forall_s : (S \rightarrow PROP_m) \rightarrow PROP_m$
$\exists$	$\$ \exists_s : (S \rightarrow PROP_m) \rightarrow PROP_m$
$\forall$	$\$ \forall_p : (S \text{ PL} \rightarrow PROP_m) \rightarrow PROP_m$
$\exists$	$\$ \exists_p : (S \text{ PL} \rightarrow PROP_m) \rightarrow PROP_m$
$<$	$\$ <_m : 'a \rightarrow 'a \text{ PL} \rightarrow PROP_m$
$\in$	$\$ \in_m : S \rightarrow S \rightarrow PROP_m$

### Type Abbreviations

**PROP<sub>m</sub>**      *PROP<sub>m</sub>*

### Fixity

<b>Binder:</b>	$\forall_p$ $\forall_s$ $\exists_p$ $\exists_s$
<b>Right Infix 240:</b>	$=_m$ $\in_m$ $\notin_m$ $<_m$ $\neq_m$
<b>Prefix 5:</b>	$\models_m$

## Definitions

<b>Stages</b>	$\vdash (\exists x \bullet x \in \text{Stages})$ $\wedge (\forall x y$ $\bullet x \in \text{Stages} \wedge y \in \text{Stages}$ $\Rightarrow (\exists z \bullet z \in \text{Stages} \wedge x \subseteq z \wedge y \subseteq z))$
$\Box_m$	$\vdash \forall s \bullet \Box s = (\lambda w \bullet \forall v \bullet v \in \text{Stages} \wedge w \subseteq v \Rightarrow s v)$
$\Diamond_m$	$\vdash \forall s \bullet \Diamond s = (\neg \Box (\neg s))$
<b>ThisStage</b>	$\vdash \text{ThisStage} \in \text{Stages}$
$\models_m$	$\vdash \forall p \bullet (\models p) \Leftrightarrow p \text{ ThisStage}$
$\forall_s$	$\vdash \forall f \bullet \$\forall f = (\lambda s \bullet \forall x \bullet x \in s \Rightarrow f x s)$
$\exists_s$	$\vdash \forall f \bullet \$\exists f = (\lambda s \bullet \exists x \bullet x \in s \wedge f x s)$
$\forall_p$	$\vdash \forall f$ $\bullet \$\forall f = (\lambda s \bullet \forall xx \bullet (\forall z \bullet z < xx \Rightarrow z \in s) \Rightarrow f xx s)$
$\exists_p$	$\vdash \forall f$ $\bullet \$\exists f = (\lambda s \bullet \exists xx \bullet (\forall z \bullet z < xx \Rightarrow z \in s) \wedge f xx s)$
$<_m$	$\vdash \forall x yy s \bullet (x < yy) s \Leftrightarrow x < yy$
$\in_m$	$\vdash \forall x y s \bullet (x \in y) s \Leftrightarrow x \in y$
<b>FORM<sub>m</sub></b>	$\vdash \forall xx y \bullet \text{FORM}_m (xx, y) = (\forall u \bullet u < xx \Leftrightarrow u \in y)$
<b>COLLAPSE<sub>m</sub></b>	$\vdash \text{COLLAPSE}_m \Leftrightarrow (\models (\forall xx \bullet \exists y \bullet \text{FORM}_m (xx, y)))$
<b>P_Comp<sub>m</sub></b>	$\vdash P\_Comp_m \Leftrightarrow (\forall \phi \bullet \models (\exists xx \bullet \forall u \bullet u < xx \Leftrightarrow \phi u))$
<b>COLLAPSE<sup>◇</sup></b>	$\vdash \text{COLLAPSE}^\diamond \Leftrightarrow (\models \Box (\forall xx \bullet \Diamond (\exists y \bullet \text{FORM}_m (xx, y))))$
<b>P_Comp<sup>◇</sup></b>	$\vdash P\_Comp^\diamond \Leftrightarrow (\forall \phi \bullet \models \Diamond (\exists xx \bullet \Box (\forall u \bullet u < xx \Leftrightarrow \phi u)))$
<b>P_Comp<sub>w</sub><sup>◇</sup></b>	$\vdash P\_Comp_w^\diamond \Leftrightarrow (\forall \phi \bullet \models \Diamond (\exists xx \bullet \forall u \bullet u < xx \Leftrightarrow \phi u))$
$=_m$	$\vdash \forall x y s \bullet (x =_m y) s \Leftrightarrow x = y$
$\neq_m$	$\vdash \forall x y \bullet x \neq_m y = (\neg x =_m y)$
$\notin_m$	$\vdash \forall x y \bullet x \notin_m y = (\neg x \in y)$

## Theorems

### Stages\_fc\_thm

$$\vdash \forall x y$$

$$\bullet x \in \text{Stages} \wedge y \in \text{Stages}$$

$$\Rightarrow (\exists z \bullet z \in \text{Stages} \wedge x \subseteq z \wedge y \subseteq z)$$

### set\_reachability\_thm

$$\vdash \forall s1 s2 x$$

$$\bullet s1 \in \text{Stages} \wedge s2 \in \text{Stages} \wedge x \in s2$$

$$\Rightarrow (\exists s3 \bullet s3 \in \text{Stages} \wedge x \in s3 \wedge s1 \subseteq s3)$$

### ◇□<sub>m</sub>-thm

$$\vdash \forall s p$$

$$\bullet (\Box p s \Leftrightarrow (\forall s2 \bullet s2 \in \text{Stages} \wedge s \subseteq s2 \Rightarrow p s2))$$

$$\wedge (\Diamond p s \Leftrightarrow (\exists s2 \bullet s2 \in \text{Stages} \wedge s \subseteq s2 \wedge p s2))$$

### distrib<sub>m</sub>-thm

$$\vdash \models \Box (A \Rightarrow B) \Rightarrow \Box A \Rightarrow \Box B$$

### D<sub>m</sub>-thm

$$\vdash \models \Box A \Rightarrow \Diamond A$$

### M<sub>m</sub>-thm

$$\vdash \models \Box A \Rightarrow A$$

### A4<sub>m</sub>-thm

$$\vdash \models \Box A \Rightarrow \Box (\Box A)$$

### □M<sub>m</sub>-thm

$$\vdash \models \Box (\Box A \Rightarrow A)$$

### C4<sub>m</sub>-thm

$$\vdash \models \Box (\Box A) \Rightarrow \Box A$$

### C<sub>m</sub>-thm

$$\vdash \models \Diamond (\Box A) \Rightarrow \Box (\Diamond A)$$

### ∀<sub>s</sub>-thm

$$\vdash \forall f s \bullet \$\forall f s \Leftrightarrow (\forall x \bullet x \in s \Rightarrow f x s)$$

### ∃<sub>s</sub>-thm

$$\vdash \forall f s \bullet \$\exists f s \Leftrightarrow (\exists x \bullet x \in s \wedge f x s)$$

$\forall_p\text{-thm}$   $\vdash \forall f s$   
 $\bullet \ \$\forall f s \Leftrightarrow (\forall xx \bullet (\forall z \bullet z < xx \Rightarrow z \in s) \Rightarrow f xx s)$

$\exists_p\text{-thm}$   $\vdash \forall f s$   
 $\bullet \ \$\exists f s \Leftrightarrow (\exists xx \bullet (\forall z \bullet z < xx \Rightarrow z \in s) \wedge f xx s)$

$CBF_m\text{-thm}$   $\vdash \models (\exists x \bullet \diamond (A x)) \Rightarrow \diamond (\exists x \bullet A x)$

$COLLAPSE\_contradiction\_thm2$   
 $COLLAPSE_m, P\_Comp_m \vdash F$

$COLLAPSE\_contradiction\_thm3$   
 $COLLAPSE^\diamond, P\_Comp^\diamond \vdash F$

$eq_m\text{-}\square\text{-thm}$   $\vdash \models x =_m y \Rightarrow \square (x =_m y)$

$\neq_m\text{-}\square\text{-thm}$   $\vdash \models x \neq_m y \Rightarrow \square (x \neq_m y)$

$\in_m\text{-}\square\text{-thm}$   $\vdash \models x \in y \Rightarrow \square (x \in y)$

$\notin_m\text{-}\square\text{-thm}$   $\vdash \models x \notin_m y \Rightarrow \square (x \notin_m y)$

$\diamond\square eq_m\text{-thm}$

$\diamond\square\notin_m\text{-}\square\text{-thm}$   $\vdash \models \diamond (x =_m y) \Rightarrow \square (x =_m y)$

$\diamond\square\in_m\text{-thm}$   $\vdash \models \diamond (x \in y) \Rightarrow \square (x \in y)$

$\diamond\square\neg\text{-thm}$   $\models \diamond \phi \Rightarrow \square \phi \vdash \models \diamond (\neg \phi) \Rightarrow \square (\neg \phi)$

$\diamond\square\wedge\text{-thm}$   $\models \diamond \phi \Rightarrow \square \phi, \models \diamond \psi \Rightarrow \square \psi \vdash \models \diamond (\phi \wedge \psi) \Rightarrow \square (\phi \wedge \psi)$

$\diamond\square\diamond\exists_p\text{-thm}$   $\forall xx \bullet \models \diamond (\psi xx) \Rightarrow \square (\psi xx)$   
 $\vdash \models \diamond (\exists xx \bullet \psi xx) \Rightarrow \square (\diamond (\exists xx \bullet \psi xx))$

$\diamond\exists_p\text{-lemma1}$   $\vdash \forall \phi$   
 $\bullet (\models \diamond (\exists x \bullet \phi x))$   
 $\Leftrightarrow (\exists s x$   
 $\bullet s \in Stages \wedge ThisStage \subseteq s \wedge x \in s \wedge \phi x s)$

## Bibliography

- [1] Alonzo Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–, 1940.
- [2] Roger Bishop Jones. Meaning, Modality and Metaphysics. *RBJones.com*, 2010.  
<http://www.rbjones.com/rbjpub/pp/doc/t045.pdf>.
- [3] Roger Bishop Jones. Miscellaneous Theory Supplements. *RBJones.com*, 2010.  
<http://www.rbjones.com/rbjpub/pp/doc/t006.pdf>.
- [4] Øystein Linnebo. Pluralities and Sets. *Journal of Philosophy*, 2011. To appear.
- [5] ds/fmu/ied/spc001. *HOL Formalised: Language and Overview*. R.D. Arthan, Lemma 1 Ltd.,  
<http://www.lemma-one.com>.
- [6] ds/fmu/ied/usr005. *ProofPower Description*. Lemma 1 Ltd., <http://www.lemma-one.com>.
- [7] lemma1/hol/usr029. *ProofPower HOL Reference Manual*. Lemma 1 Ltd., [rda@lemma-one.com](mailto:rda@lemma-one.com).

# Index

$=_m$ .....	14, 18, 19	<i>FORM</i> .....	6, 17
$\square$ .....	18	<i>FORM<sub>m</sub></i> .....	12, 18, 19
$\square M_m$ -thm .....	10, 19	<i>FORM<sub>m</sub>-thm</i> .....	12
$\square_m$ .....	9, 18, 19	<i>M<sub>m</sub>-thm</i> .....	10, 19
$\diamond$ .....	18	<i>P-Comp</i> .....	6, 17
$\diamond \square \diamond \exists_p$ -thm .....	15, 20	<i>P-Comp<sub>m</sub></i> .....	12, 18, 19
$\diamond \square \in_m$ -thm .....	15, 20	<i>P-Comp<sub>w</sub><sup>◇</sup></i> .....	13, 18, 19
$\diamond \square \wedge$ -thm .....	15, 20	<i>P-Comp<sup>◇</sup></i> .....	13, 18, 19
$\diamond \square \neg$ -thm .....	15, 20	<i>PL</i> .....	6, 17
$\diamond \square \notin_m$ - $\square$ -thm .....	20	<i>PROP<sub>m</sub></i> .....	18
$\diamond \square eq_m$ -thm .....	15, 20	<i>S</i> .....	6, 17
$\diamond \square_m$ -thm .....	10, 19	<i>set_reachability_thm</i> .....	8, 19
$\diamond \exists_p$ -lemma1 .....	15, 20	<i>Stages</i> .....	8, 18, 19
$\diamond_m$ .....	9, 18, 19	<i>Stages_fc_thm</i> .....	8, 19
$\exists$ .....	18	<i>t046a</i> .....	5, 17
$\exists_p$ .....	11, 18, 19	<i>t046b</i> .....	7, 18
$\exists_p$ -thm .....	11, 20	<i>ThisStage</i> .....	9, 18, 19
$\exists_s$ .....	10, 18, 19		
$\exists_s$ -thm .....	10, 19		
$\forall$ .....	18		
$\forall_p$ .....	11, 18, 19		
$\forall_p$ -thm .....	11, 20		
$\forall_s$ .....	10, 18, 19		
$\forall_s$ -thm .....	10, 19		
$\in$ .....	17, 18		
$\in_m$ .....	12, 18, 19		
$\in_m$ - $\square$ -thm .....	14, 20		
$\in_s$ .....	6, 17		
$\models$ .....	18		
$\models_m$ .....	9, 18, 19		
$\neq \in_m$ - $\square$ -thm .....	14		
$\neq_m$ .....	14, 18, 19		
$\neq_m$ - $\square$ -thm .....	20		
$\notin_m$ .....	18, 19		
$\notin_m$ - $\square$ -thm .....	20		
$<$ .....	17, 18		
$<_m$ .....	11, 18, 19		
$<_p$ .....	6, 17		
<i>A4<sub>m</sub>-thm</i> .....	10, 19		
<i>C4<sub>m</sub>-thm</i> .....	10, 19		
<i>CBF<sub>m</sub>-thm</i> .....	11, 20		
<i>COLLAPSE</i> .....	6, 17		
<i>COLLAPSE_contradiction_thm</i> .....	6, 17		
<i>COLLAPSE_contradiction_thm2</i> .....	12, 20		
<i>COLLAPSE_contradiction_thm3</i> .....	13, 20		
<i>COLLAPSE<sub>m</sub></i> .....	12, 18, 19		
<i>COLLAPSE<sup>◇</sup></i> .....	13, 18, 19		
<i>C<sub>m</sub>-thm</i> .....	10, 19		
<i>distrib<sub>m</sub>-thm</i> .....	10, 19		
<i>D<sub>m</sub>-thm</i> .....	10, 19		
<i>eq <math>\in_m</math>-<math>\square</math>-thm</i> .....	14		
<i>eq<sub>m</sub>-<math>\square</math>-thm</i> .....	20		