NFU and NF in ProofPower-HOL

Roger Bishop Jones

Abstract

Three formalisations in ProofPower-HOL are undertaken of NFU and NF. One is based on Hailperin's axioms. Another tries to follow Quine's original formulation by expressing stratified comprehension as a single higher-order axiom (axiom schemes are not supported by ProofPower. The last is a finite axiomatisation based on one originating with Holmes.

Created 2006/09/25

Last Change Date: 2009/10/15 19:46:58

http://www.rbjones.com/rbjpub/pp/doc/t019.pdf

Id: t019.doc,v 1.11 2009/10/15 19:46:58 rbj Exp

Contents

To Do

- Everything.

References

[1] T.E. Forster. *Set Theory with a Universal Set - Exploring an Untyped Universe*. Oxford University Press, 1992.

[2] M.R. Holmes. *Elementary Set Theory with a Universal Set*, volume 10 of *Cahier du Centre de Logique*. 1998.

[3] Roger Bishop Jones. Membership Structures. *RBJones.com*, 2010. `http://www.rbjones.com/rbjpub/pp/doc/t004.pdf`.

1 INTRODUCTION

The material here concerns non-well-founded formal foundations for mathematics, particularly those with a universal set, and more specifically Quine's NF and its relative NFU.

Though my primary sources of information about NF and NFU are Forster [1] and Holmes [2] the formalisations here follow neither closely. In a previous exploration I transcribed into ProofPower-HOL both a finite axiom system due to Hailperin more or less as presented in Forster [1], and a finite axiomatisation due to Holmes more or less as presented in [2]. These first transcriptions are sufficiently unsatisfactory in various ways that a completely fresh start seems a good idea.

These new approaches have the following aims:

- To attempt an axiomatisation closer to Quine's original in having an explicit axiom of stratified comprehension, rendered as a higher order axiom (axiom schemes are not supported by ProofPower).
- To produce a finite axiomatisation, not involving stratified comprehension, which is more compact and uniform in presentation than my previous rendering of Holmes' system. This is to have some similarities in character to the Hailperin axioms, but substantially based on Holmes and retaining most of the latter's accessibility.

In both cases two theories will be presented, the first being NFU and the second NF, defined as a child of NFU by adding a further axiom. The names of the four theories will be *nfu_s*, *nf_s*, *nfu_f*, *nf_f*, in which the 's' suggests a schema of comprehension and the 'f' that the axiomatisation is finite.

Before addressing these we provide a rendering of Hailperin's axioms.

2 NF FOLLOWING HAILPERIN

This is based on Forster's presentation of Hailperin's axiom in [1], as I have not yet been able to look at the original. I have not understood this, in two different ways. First I do not yet understand how this axiom system works, how for example one might derive from these axioms those of Holmes [2] or demonstrate closure under stratified comprehension. Worse however, I am not at all confident that I have a good rendition of Hailperin's axioms. In particular it is not clear to me how I should be treating the ordered pair and ordered triple constructors.

I understand from Forster firstly that the ordered pair constructor must be construed in the manner of Wiener-Kuratowski, but that the triples are not an iteration of the pair constructor, but are constructed so that all three constituents are at the same depth.

What is not clear is which of these should be taken as primitive and whether the axioms encompass the characterisation of the pairs and triples.

What I have done is treat unordered pairs as primitive and defined the ordered pairs and triples in terms of them, and then I have added an extra axiom for the pairs.

I have as yet done almost nothing with these axioms, they are here for reference.

2.1 Technical Prelude

First of all, we must give the the ML commands to introduce the new theory "nf_h" as a child of the theory "rbjmisc".

SML

```
(* open_theory "rbjmisc"; *)
open_theory "fixp";
force_new_theory "nf_h";
set_merge_pcs["hol1", "'savedthm_cs_∃_proof"];
```

In the context in which the development is taking place there is already a set theory, and we will make occasional use of it. The normal set theoretic symbols are interpreted in this prior set theory, and the new set theory (NF) which we are introducing will therefore use symbols systematically subscripted with a small roman 'n'.

Here is the new type for NF:

SML

```
new_type ("NF", 0);
```

This axioms use membership, the unit set and ordered pair constructors. I will take the ordinary pair constructor as primitive and define unit set and ordered pair in terms of them before introducing the axiom.

SML

```
declare_infix (310, "∈ₙ");
declare_infix (310, "∈");
new_const("∈ₙ", ⌜:NF → NF → BOOL⌝);
declare_alias("∈", ⌜$∈ₙ⌝);
new_const("pairₙ", ⌜:NF × NF → NF⌝);
declare_prefix (320, "ι");
```

HOL Constant

$$\$\boldsymbol{\iota} : NF \to NF$$

$$\forall u\bullet\ \iota\ u = pair_n\ (u,\ u)$$

HOL Constant

$$\boldsymbol{wkp} : NF \times NF \to NF$$

$$\forall u\ v\bullet\ wkp(u,\ v) = pair_n(\iota\ u,\ pair_n(u,\ v))$$

HOL Constant

$$\boldsymbol{Kt} : NF \times NF \times NF \to NF$$

$$\forall u\ v\ w\bullet\ Kt(u,\ v,\ w) = wkp(\iota\ \iota\ u,\ wkp(v,\ w))$$

2.2 The Hailperin Axioms

SML

```
val Ext_n = new_axiom(["Ext_n"],
      ⌜∀u v• (∀x• x ∈ u ⇔ x ∈ v) ⇒ u = v⌝);
val pair_n = new_axiom(["pair_n"],
      ⌜∀u v• (∀x• x ∈ pair_n(u,v) ⇔ x = u ∨ x = v)⌝);
val P1 = new_axiom(["P1"],
      ⌜∀u v• ∃ y• ∀ x• x ∈ y ⇔ ¬ x ∈ u ∨ ¬ x ∈ v⌝);
val P2 = new_axiom(["P2"],
      ⌜∀u•∃v•∀x y• wkp(ι x, ι y) ∈ v ⇔ wkp(x, y) ∈ u⌝);
val P3 = new_axiom(["P3"],
      ⌜∀u•∃v•∀x y z• Kt(x, y, z) ∈ v ⇔ wkp(x, y) ∈ u⌝);
val P4 = new_axiom(["P4"],
      ⌜∀u•∃v•∀x y z• Kt(x, z, y) ∈ v ⇔ wkp(x, y) ∈ u⌝);
val P5 = new_axiom(["P5"],
      ⌜∀u•∃v•∀x y• wkp(x, y) ∈ v ⇔ x ∈ u⌝);
val P6 = new_axiom(["P6"],
      ⌜∀u•∃v•∀x• x ∈ v ⇔ ∀z• wkp(z, ι x) ∈ u⌝);
val P7 = new_axiom(["P7"],
      ⌜∀u•∃v•∀x y• wkp(y, x) ∈ u ⇔ wkp(x, y) ∈ v⌝);
val P8 = new_axiom(["P8"],
      ⌜∃v•∀x• x ∈ v ⇔ ∃y• x = ι y⌝);
val P9 = new_axiom(["P9"],
      ⌜∃v•∀x y• wkp(ι x, y) ∈ v ⇔ x ∈ y⌝);
```

The following axiom says that for every two sets u and v there exists a set y of those elements x which appear in neither u nor v. This is of course the complement of the union of the two sets, and the axiom allows us to define complement and union and various other operations over sets.

2.3 Definitions

Many of the following definitions depend upon a consistency proof, the details of which are not presented.

Axiom P1 allows the definition of complementation and intersection, these two operators are the set theoretic counterpart to negation and conjunction which provide a universal set of operators for the propositional calculus. Axiom P1 therefore leads by itself to the definition of any set operations which correspond to propositional truth functions.

2.3.1 Complement

SML

$$\mid \mathit{declare\_postfix}\ (\mathit{320},\ \texttt{"}^{c}\texttt{"});$$

HOL Constant

$$\mid\ \$^{\boldsymbol{c}} : \mathit{NF} \to \mathit{NF}$$

$$\mid\ \forall u\ x\bullet\ x \in u^{\ c} \Leftrightarrow \neg\ x \in u$$

2.3.2 Intersection

SML

$$\mid \mathit{declare\_infix}(\mathit{310},\ \texttt{"}\cap_n\texttt{"});$$

HOL Constant

$$\mid\ \$\cap_{\boldsymbol{n}} : \mathit{NF} \to \mathit{NF} \to \mathit{NF}$$

$$\mid\ \forall u\ v\ x\bullet\ x \in (u \cap_n v) \Leftrightarrow x \in u \wedge x \in v$$

2.3.3 Union

SML

$$\mid \mathit{declare\_infix}(\mathit{305},\ \texttt{"}\cup_n\texttt{"});$$

HOL Constant

$$\mid\ \$\cup_{\boldsymbol{n}} : \mathit{NF} \to \mathit{NF} \to \mathit{NF}$$

$$\mid\ \forall u\ v\bullet\ u \cup_n v = ((u^{\ c}) \cap_n (v^{\ c}))^{\ c}$$

3 NFU USING STRATIFIED COMPREHENSION

3.1 Introducing the Theory

The notion of stratified comprehension is defined in a previous document [3] and applied here in this axiomatisation. The theory containing this prior material is called 'membership', and 'nfu_s' is therefore introduced as a child of that theory.

SML

```
open_theory "membership";
force_new_theory "nfu_s";
set_merge_pcs["hol1", "'savedthm_cs_∃_proof"];
```

First we introduce a new type for the domain of the set theory. The elements of this type are the sets of NF(U).

SML

```
new_type("nfs", 0);
```

We already have a polymorphic equality which will work over this type, but we will need a constant for the membership relation over this type. The usual membership symbol is already in use for the typed membership in ProofPower-HOL so we use that symbol subscripted with "s" for membership in NF and introduce an alias to allow the subscript to be omitted where no ambiguity results.

SML

```
declare_infix (310, "∈s");
new_const("∈s", ⌜:nfs → nfs → BOOL⌝);
declare_alias("∈", ⌜$∈s⌝);
```

3.2 Weak Extensionality

SML

```
val WkExt_ax = new_axiom (["WkExt"],
      ⌜∀x y:nfs• (∃z• z ∈ x ∨ z ∈ y) ⇒ (x = y ⇔ (∀z• z ∈ x ⇔ z ∈ y))⌝);
```

3.3 Notation for Existence and Comprehension

There is special syntactic support for the typed set theory in ProofPower-HOL, notably for the usual way of writing a set comprehension. The following two definitions will allow us to make use of this notation.

First a notation for asserting the existence of a set in NF(U).

HOL Constant

$$\exists_s : nfs\ SET \to BOOL$$

$$\forall s\bullet \exists_s\ s \Leftrightarrow \exists a{:}nfs\bullet \forall x{:}nfs\bullet\ x \in s \Leftrightarrow x \in_s a$$

Thus ⌜∃s⌝ when applied to a term of type ⌜:nfs SET⌝ asserts the existence of a set in NF(U) (of type ⌜:nfs⌝) with the same extension.

The following constant enables us to pull out of the air a set with some specified extension, and therefore is analogous to the usual set abstraction notation.

HOL Constant

$$\mathbf{\Lambda} : nfs\ SET \to nfs$$

$$\forall s{:}nfs\ SET\bullet \exists_s\ s \Rightarrow \forall e\bullet\ e \in_s \Lambda\ s \Leftrightarrow e \in s$$

Thus applying ⌜Λ⌝ to an object of type ⌜:nfs SET⌝ will yield a value of type ⌜:nfs⌝ which will have the same extension if such a set exists. To make use of this we need to be able to prove that the required set exists.

3.4 Stratified Comprehension

Stratified comprehension is an axiom scheme in first order logic. We do not have support for axiom schemes, but we do have a higher order logic.

If we were axiomatising separation in ZF, a higher order axiom of separation could be used quantifying over properties of sets. This would be slightly stronger than the first order axiom of separation, since not all properties in the range of a higher-order quantifier are expressible as formulae of first order logic. If this extra strength had to be avoided, then the property of properties which consists in their being expressible in first order logic can be defined in higher order logic, and the quantification could be restricted to these properties.

A similar technique can be used for stratified comprehension. The property of being a property expressible by a stratified first order formula is definable in higher order logic, and an axiom could quantify over these properties. This property has been defined in the theory 'membership' in [3].

There is one further complication. The usual schema involves universal quantification over any number of variables which may appear free in the comprehension. To express this in a single higher order axiom requires one further subterfuge.

A variable valuation may be represented as a function from variable names to sets. We quantify over such valuations.

SML

```
val StratComp_ax = new_axiom(["StratComp"],
      ⌜∀va; p:(nfs→(nfs→BOOL))→((CHAR)LIST→nfs)→(nfs)SET•
      Stratified p ⇒ ∃s (p $∈s va)⌝);
```

I need to do some checks on this since the formulation was not straightforward and there is a good chance of errors.

One way of checking is to see that it gives the results one expects, but that doesn't ensure that it is not too strong.

The following conjecture is worth checking. (and so far doesn't look at all plausible!)

SML

```
new_conjecture(["Strat"],
      ⌜(∀va p•   Stratified p ⇒ ∃s (p $∈s va))
      ⇔
      StratCompClosed "" (Universe, $∈s)⌝);
```

3.5 Deriving the Basic Principles

We want to be able to prove all the principles in the finite axiomatisation which we use later in the document. Since they aren't axioms here I shall call them the *Basic Principles*.

They will have the same names here as later, but with the suffix "_thm" rather than "_ax". Details of the proofs are not shown, here is the list of the principles which have been demonstrated:

3.6 Strong Extensionality

We start a new theory for this so that it remains possibly to use nfu.

SML

```
open_theory "nfu_s";
force_new_theory "nf_s";
set_merge_pcs["hol1", "'savedthm_cs_∃_proof"];
```

We could add something weak to upgrade the qualified extensionality of NFU, but instead plump for a full statement of extensionality. We might express this as the implication from equal extensions to equality, but the equivalence with equality on the left is more useful, so we might as well have that.

SML

```
new_axiom(["Extensionality"], ⌜∀x y• x = y ⇔ (∀z• z ∈s x ⇔ z ∈s y)⌝);
```

4 FINITE AXIOMATISATION OF NFU

This is mainly based on Holmes' axiom system [2].

The following points give some indication of the motivation and character of the differences between the system present here and that of Holmes.

- It seems likely that at least the presentation and perhaps also in some degree the substance of Holmes' axioms is pedagogical. We have no such purpose in mind here, and are aiming for a system which is practical for formal machine checked reasoning, but still intelligible.
- As in my previous version, the pragmatics will result in the system being as extensional as possible and not distinguishing between the empty set and the urelements more often than is necessary. This is expected to make automation of reasoning more effective.
- A uniform style of axiomatisation by existential assertion will be adopted. This means that the existence of sets is stipulated before the introduction of operations which yield the stipulated sets.
- The axiom system will be presented, as far as is practical, as a whole prior to the development of the theory, rather than interleaved with the development as it is in [2] and in my previous version.
- The question will be considered of how stratified comprehension can be automated, and the convenience of the axioms for this purpose will be taking into account.

4.1 Axioms I

First some preliminaries, setting up a new theory, a new type for the NF(U) sets and declaring the constant for the membership relation.

SML
```
(* open_theory "hol";*)
open_theory "fixp";
force_new_theory "nfu_f";
force_new_pc "'nfu_f1";
declare_infix (290, "∈nf");
new_type ("SETnf", 0);
new_const ("∈nf", ⌜:SETnf → SETnf → BOOL⌝);
set_merge_pcs["hol1", "'savedthm_cs_∃_proof"];
```

A principle difference in presentation is that this presentation will be primarily *existential*, i.e. consisting of set existence assertions. Holmes' more pedagogical presentation takes the more commmon approach of giving axioms for new set constructors in which the existence claim is implicit.

In the case of NFU there are special details which complicate the characterisation but which are properly irrelevant to the axiom system (it is desirable to assert that operations over sets always yield sets, even where the required extension is empty, but the bald existence claim need not do this).

We would begin with a weak axiom of extensionality for NFU to be strengthened later for NF.
SML
```
val weak_ext_ax = new_axiom(["weak_ext"],
      ⌜∀x y• (∃z• z ∈nf x ∨ z ∈nf y) ⇒ (x = y ⇔ ∀z• z ∈nf x ⇔ z ∈nf y)⌝);
```

Instead of the universe we begin with the empty set:
SML
```
val empty_set_ax = new_axiom(["empty_set"],
      ⌜∃x• ∀y• ¬ y ∈nf x⌝);
```

This nextx axiom is taken from Hailperin and states more concisely than Holmes' the sets form a boolean algebra.
SML
```
val nand_ax = new_axiom(["nand"],
      ⌜∀v w• ∃x• ∀y• y ∈nf x ⇔ ¬ y ∈nf v ∨ ¬ y ∈nf w⌝);

val union_ax = new_axiom(["union"],
      ⌜∀v• ∃x• ∀y• y ∈nf x ⇔ ∃z• z ∈nf v ∧ y ∈nf z⌝);

val unit_ax = new_axiom(["unit"],
      ⌜∀v• ∃x• ∀y• y ∈nf x ⇔ y = v⌝);
```

The axioms from here onward depend upon the ordered pair constructor. I will therefore break here in statement of the axioms to undertake various definitions and some elementary proofs before completing the axioms.

4.2 Definitions

Now we introduce a collection of constants by definition (or at least, by conservative extension). The minor complications which arise here from the possibility of urelements will be apparent.

The subsections here follow the early chapter headings in Holmes [2].

4.2.1 The Set Concept

We are free to take any of the objects with no members as the empty set (all the others will then be urelements). We do this by introding a constant to designate that object.

HOL Constant

$$\varnothing : SET_{nf}$$

$$\forall x \bullet \neg\ x \in_{nf} \varnothing$$

Once we have "chosen" the empty set we can define a predicate which distinguished the sets from the urelements. Alternatively this predicate could have been taken as primitive, instead of our choice of empty set.

HOL Constant

$$\boldsymbol{Set} : SET_{nf} \to BOOL$$

$$\forall x \bullet Set\ x \Leftrightarrow x = \varnothing \vee \exists y \bullet\ y \in_{nf} x$$

We might as well have a name for the other things:

HOL Constant

$$\boldsymbol{Ur} : SET_{nf} \to BOOL$$

$$\forall x \bullet Ur\ x \Leftrightarrow \neg\ Set\ x$$

4.2.2 Comprehension

Notation for set comprehension will probably be useful. We have syntax for sets, which in this context will yield values of type $\ulcorner : SET_{nf}\ SET \urcorner$, so we define two new constants, one to assert the existence of a set with a particular extension, and the other to convert an object of type $\ulcorner : SET_{nf}\ SET \urcorner$ to one of type $\ulcorner : SET_{nf} \urcorner$.

The constant $\ulcorner \exists_{nf} \urcorner$ simply asserts the existence of a set with some particular extension.

HOL Constant

$$\exists_{\boldsymbol{nf}} : SET_{nf}\ SET \to BOOL$$

$$\forall s \bullet \exists_{nf}\ s \Leftrightarrow \exists a \bullet Set\ a \wedge \forall x \bullet\ x \in_{nf} a \Leftrightarrow x \in s$$

The constant $\ulcorner \Upsilon_{nf} \urcorner$ \$simulates set comprehension in NFU. If applied to an extension (as an object of type $\ulcorner : (NFU)SET \urcorner$) for which a set exists it yields that set.

HOL Constant

$$\boldsymbol{\Upsilon_{nf}} : SET_{nf}\ SET \to SET_{nf}$$

$$\forall s \bullet \exists_{nf}\ s \Rightarrow Set\ (\Upsilon_{nf}\ s) \wedge \forall x \bullet\ x \in_{nf} (\Upsilon_{nf}\ s) \Leftrightarrow x \in s$$

Of course this is all mere notation, there are no assertions made here about which sets exist.

4.2.3 Boolean operations on Sets

SML

$$declare\_postfix\ (320,\ "^c");$$

HOL Constant

$$\$^{\boldsymbol{c}} : SET_{nf} \to SET_{nf}$$

$$\forall x\bullet\ Set\ (x\ ^c) \wedge \forall y\bullet\ y \in_{nf} (x\ ^c) \Leftrightarrow \neg\ y \in_{nf} x$$

The Universe is the complement of the empty set.

HOL Constant

$$\boldsymbol{V} : SET_{nf}$$

$$V = \varnothing\ ^c$$

SML

$$declare\_infix(305,\ "\cap_{nf}");$$

HOL Constant

$$\$\cap_{\boldsymbol{nf}} : SET_{nf} \to SET_{nf} \to SET_{nf}$$

$$\forall a\ b\bullet\ Set\ (a \cap_{nf} b) \wedge \forall x\bullet\ x \in_{nf} (a \cap_{nf} b) \Leftrightarrow x \in_{nf} a \wedge x \in_{nf} b$$

We are now able to defin union:

SML

$$declare\_infix(300,\ "\cup_{nf}");$$

HOL Constant

$$\$\cup_{\boldsymbol{nf}} : SET_{nf} \to\ SET_{nf} \to SET_{nf}$$

$$\forall a\ b\bullet\ (a \cup_{nf} b) = (a\ ^c \cap_{nf} b\ ^c)^c$$

SML

$$declare\_infix(300,\ "\backslash\backslash_{nf}");$$

HOL Constant

$$\$\backslash_{\boldsymbol{nf}} : SET_{nf} \to\ SET_{nf} \to SET_{nf}$$

$$\forall a\ b\bullet\ (a \setminus_{nf} b) = a \cap_{nf} (b\ ^c)$$

SML

$$declare\_infix(300,\ "\Delta_{nf}");$$

HOL Constant

$\$\boldsymbol{\Delta_{nf}} : SET_{nf} \to SET_{nf} \to SET_{nf}$

$\forall a\ b \bullet (a\ \Delta_{nf}\ b) = (b \setminus_{nf} a) \cup_{nf} (a \setminus_{nf} b)$

SML

```
declare_infix(290, "⊆nf");
declare_infix(290, "⊂nf");
```

HOL Constant

$\$\boldsymbol{\subseteq_{nf}} : SET_{nf} \to SET_{nf} \to BOOL$

$\forall a\ b \bullet (a \subseteq_{nf} b) \Leftrightarrow \forall x \bullet x \in_{nf} a \Rightarrow x \in_{nf} b$

HOL Constant

$\$\boldsymbol{\subset_{nf}} : SET_{nf} \to SET_{nf} \to BOOL$

$\forall a\ b \bullet (a \subset_{nf} b) \Leftrightarrow (a \subseteq_{nf} b) \wedge \neg\ b \subseteq_{nf} a$

Up until now we have used only three axioms (*weak_ext, empty_set, nand*), now we invoke *union* the distributed union axiom.

I show here as a sample, the consistency proof which was necessary in introducing the specification of union. Normally these proofs are not printed, though they are always present unless the specification can be proven conservative by the proof tool without assistance. Often, as in this case, the proofs are more complex than might have been expected.

SML

```
declare_prefix(310, "⋃nf");

set_goal ([], ⌜∃ $⋃nf:SET_nf → SET_nf•
        ∀x• Set (⋃nf x) ∧ (∀y• y ∈nf (⋃nf x) ⇔ ∃z• z ∈nf x ∧ y ∈nf z)⌝);
a (prove_∃_tac THEN strip_tac);
a (strip_asm_tac (∀_elim ⌜x'⌝ union_ax));
a (∃_tac ⌜if ∃s t• t ∈nf x' ∧ s ∈nf t then x else ∅⌝
        THEN CASES_T ⌜∃s t• t ∈nf x' ∧ s ∈nf t⌝
        (fn x => rewrite_tac [x] THEN strip_asm_tac x));
(* *** Goal "1" *** *)
a (asm_rewrite_tac[get_spec ⌜Set⌝]
        THEN REPEAT strip_tac
        THEN ∃_tac ⌜s⌝
        THEN ∃_tac ⌜t⌝
        THEN REPEAT strip_tac);
(* *** Goal "2" *** *)
a (rewrite_tac [get_spec ⌜Set⌝, get_spec ⌜∅⌝]);
save_cs_∃_thm (pop_thm());
```

HOL Constant

$\$\bigcup_{nf} : SET_{nf} \rightarrow SET_{nf}$

$\forall x \bullet Set\ (\bigcup_{nf}\ x) \wedge (\forall y \bullet\ y \in_{nf}\ (\bigcup_{nf}\ x) \Leftrightarrow \exists z \bullet\ z \in_{nf}\ x \wedge y \in_{nf}\ z)$

4.3 Interim Proof Contexts

It is now possible to automate elementary proofs, typically of algebraic laws in the theory so far. To this end we demonstrate (proofs not presented) various theorems which will be useful and integrate these into a 'proof context'.

The proof and use of theorems about the algebra of sets in nfu is complicated by the presence of urelements in the domain of discourse. This would naturally suggest that this kind of theorem would involve restricted quantification over just the sets. This kind of theorem is harder to apply than one involving unrestricted quantification because of the need to prove that the values being manipulated are all sets. To limit the inconvenience occasioned by these considerations the definitions of the various operations over sets have been written so that they do not care whether their objects are sets or not, but will always yield sets with an extension determined by the extension of the operands. The result of one of these operations will always be a set, even if the operands are not.

The plan is then that the proof of an equation between expressions begins with the demonstration that the operands are sets, which will usually be trivial, and then proceeds by application of a version of extensionality which is good for all sets. The subset relations are defined extensionality so they can be expanded without demonstrating that the arguments are sets.

First we prove the version of extensionality which applies to all sets.

nfu_f_set_ext_thm =
$\vdash \forall\ x\ y \bullet\ Set\ x \wedge Set\ y \Rightarrow (x = y \Leftrightarrow (\forall\ z \bullet\ z \in_{nf}\ x \Leftrightarrow z \in_{nf}\ y))$

Then we provide a theorem to assist in proving that expressions do denote sets.

nfu_f_set_clauses =
$\vdash \forall\ x\ y$
$\bullet\ Set\ \varnothing$
$\wedge\ Set\ V$
$\wedge\ Set\ (x\ ^{c})$
$\wedge\ Set\ (x \cap_{nf}\ y)$
$\wedge\ Set\ (x \cup_{nf}\ y)$
$\wedge\ Set\ (x \setminus_{nf}\ y)$
$\wedge\ Set\ (x\ \Delta_{nf}\ y)$
$\wedge\ Set\ (\bigcup_{nf}\ x)$

Now we need extensional characterisations of the set operators. These are split into two sets, those which are uncontroversial enough to apply normally, and those which are not. The latter group in this case consists of clauses which introduce existential quantifiers.

```
nfu_f_op_ext_clauses1 =
  ⊢ ∀ x y z
    • ¬ z ∈nf ∅
        ∧ z ∈nf V
        ∧ (z ∈nf x ᶜ ⇔ ¬ z ∈nf x)
        ∧ (z ∈nf x ∩nf y ⇔ z ∈nf x ∧ z ∈nf y)
        ∧ (z ∈nf x ∪nf y ⇔ z ∈nf x ∨ z ∈nf y)
        ∧ (z ∈nf x \nf y ⇔ z ∈nf x ∧ ¬ z ∈nf y)
        ∧ (z ∈nf x Δnf y ⇔ z ∈nf x ∧ ¬ z ∈nf y ∨ ¬ z ∈nf x ∧ z ∈nf y)

nfu_f_op_ext_clauses2 =
  ⊢ ∀ x y z
    • (z ∈nf (⋃nf x) ⇔ ∃ y• y ∈nf x ∧ z ∈nf y)
```

We are now able to define a 'proof context' in which we can prove elementary algebraic equations such as those cited towards the end of Chapter 3 of Holmes [2].

```
val it = [
    ⊢ A ∩nf B = B ∩nf A,
    ⊢ A ∪nf B = B ∪nf A,
    ⊢ (A ∩nf B) ∩nf C = A ∩nf B ∩nf C,
    ⊢ (A ∪nf B) ∪nf C = A ∪nf B ∪nf C,
    ⊢ A ∩nf (B ∪nf C) = A ∩nf B ∪nf A ∩nf C,
    ⊢ A ∪nf B ∩nf C = (A ∪nf B) ∩nf (A ∪nf C)
    ⊢ A ∩nf ∅ = ∅,
    ⊢ A ∪nf V = V,
    ⊢ (A ∪nf B) ᶜ = A ᶜ ∩nf B ᶜ,
    ⊢ (A ∩nf B) ᶜ = A ᶜ ∪nf B ᶜ,
    ⊢ A ᶜ ∩nf A = ∅,
    ⊢ A ᶜ ∪nf A = V,
    ⊢ V ᶜ = ∅,
    ⊢ ∅ ᶜ = V,
    ⊢ ⋃nf ∅ = ∅,
    ⊢ ∅ ⊆nf A,
    ⊢ A ⊆nf V]
: THM list
```

The rules shown by Holmes with a variable on the right rather than an expression are not provable since they are false if that variable denotes a urelement. They are provable only as conditional on the variable denoting a set.

These results are provable on demand as required and are not therefore stored in our theory, however some of them can be useful in automatic proof so we prove and store those separately and add them to our proof context.

$$
\begin{array}{l}
nfu\_f\_\varnothing V\_clauses = \\
\quad \vdash \forall A\bullet\ V\ ^{c} = \varnothing \\
\qquad \wedge\ \varnothing\ ^{c} = V \\
\qquad \wedge\ A \cap_{nf} \varnothing = \varnothing \\
\qquad \wedge\ \varnothing \cap_{nf} A = \varnothing \\
\qquad \wedge\ (A\ ^{c}) \cap_{nf} A = \varnothing \\
\qquad \wedge\ A \cap_{nf} (A\ ^{c}) = \varnothing \\
\qquad \wedge\ A \cup_{nf} V = V \\
\qquad \wedge\ V \cup_{nf} A = V \\
\qquad \wedge\ (A\ ^{c}) \cup_{nf} A = V \\
\qquad \wedge\ A \cup_{nf} (A\ ^{c}) = V \\
\qquad \wedge\ A \setminus_{nf} A = \varnothing \\
\qquad \wedge\ A\ \Delta_{nf}\ A = \varnothing \\
\qquad \wedge\ A\ \Delta_{nf}\ (A\ ^{c}) = V \\
\qquad \wedge\ (A\ ^{c})\ \Delta_{nf}\ A = V
\end{array}
$$

4.4 Building Finite Structures

SML

$$
declare\_prefix\ (320,\ \texttt{"}\iota\texttt{"});
$$

HOL Constant

$$
\$\boldsymbol{\iota} : SET_{nf} \rightarrow SET_{nf}
$$

$$
\forall x\bullet\ Set\ (\iota\ x) \wedge \forall y\bullet\ y \in_{nf} (\iota\ x) \Leftrightarrow y = x
$$

$$
\begin{array}{l}
\quad \vdash \forall\ x\bullet\ Set\ (\iota\ x) \\
\quad \vdash \forall\ x\ y\bullet\ x \in_{nf} \iota\ y \Leftrightarrow x = y \\
\quad \vdash \forall\ x\ y\bullet\ \iota\ x = \iota\ y \Leftrightarrow x = y
\end{array}
$$

4.4.1 pairs

The ordered pair constructor appears in the statement of several axioms, and unless we bundle them all together we will have to give a name to it.

We are now able to introduce by definition an ordered pair constructor. The axioms ultimately will be incompatible with the ordered pairs being those of the Wiener-Kuratowski construction. However, this axiomatic constraint which rules them out does not appear until later, and at this point it is possible to introduce ordered pairs using a loose definition which could be interpreted as WK pairs, and hence can be shown consistent using them.

We therefore proceed by first defining unordered and then Wiener-Kuratowski pairs, and then introduce by conservative extension a kind of ordered pair which might be but need not be (and ultimately will be forced not to be) Wiener-Kuratowski.

A pair is the union of two singletons.

HOL Constant

$$\boldsymbol{pair} : SET_{nf} \times SET_{nf} \to SET_{nf}$$

$$\forall x\ y\bullet\ pair\ (x,\ y) = (\iota\ x) \cup_{nf} (\iota\ y)$$

$$pair\_thm =$$
$$\vdash \forall\ a\ b\bullet\ Set\ (pair\ (a,\ b)) \wedge (\forall\ x\bullet\ x \in_{nf} pair\ (a,\ b) \Leftrightarrow x = a \vee x = b)$$

$$\iota\_eq\_pair\_thm =$$
$$\vdash \forall\ a\ b\ c\bullet\ \iota\ a = pair\ (b,\ c) \Leftrightarrow b = a \wedge c = a$$

$$pair\_eq\_\iota\_thm =$$
$$\vdash \forall\ a\ b\ c\bullet\ pair\ (b,\ c) = \iota\ a \Leftrightarrow b = a \wedge c = a$$

$$pair\_eq\_pair\_thm =$$
$$\vdash \forall\ a\ b\ c\ d\bullet\ pair\ (a,\ b) = pair\ (c,\ d) \Leftrightarrow a = c \wedge b = d \vee a = d \wedge b = c$$

HOL Constant

$$\$\boldsymbol{wkp} : SET_{nf} \times SET_{nf} \to SET_{nf}$$

$$\forall x\ y\bullet\ wkp\ (x,\ y) = pair\ (\iota\ x,\ pair\ (x,\ y))$$

The following theorems are proven and used in the introduction of *op*.

$$\vdash \forall\ a\ b\bullet\ Set\ (wkp\ (a,\ b))$$
$$\wedge\ (\forall\ x\bullet\ x \in_{nf} wkp\ (a,\ b) \Leftrightarrow x = \iota\ a \vee x = pair\ (a,\ b))$$

$$\vdash \forall\ a\ b\ c\ d\bullet\ wkp\ (a,\ b) = wkp\ (c,\ d) \Leftrightarrow a = c \wedge b = d$$

Having established that a pair constructor exists using a specific encoding, we can now introduce a new name for a pair constructor which is not tied to this particular coding. The introduction of this constant involves a consistency proof in which the *wkp* is used (script not shown).

HOL Constant

$$\boldsymbol{op} : SET_{nf} \times SET_{nf} \to SET_{nf}$$

$$\forall a\ b\ c\ d\bullet\ op\ (a,\ b) = op\ (c,\ d)$$
$$\Leftrightarrow a = c \wedge b = d$$

HOL Constant

$$\boldsymbol{fst} : SET_{nf} \to SET_{nf}$$

$$\forall a\ b\bullet\ fst\ (op\ (a,\ b)) = a$$

HOL Constant

$$\boldsymbol{snd} : SET_{nf} \to SET_{nf}$$

$$\forall a\ b\bullet\ snd\ (op\ (a,\ b)) = b$$

First we assert the existence of cartesian products:

SML

```
val snd_def = get_spec ⌜snd⌝;
```

4.5 Axioms II

Now that we have ordered pairs we can assert the existence of cartesian products.

SML

```
val ×_ax = new_axiom (["×"],
      ⌜∀u v• ∃w• ∀x• x ∈nf w ⇔ ∃y z• y ∈nf u ∧ z ∈nf v ∧ x = op (y, z)⌝);
```

The following asserts the existence of the inverse (converse?) of a relation.

SML

```
val rel_inv_ax = new_axiom (["rel_inv"],
      ⌜∀u• ∃v• ∀w• w ∈nf v ⇔ ∃x y• w = op (x, y) ∧ op(y, x) ∈nf u⌝);
```

Next, the composition of two relations.

SML

```
val rel_comp_ax = new_axiom (["rel_comp"],
      ⌜∀t u• ∃v• ∀w• w ∈nf v ⇔
      ∃x y z• w = op (x, z) ∧ op(x, y) ∈nf t ∧ op(y, z) ∈nf u⌝);
```

The domain of a relation exists.

SML

```
val dom_ax = new_axiom (["dom"],
      ⌜∀u• ∃v• ∀w• w ∈nf v ⇔ ∃x• op (w, x) ∈nf u⌝);
```

Singleton Images exist.

SML

```
val sing_image_ax = new_axiom (["sing_image"],
      ⌜∀u• ∃v• ∀w• w ∈nf v ⇔ ∃x y• op(x,y) ∈nf u ∧ w = op(ι x, ι y)⌝);
```

The equality relation exists.

SML

```
val eq_ax = new_axiom (["eq"],
      ⌜∃v• ∀w• w ∈nf v ⇔ ∃x• w = op (x, x)⌝);
```

The projection relations exist.

SML

```
val proj_ax = new_axiom (["proj"],
      ⌜∃f s• (∀w• w ∈nf f ⇔ ∃x y• w = op (op (x, y), x))
      ∧ (∀w• w ∈nf s ⇔ ∃x y• w = op (op (x, y), y))⌝);
```

We now should have enough axioms for stratified comprehension, and therefore as much as we would need for NF apart from the strengthening of extensionality.

4.6 Finite Structures Continued

We now define the cartesian product constructor.

SML

$$declare\_infix\ (300,\ "\times_{nf}");$$

HOL Constant

$$\$\times_{\boldsymbol{nf}} : SET_{nf} \to SET_{nf} \to SET_{nf}$$

$$\forall u\ v \bullet\ Set\ (u\ \times_{nf}\ v)$$
$$\wedge$$
$$\forall x \bullet\ x\ \in_{nf}\ u\ \times_{nf}\ v$$
$$\Leftrightarrow$$
$$\exists y\ z \bullet\ y\ \in_{nf}\ u \wedge z\ \in_{nf}\ v \wedge x = op\ (y,\ z)$$

4.7 The Theory of Relations

A relation is a set of ordered pairs.

HOL Constant

$$\boldsymbol{Rel} : SET_{nf} \to BOOL$$

$$\forall r \bullet\ Rel\ r \Leftrightarrow \forall x \bullet\ x\ \in_{nf}\ r \Rightarrow \exists y\ z \bullet\ x = op(y,\ z)$$

The universal relation:

HOL Constant

$$\boldsymbol{V^2} : SET_{nf}$$

$$V^2 = V\ \times_{nf}\ V$$

The complement of a relation:

SML

$$declare\_postfix\ (300,\ "^{rc}");$$

HOL Constant

$$\$^{\boldsymbol{rc}} : SET_{nf} \to SET_{nf}$$

$$\forall R \bullet\ R^{\ rc} = V^2\ \setminus_{nf}\ R$$

Every relation has an inverse.

SML

$$declare\_postfix\ (320,\ "^{-1\,nf}");$$

HOL Constant

$\$^{-1nf} : SET_{nf} \to SET_{nf}$

$\forall r\bullet\ Set(r^{-1\,nf}) \wedge \forall x\bullet\ x \in_{nf} r^{-1\,nf} \Leftrightarrow \exists y\ z\bullet\ x = op(y,\ z) \wedge op(z,\ y) \in_{nf} r$

$rel\_inv\_\varnothing\_thm = \vdash \varnothing^{-1\,nf} = \varnothing$

Relations can be composed. Relational composition is also known as relational product.

HOL Constant

$\$^{\circ}_{9nf} : SET_{nf} \to SET_{nf} \to SET_{nf}$

$\forall r\ s\bullet\ Set(r\ ^{\circ}_{9nf}\ s)$
$\quad \wedge \forall x\bullet\ x \in_{nf} r\ ^{\circ}_{9nf}\ s$
$\quad\quad \Leftrightarrow \exists u\ v\ w\bullet\ x = op(u,\ w) \wedge op(u,\ v) \in_{nf} r \wedge op(v,\ w) \in_{nf} s$

Powers of relations:

SML

$declare\_infix\ (300,\ \texttt{"}\restriction^{r}\texttt{"})$;

HOL Constant

$\$\restriction^{r} : SET_{nf} \to \mathbb{N} \to SET_{nf}$

$\forall R\bullet\ R \restriction^{r} 0 = V^{2} \wedge$
$\forall n\bullet\ R \restriction^{r} (n + 1) = (R \restriction^{r} n)\ ^{\circ}_{9nf}\ R$

The domain of a relation is a set.

HOL Constant

$dom : SET_{nf} \to SET_{nf}$

$\forall r\bullet\ Set(dom\ r)$
$\quad \wedge \forall x\bullet\ x \in_{nf} (dom\ r) \Leftrightarrow \exists y\bullet\ op(x,\ y) \in_{nf} r$

$dom\_\varnothing\_thm = \vdash dom\ \varnothing = \varnothing$

So is the range.

HOL Constant

$rng : SET_{nf} \to SET_{nf}$

$\forall r\bullet\ rng\ r = dom\ (r^{-1\,nf})$

HOL Constant

$field : SET_{nf} \to SET_{nf}$

$\forall r\bullet\ field\ r = (dom\ r) \cup_{nf} (rng\ r)$

$rng\_\varnothing\_thm = \vdash rng\ \varnothing = \varnothing$

4.8 Proof Support II

It is now possible to automate elementary proofs, typically of algebraic laws in the theory so far. To this end we demonstrate (proofs not presented) various theorems which will be useful and integrate these into a 'proof context'.

The proof and use of theorems about the algebra of sets in nfu is complicated by the presence of urelements in the domain of discourse. This would naturally suggest that this kind of theorem would involve restricted quantification over just the sets. This kind of theorem is harder to apply than one involving unrestricted quantification because of the need to prove that the values being manipulated are all sets. To limit the inconvenience occasioned by these considerations the definitions of the various operations over sets have been written so that they do not care whether their objects are sets or not, but will always yield sets with an extension determined by the extension of the operands. The result of one of these operations will always be a set, even if the operands are not.

The plan is then that the proof of an equation between expressions begins with the demonstration that the operands are sets, which will usually be trivial, and then proceeds by application of a version of extensionality which is good for all sets. The subset relations are defined extensionality so they can be expanded without demonstrating that the arguments are sets.

Then we provide a theorem to assist in proving that expressions do denote sets.

$$
\begin{array}{l}
nfu\_f\_set\_clauses = \\
\quad \vdash \forall\ x\ y \\
\quad\quad \bullet\ Set\ \varnothing \\
\quad\quad\quad \wedge\ Set\ V \\
\quad\quad\ \wedge\ Set\ V^{2} \\
\quad\quad\quad \wedge\ Set\ (x\ ^{c}) \\
\quad\quad\quad \wedge\ Set\ (x \cap_{nf} y) \\
\quad\quad\quad \wedge\ Set\ (x \cup_{nf} y) \\
\quad\quad\quad \wedge\ Set\ (x \setminus_{nf} y) \\
\quad\quad\quad \wedge\ Set\ (x\ \Delta_{nf}\ y) \\
\quad\quad\quad \wedge\ Set\ (\bigcup_{nf} x) \\
\quad\quad\ \wedge\ Set\ (\iota\ x) \\
\quad\quad\ \wedge\ Set\ (pair(x,y)) \\
\quad\quad\ \wedge\ Set\ (wkp(x,y)) \\
\quad\quad\ \wedge\ Set\ (x \times_{nf} y) \\
\quad\quad\ \wedge\ Set\ (x\ ^{-1\,nf}) \\
\quad\quad\ \wedge\ Set\ (x\ \circ_{nf}\ y) \\
\quad\quad\quad \wedge\ Set\ (x\ \restriction^{r}\ n) \\
\quad\quad\ \wedge\ Set\ (dom\ x) \\
\quad\quad\ \wedge\ Set\ (rng\ x) \\
\quad\quad\ \wedge\ Set\ (field\ x)
\end{array}
$$

Now we need extensional characterisations of the set operators:

$$
\begin{array}{l}
nfu\_f\_op\_ext\_clauses1 = \\
\quad \vdash \forall\ w\ x\ y\ z \\
\quad\quad \bullet\ \neg\ z \in_{nf} \varnothing \\
\quad\quad\quad \wedge\ z \in_{nf} V
\end{array}
$$

$$
\begin{array}{l}
\quad \wedge\ op(w,\ x) \in_{nf} V^2 \\
\quad\quad \wedge\ (z \in_{nf} x^{\ c} \Leftrightarrow \neg\ z \in_{nf} x) \\
\quad\quad \wedge\ (z \in_{nf} x \cap_{nf} y \Leftrightarrow z \in_{nf} x \wedge z \in_{nf} y) \\
\quad\quad \wedge\ (z \in_{nf} x \cup_{nf} y \Leftrightarrow z \in_{nf} x \vee z \in_{nf} y) \\
\quad\quad \wedge\ (z \in_{nf} x \setminus_{nf} y \Leftrightarrow z \in_{nf} x \wedge \neg\ z \in_{nf} y) \\
\quad\quad \wedge\ (z \in_{nf} x\ \Delta_{nf}\ y \Leftrightarrow z \in_{nf} x \wedge \neg\ z \in_{nf} y \vee \neg\ z \in_{nf} x \wedge z \in_{nf} y) \\
\quad\quad \wedge\ (op(w,\ x) \in_{nf} y \times_{nf} z \Leftrightarrow w \in_{nf} y \wedge x \in_{nf} z) \\
\\
nfu\_f\_op\_ext\_clauses2 = \\
\quad \vdash\ \forall\ x\ y\ z \\
\quad\quad \bullet\ (x \in_{nf} \bigcup_{nf} y \Leftrightarrow (\exists\ z \bullet\ z \in_{nf} y \wedge x \in_{nf} z)) \\
\quad\quad\quad \wedge\ (x \in_{nf} y \times_{nf} z \Leftrightarrow (\exists\ v\ w \bullet\ v \in_{nf} y \wedge w \in_{nf} z \wedge x = op\ (v,\ w))) \\
\quad\quad\quad \wedge\ (x \in_{nf} y^{\ -1\,nf} \Leftrightarrow (\exists\ v\ w \bullet\ x = op\ (v,\ w) \wedge op\ (w,\ v) \in_{nf} y)) \\
\quad\quad\quad \wedge\ (x \in_{nf} y\ \circ_{9nf}\ z \Leftrightarrow \\
\quad\quad\quad\quad\quad (\exists\ u\ v\ w \bullet\ x = op\ (u,\ w) \wedge op\ (u,\ v) \in_{nf} y \wedge op\ (v,\ w) \in_{nf} z)) \\
\quad\quad\quad \wedge\ (x \in_{nf} y\ \restriction^{r}\ (n + 1) \Leftrightarrow \\
\quad\quad\quad\quad\quad (\exists u\ v\ w \bullet\ x = op(u,\ w) \wedge op(u,\ v) \in_{nf} y\ \restriction^{r}\ n \wedge op(v,\ w) \in_{nf} y)) \\
\quad\quad\quad \wedge\ (x \in_{nf} dom\ y \Leftrightarrow (\exists\ u \bullet\ op\ (x,\ u) \in_{nf} y)) \\
\quad\quad\quad \wedge\ (x \in_{nf} rng\ y \Leftrightarrow (\exists\ u \bullet\ op\ (u,\ x) \in_{nf} y)) \\
\quad\quad\quad \wedge\ (x \in_{nf} field\ y \Leftrightarrow (\exists\ u \bullet\ op\ (x,\ u) \in_{nf} y \vee op\ (u,\ x) \in_{nf} y)) \\
\\
\\
nfu\_f\_\varnothing V\_clauses = \\
\quad \vdash\ \ \forall A \bullet\ V^{\ c} = \varnothing \\
\quad\quad \wedge\ \varnothing^{\ c} = V \\
\quad\quad \wedge\ A \cap_{nf} \varnothing = \varnothing \\
\quad\quad \wedge\ \varnothing \cap_{nf} A = \varnothing \\
\quad\quad \wedge\ (A^{\ c}) \cap_{nf} A = \varnothing \\
\quad\quad \wedge\ A \cap_{nf} (A^{\ c}) = \varnothing \\
\quad\quad \wedge\ A \cup_{nf} V = V \\
\quad\quad \wedge\ V \cup_{nf} A = V \\
\quad\quad \wedge\ (A^{\ c}) \cup_{nf} A = V \\
\quad\quad \wedge\ A \cup_{nf} (A^{\ c}) = V \\
\quad\quad \wedge\ A \setminus_{nf} A = \varnothing \\
\quad\quad \wedge\ A\ \Delta_{nf}\ A = \varnothing \\
\quad\quad \wedge\ A\ \Delta_{nf}\ (A^{\ c}) = V \\
\quad\quad \wedge\ (A^{\ c})\ \Delta_{nf}\ A = V \\
\quad\quad \wedge\ \bigcup_{nf} V = V \\
\quad\quad \wedge\ \bigcup_{nf} \varnothing = \varnothing \\
\quad\quad \wedge\ \varnothing^{\ -1\,nf} = \varnothing \\
\quad\quad \wedge\ \varnothing\ \circ_{9nf}\ x = \varnothing \\
\quad\quad \wedge\ x\ \circ_{9nf}\ \varnothing = \varnothing \\
\quad\quad \wedge\ dom\ \varnothing = \varnothing \\
\quad\quad \wedge\ rng\ \varnothing = \varnothing \\
\quad\quad \wedge\ field\ \varnothing = \varnothing
\end{array}
$$

$\wedge\ Rel\ \varnothing$

4.9 Stratified Comprehension

By adopting a finite axiomatisation of NF(U) we have evaded the difficulty of formally expressing in HOL the axiom of stratified comprehension.

The practical issue which arises whether stratified comprehension is an axiom or not, is that of establishing the existence and properties (extension) of sets corresponding to stratified properties. Whether an higher order axiom of stratified comprehension or a finite first order axiom system is adopted there is some work to required to make stratified comprehension workable. Ih Holmes's book [2] this appears as a metatheoretic interlude at about this place in the exposition, in which it is proven that the axioms now in place suffice to prove the existence of a set extensionally identical with each property of sets which is expressible as a stratified first order formula. Once this metatheoretic result is in place, it suffices in an informal treatment to accept the existence of a set once a suitable stratified formula has been exhibited. In our more rigid mechanically checked system, this method will not suffice.

4.10 Introducing Functions

Here I may begin to diverge more from Holmes, since the material on functions does not contribute to the axiom system, and I have in mind the use of functions in the appication of NFU to the construction of category theoretic foundations.

A function is a many-one relation:

HOL Constant

$$\boldsymbol{ManyOne_{nf}} : SET_{nf} \to BOOL$$

$$\forall r\bullet\ ManyOne_{nf}\ r \Leftrightarrow \forall x\bullet\ x \in_{nf} dom\ r \Rightarrow$$
$$\forall y\ z\bullet\ op(x,\ y) \in_{nf} r \wedge op(x,\ z) \in_{nf} r \Rightarrow y = z$$

Identity functions will prove useful later (though we have yet to prove that they exist):

HOL Constant

$$\boldsymbol{id} : SET_{nf} \to SET_{nf}$$

$$\forall a\ x\bullet\ x \in_{nf} (id\ a) = \exists y\bullet\ y \in_{nf} a \wedge x = op(y,\ y)$$

Function application is an infix suffix u.

SML

```
declare_infix (320, "nf");
```

HOL Constant

$$\$_{\boldsymbol{nf}} : SET_{nf} \to SET_{nf} \to SET_{nf}$$

$$\forall f\ a\ v\bullet\ op(a,\ v) \in_{nf} f \Rightarrow f\ _{nf}\ a = v$$

4.11 Cantorian Sets

4.12 Strong Extensionality

Once again, we introduce a new theory for the strongly extensional system, First of all, we must give the the ML commands to introduce the new theory "nff" as a child of the theory "nfu_f".

```
SML
|open_theory "nfu_f";
|force_new_theory "nff";
|set_merge_pcs["hol1", "'savedthm_cs_∃_proof", "'nfu_f1", "'nfu_f2"];
```

The distinguishing feature of NF is that there are no urelements. We might as well make this the new axiom.

```
SML
|val N13_ax = new_axiom (["N13"],
|       ⌜¬ ∃x• Ur x⌝);
```

This of course means that everything is a set...

```
|nff_Set_thm =
|       ⊢ ∀ x• Set x
```

and hence that extensionality is now unconditional:

```
|nff_ext_thm =
|       ⊢ ∀ x y• x = y ⇔ (∀ z• z ∈nf x ⇔ z ∈nf y)
```

At the elementary level we have so far reached this doesn't make a big difference so we will say no more *pro-tem*.

Doesn't make much difference, unless perhaps it actually makes the system inconsistent,

5 The Theory nf_h

5.1 Parents

fixp

5.2 Constants

$\$\in_{\boldsymbol{n}}$	$NF \to NF \to BOOL$
$\boldsymbol{pair_n}$	$NF \times NF \to NF$
$\$\boldsymbol{\iota}$	$NF \to NF$
$\boldsymbol{wkp}$	$NF \times NF \to NF$
$\boldsymbol{Kt}$	$NF \times NF \times NF \to NF$
$\$^{\boldsymbol{c}}$	$NF \to NF$
$\$\cap_{\boldsymbol{n}}$	$NF \to NF \to NF$
$\$\cup_{\boldsymbol{n}}$	$NF \to NF \to NF$

5.3 Aliases

$\in$	$\$\in_n : NF \to NF \to BOOL$

5.4 Types

NF

5.5 Fixity

Right Infix 305:
$\cup_{\boldsymbol{n}}$

Right Infix 310:
$\in_{\boldsymbol{n}}$ $\in$ $\cap_{\boldsymbol{n}}$

Postfix 320: $^{\boldsymbol{c}}$

Prefix 320: $\boldsymbol{\iota}$

5.6 Axioms

$\boldsymbol{Ext_n}$	$\vdash \forall\ u\ v\bullet\ (\forall\ x\bullet\ x \in u \Leftrightarrow x \in v) \Rightarrow u = v$
$\boldsymbol{pair_n}$	$\vdash \forall\ u\ v\ x\bullet\ x \in pair_n\ (u,\ v) \Leftrightarrow x = u \vee x = v$
$\boldsymbol{P1}$	$\vdash \forall\ u\ v\bullet \exists\ y\bullet \forall\ x\bullet\ x \in y \Leftrightarrow \neg\ x \in u \vee \neg\ x \in v$
$\boldsymbol{P2}$	$\vdash \forall\ u$ $\bullet \exists\ v\bullet \forall\ x\ y\bullet\ wkp\ (\iota\ x,\ \iota\ y) \in v \Leftrightarrow wkp\ (x,\ y) \in u$
$\boldsymbol{P3}$	$\vdash \forall\ u$ $\bullet \exists\ v\bullet \forall\ x\ y\ z\bullet\ Kt\ (x,\ y,\ z) \in v \Leftrightarrow wkp\ (x,\ y) \in u$
$\boldsymbol{P4}$	$\vdash \forall\ u$ $\bullet \exists\ v\bullet \forall\ x\ y\ z\bullet\ Kt\ (x,\ z,\ y) \in v \Leftrightarrow wkp\ (x,\ y) \in u$
$\boldsymbol{P5}$	$\vdash \forall\ u\bullet \exists\ v\bullet \forall\ x\ y\bullet\ wkp\ (x,\ y) \in v \Leftrightarrow x \in u$
$\boldsymbol{P6}$	$\vdash \forall\ u\bullet \exists\ v\bullet \forall\ x\bullet\ x \in v \Leftrightarrow (\forall\ z\bullet\ wkp\ (z,\ \iota\ x) \in u)$
$\boldsymbol{P7}$	$\vdash \forall\ u\bullet \exists\ v\bullet \forall\ x\ y\bullet\ wkp\ (y,\ x) \in u \Leftrightarrow wkp\ (x,\ y) \in v$
$\boldsymbol{P8}$	$\vdash \exists\ v\bullet \forall\ x\bullet\ x \in v \Leftrightarrow (\exists\ y\bullet\ x = \iota\ y)$
$\boldsymbol{P9}$	$\vdash \exists\ v\bullet \forall\ x\ y\bullet\ wkp\ (\iota\ x,\ y) \in v \Leftrightarrow x \in y$

5.7 Definitions

$\boldsymbol{\iota}$	$\vdash \forall\ u\bullet\ \iota\ u = pair_n\ (u,\ u)$
$\boldsymbol{wkp}$	$\vdash \forall\ u\ v\bullet\ wkp\ (u,\ v) = pair_n\ (\iota\ u,\ pair_n\ (u,\ v))$
$\boldsymbol{Kt}$	$\vdash \forall\ u\ v\ w\bullet\ Kt\ (u,\ v,\ w) = wkp\ (\iota\ \iota\ u,\ wkp\ (v,\ w))$
$\boldsymbol{c}$	$\vdash \forall\ u\ x\bullet\ x \in u^{\ c} \Leftrightarrow \neg\ x \in u$
$\boldsymbol{\cap_n}$	$\vdash \forall\ u\ v\ x\bullet\ x \in u \cap_n v \Leftrightarrow x \in u \wedge x \in v$
$\boldsymbol{\cup_n}$	$\vdash \forall\ u\ v\bullet\ u \cup_n v = (u^{\ c} \cap_n v^{\ c})^{\ c}$

6 The Theory nfu_s

6.1 Parents

membership

6.2 Children

nf_s

6.3 Constants

$\$\in_{\boldsymbol{s}}$	$((BOOL,\ nfs)\ VA,\ nfs)\ VA$
$\exists_{\boldsymbol{s}}$	$(BOOL,\ nfs\ \mathbb{P})\ VA$
$\boldsymbol{\Lambda}$	$(nfs,\ nfs\ \mathbb{P})\ VA$

6.4 Aliases

$\in$	$\$\in_s : ((BOOL,\ nfs)\ VA,\ nfs)\ VA$

6.5 Types

nfs

6.6 Fixity

Right Infix 310:

$\in_{\boldsymbol{s}}$

6.7 Axioms

WkExt	$\vdash \forall\ x\ y$ $\bullet\ (\exists\ z\bullet\ z \in x \vee z \in y)$ $\Rightarrow (x = y \Leftrightarrow (\forall\ z\bullet\ z \in x \Leftrightarrow z \in y))$
StratComp	$\vdash \forall\ va\ p\bullet\ Stratified\ p \Rightarrow \exists_s\ (p\ \$\in\ va)$

6.8 Definitions

$\exists_{\boldsymbol{s}}$	$\vdash \forall\ s\bullet\ \exists_s\ s \Leftrightarrow (\exists\ a\bullet \forall\ x\bullet\ x \in s \Leftrightarrow x \in a)$
$\boldsymbol{\Lambda}$	$\vdash \forall\ s\bullet\ \exists_s\ s \Rightarrow (\forall\ e\bullet\ e \in \Lambda\ s \Leftrightarrow e \in s)$

7 The Theory nfu_f

7.1 Parents

fixp

7.2 Children

metaci *metapi* *pre_func* *nff*

7.3 Constants

$\$\in_{nf}$	$SET_{nf} \to SET_{nf} \to BOOL$
$\varnothing$	SET_{nf}
$\boldsymbol{Set}$	$SET_{nf} \to BOOL$
$\boldsymbol{Ur}$	$SET_{nf} \to BOOL$
$\exists_{nf}$	$SET_{nf}\ \mathbb{P} \to BOOL$
$\boldsymbol{\Upsilon}_{nf}$	$SET_{nf}\ \mathbb{P} \to SET_{nf}$
$\$^{c}$	$SET_{nf} \to SET_{nf}$
$\boldsymbol{V}$	SET_{nf}
$\$\cap_{nf}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$
$\$\cup_{nf}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$
$\$\setminus_{nf}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$
$\$\boldsymbol{\Delta}_{nf}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$
$\$\subseteq_{nf}$	$SET_{nf} \to SET_{nf} \to BOOL$
$\$\subset_{nf}$	$SET_{nf} \to SET_{nf} \to BOOL$
$\$\bigcup_{nf}$	$SET_{nf} \to SET_{nf}$
$\$\boldsymbol{\iota}$	$SET_{nf} \to SET_{nf}$
$\boldsymbol{pair}$	$SET_{nf} \times SET_{nf} \to SET_{nf}$
$\boldsymbol{wkp}$	$SET_{nf} \times SET_{nf} \to SET_{nf}$
$\boldsymbol{op}$	$SET_{nf} \times SET_{nf} \to SET_{nf}$
$\boldsymbol{fst}$	$SET_{nf} \to SET_{nf}$
$\boldsymbol{snd}$	$SET_{nf} \to SET_{nf}$
$\$\times_{nf}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$
$\boldsymbol{Rel}$	$SET_{nf} \to BOOL$
$\boldsymbol{V^2}$	SET_{nf}
$\$^{\boldsymbol{rc}}$	$SET_{nf} \to SET_{nf}$
$\$^{\boldsymbol{-1nf}}$	$SET_{nf} \to SET_{nf}$
$\$\circ_{\boldsymbol{9nf}}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$
$\$\restriction^{r}$	$SET_{nf} \to \mathbb{N} \to SET_{nf}$
$\boldsymbol{dom}$	$SET_{nf} \to SET_{nf}$
$\boldsymbol{rng}$	$SET_{nf} \to SET_{nf}$
$\boldsymbol{field}$	$SET_{nf} \to SET_{nf}$
$\boldsymbol{ManyOne}_{nf}$	$SET_{nf} \to BOOL$
$\boldsymbol{id}$	$SET_{nf} \to SET_{nf}$
$\$_{nf}$	$SET_{nf} \to SET_{nf} \to SET_{nf}$

7.4 Types

$\boldsymbol{SET}_{nf}$

7.5 Fixity

Right Infix 290:
$\subseteq_{nf}$ $\in_{nf}$ $\subset_{nf}$

Right Infix 300:
$\setminus_{nf}$ $\mathbf{\Delta}_{nf}$ $\times_{nf}$ $\circ_{nf}$ $\cup_{nf}$ $\restriction^{r}$

Right Infix 305:
$\cap_{nf}$

Right Infix 320:
$\mathbf{nf}$

Postfix 300:
$\mathbf{rc}$

Postfix 320:
$\mathbf{c}$ $\mathbf{-1nf}$

Prefix 310:
$\bigcup_{nf}$

Prefix 320:
$\boldsymbol{\iota}$

7.6 Axioms

weak_ext
$\vdash \forall\ x\ y$
$\bullet\ (\exists\ z\bullet\ z \in_{nf} x \vee z \in_{nf} y)$
$\Rightarrow (x = y \Leftrightarrow (\forall\ z\bullet\ z \in_{nf} x \Leftrightarrow z \in_{nf} y))$

empty_set
$\vdash \exists\ x\bullet \forall\ y\bullet \neg\ y \in_{nf} x$

nand
$\vdash \forall\ v\ w\bullet \exists\ x\bullet \forall\ y\bullet\ y \in_{nf} x \Leftrightarrow \neg\ y \in_{nf} v \vee \neg\ y \in_{nf} w$

union
$\vdash \forall\ v\bullet \exists\ x\bullet \forall\ y\bullet\ y \in_{nf} x \Leftrightarrow (\exists\ z\bullet\ z \in_{nf} v \wedge y \in_{nf} z)$

unit
$\vdash \forall\ v\bullet \exists\ x\bullet \forall\ y\bullet\ y \in_{nf} x \Leftrightarrow y = v$

$\times$
$\vdash \forall\ u\ v$
$\bullet \exists\ w$
$\bullet \forall\ x$
$\bullet\ x \in_{nf} w$
$\Leftrightarrow (\exists\ y\ z$
$\bullet\ y \in_{nf} u \wedge z \in_{nf} v \wedge x = op\ (y, z))$

rel_inv
$\vdash \forall\ u$
$\bullet \exists\ v$
$\bullet \forall\ w$
$\bullet\ w \in_{nf} v$
$\Leftrightarrow (\exists\ x\ y\bullet\ w = op\ (x, y) \wedge op\ (y, x) \in_{nf} u)$

rel_comp
$\vdash \forall\ t\ u$
$\bullet \exists\ v$
$\bullet \forall\ w$
$\bullet\ w \in_{nf} v$
$\Leftrightarrow (\exists\ x\ y\ z$
$\bullet\ w = op\ (x, z)$
$\wedge\ op\ (x, y) \in_{nf} t$
$\wedge\ op\ (y, z) \in_{nf} u)$

dom
$\vdash \forall\ u\bullet \exists\ v\bullet \forall\ w\bullet\ w \in_{nf} v \Leftrightarrow (\exists\ x\bullet\ op\ (w, x) \in_{nf} u)$

sing_image
$\vdash \forall\ u$
$\bullet \exists\ v$
$\bullet \forall\ w$
$\bullet\ w \in_{nf} v$
$\Leftrightarrow (\exists\ x\ y$
$\bullet\ op\ (x, y) \in_{nf} u \wedge w = op\ (\iota\ x, \iota\ y))$

eq
$\vdash \exists\ v\bullet \forall\ w\bullet\ w \in_{nf} v \Leftrightarrow (\exists\ x\bullet\ w = op\ (x, x))$

proj $\vdash \exists\ f\ s$
$\bullet\ (\forall\ w\bullet\ w \in_{nf} f \Leftrightarrow (\exists\ x\ y\bullet\ w = op\ (op\ (x,\ y),\ x)))$
$\wedge\ (\forall\ w$
$\bullet\ w \in_{nf} s \Leftrightarrow (\exists\ x\ y\bullet\ w = op\ (op\ (x,\ y),\ y)))$

7.7 Definitions

$\varnothing$ $\vdash \forall\ x\bullet\ \neg\ x \in_{nf} \varnothing$
Set $\vdash \forall\ x\bullet\ Set\ x \Leftrightarrow x = \varnothing \vee (\exists\ y\bullet\ y \in_{nf} x)$
Ur $\vdash \forall\ x\bullet\ Ur\ x \Leftrightarrow \neg\ Set\ x$
$\exists_{nf}$ $\vdash \forall\ s\bullet\ \exists_{nf}\ s \Leftrightarrow (\exists\ a\bullet\ Set\ a \wedge (\forall\ x\bullet\ x \in_{nf} a \Leftrightarrow x \in s))$
Υ_{nf} $\vdash \forall\ s$
$\bullet\ \exists_{nf}\ s$
$\Rightarrow Set\ (\Upsilon_{nf}\ s) \wedge (\forall\ x\bullet\ x \in_{nf} \Upsilon_{nf}\ s \Leftrightarrow x \in s)$
^{c} $\vdash \forall\ x\bullet\ Set\ (x\ ^c) \wedge (\forall\ y\bullet\ y \in_{nf} x\ ^c \Leftrightarrow \neg\ y \in_{nf} x)$
V $\vdash V = \varnothing\ ^c$
$\cap_{nf}$ $\vdash \forall\ a\ b$
$\bullet\ Set\ (a \cap_{nf} b)$
$\wedge\ (\forall\ x\bullet\ x \in_{nf} a \cap_{nf} b \Leftrightarrow x \in_{nf} a \wedge x \in_{nf} b)$
$\cup_{nf}$ $\vdash \forall\ a\ b\bullet\ a \cup_{nf} b = (a\ ^c \cap_{nf} b\ ^c)\ ^c$
$\setminus_{nf}$ $\vdash \forall\ a\ b\bullet\ a \setminus_{nf} b = a \cap_{nf} b\ ^c$
Δ_{nf} $\vdash \forall\ a\ b\bullet\ a\ \Delta_{nf}\ b = (b \setminus_{nf} a) \cup_{nf} a \setminus_{nf} b$
$\subseteq_{nf}$ $\vdash \forall\ a\ b\bullet\ a \subseteq_{nf} b \Leftrightarrow (\forall\ x\bullet\ x \in_{nf} a \Rightarrow x \in_{nf} b)$
$\subset_{nf}$ $\vdash \forall\ a\ b\bullet\ a \subset_{nf} b \Leftrightarrow a \subseteq_{nf} b \wedge \neg\ b \subseteq_{nf} a$
$\bigcup_{nf}$ $\vdash \forall\ x$
$\bullet\ Set\ (\bigcup_{nf} x)$
$\wedge\ (\forall\ y$
$\bullet\ y \in_{nf} \bigcup_{nf} x \Leftrightarrow (\exists\ z\bullet\ z \in_{nf} x \wedge y \in_{nf} z))$
ι $\vdash \forall\ x\bullet\ Set\ (\iota\ x) \wedge (\forall\ y\bullet\ y \in_{nf} \iota\ x \Leftrightarrow y = x)$
pair $\vdash \forall\ x\ y\bullet\ pair\ (x,\ y) = \iota\ x \cup_{nf} \iota\ y$
wkp $\vdash \forall\ x\ y\bullet\ wkp\ (x,\ y) = pair\ (\iota\ x,\ pair\ (x,\ y))$
op $\vdash \forall\ a\ b\ c\ d\bullet\ op\ (a,\ b) = op\ (c,\ d) \Leftrightarrow a = c \wedge b = d$
fst $\vdash \forall\ a\ b\bullet\ fst\ (op\ (a,\ b)) = a$
snd $\vdash \forall\ a\ b\bullet\ snd\ (op\ (a,\ b)) = b$
$\times_{nf}$ $\vdash \forall\ u\ v$
$\bullet\ Set\ (u \times_{nf} v)$
$\wedge\ (\forall\ x$
$\bullet\ x \in_{nf} u \times_{nf} v$
$\Leftrightarrow (\exists\ y\ z$
$\bullet\ y \in_{nf} u \wedge z \in_{nf} v \wedge x = op\ (y,\ z)))$
Rel $\vdash \forall\ r\bullet\ Rel\ r \Leftrightarrow (\forall\ x\bullet\ x \in_{nf} r \Rightarrow (\exists\ y\ z\bullet\ x = op\ (y,\ z)))$
$\mathbf{V^2}$ $\vdash V^2 = V \times_{nf} V$
rc $\vdash \forall\ R\bullet\ R\ ^{rc} = V^2 \setminus_{nf} R$
$^{-1nf}$ $\vdash \forall\ r$
$\bullet\ Set\ (r\ ^{-1\,nf})$
$\wedge\ (\forall\ x$
$\bullet\ x \in_{nf} r\ ^{-1\,nf}$
$\Leftrightarrow (\exists\ y\ z\bullet\ x = op\ (y,\ z) \wedge op\ (z,\ y) \in_{nf} r))$
$\fatsemi_{nf}$ $\vdash \forall\ r\ s$
$\bullet\ Set\ (r \fatsemi_{nf} s)$
$\wedge\ (\forall\ x$

$\bullet\ x \in_{nf} r \circ_{9nf} s$
$\Leftrightarrow (\exists\ u\ v\ w$
$\bullet\ x = op\ (u, w)$
$\wedge\ op\ (u, v) \in_{nf} r$
$\wedge\ op\ (v, w) \in_{nf} s))$

$\restriction^{r}$ $\vdash \forall\ R$
$\bullet\ R \restriction^{r} 0 = V^{2}$
$\wedge\ (\forall\ n \bullet\ R \restriction^{r} n + 1 = (R \restriction^{r} n) \circ_{9nf} R)$

dom $\vdash \forall\ r$
$\bullet\ Set\ (dom\ r)$
$\wedge\ (\forall\ x \bullet\ x \in_{nf} dom\ r \Leftrightarrow (\exists\ y \bullet\ op\ (x, y) \in_{nf} r))$

rng $\vdash \forall\ r \bullet\ rng\ r = dom\ (r^{-1\,nf})$

field $\vdash \forall\ r \bullet\ field\ r = dom\ r \cup_{nf} rng\ r$

ManyOne$_{nf}$ $\vdash \forall\ r$
$\bullet\ ManyOne_{nf}\ r$
$\Leftrightarrow (\forall\ x$
$\bullet\ x \in_{nf} dom\ r$
$\Rightarrow (\forall\ y\ z$
$\bullet\ op\ (x, y) \in_{nf} r \wedge op\ (x, z) \in_{nf} r$
$\Rightarrow y = z))$

id $\vdash ConstSpec$
$(\lambda\ id'$
$\bullet \forall\ a\ x$
$\bullet\ x \in_{nf} id'\ a$
$\Leftrightarrow (\exists\ y \bullet\ y \in_{nf} a \wedge x = op\ (y, y)))$
id

nf $\vdash ConstSpec$
$(\lambda\ {}_{nf}{}' \bullet \forall\ f\ a\ v \bullet\ op\ (a, v) \in_{nf} f \Rightarrow {}_{nf}{}'\ f\ a = v)$
$\$_{nf}$

7.8 Theorems

set_ext_thm $\vdash \forall\ x\ y$
$\bullet\ Set\ x \wedge Set\ y$
$\Rightarrow (x = y \Leftrightarrow (\forall\ z \bullet\ z \in_{nf} x \Leftrightarrow z \in_{nf} y))$

pair_eq_clauses
$\vdash \forall\ a\ b\ c\ d$
$\bullet\ (\iota\ a = pair\ (b, c) \Leftrightarrow b = a \wedge c = a)$
$\wedge\ (pair\ (b, c) = \iota\ a \Leftrightarrow b = a \wedge c = a)$
$\wedge\ (pair\ (a, b) = pair\ (c, d)$
$\Leftrightarrow a = c \wedge b = d \vee a = d \wedge b = c)$
$\wedge\ (wkp\ (a, b) = wkp\ (c, d) \Leftrightarrow a = c \wedge b = d)$
$\wedge\ (op\ (a, b) = op\ (c, d) \Leftrightarrow a = c \wedge b = d)$
$\wedge\ fst\ (op\ (a, b)) = a$
$\wedge\ snd\ (op\ (a, b)) = b$

set_clauses $\vdash \forall\ x\ y\ n$
$\bullet\ Set\ \varnothing$
$\wedge\ Set\ V$
$\wedge\ Set\ V^{2}$
$\wedge\ Set\ (x^{c})$
$\wedge\ Set\ (x \cap_{nf} y)$

$\wedge\ Set\ (x \cup_{nf}\ y)$
$\wedge\ Set\ (x \setminus_{nf}\ y)$
$\wedge\ Set\ (x\ \Delta_{nf}\ y)$
$\wedge\ Set\ (\bigcup_{nf}\ x)$
$\wedge\ Set\ (\iota\ x)$
$\wedge\ Set\ (pair\ (x,\ y))$
$\wedge\ Set\ (wkp\ (x,\ y))$
$\wedge\ Set\ (x \times_{nf}\ y)$
$\wedge\ Set\ (x^{-1\,nf})$
$\wedge\ Set\ (x\ \fatsemi_{nf}\ y)$
$\wedge\ Set\ (x \restriction^{r}\ n)$
$\wedge\ Set\ (dom\ x)$
$\wedge\ Set\ (rng\ x)$
$\wedge\ Set\ (field\ x)$

***op_ext_clauses*1**

$\vdash \forall\ w\ x\ y\ z$
$\bullet \neg\ z \in_{nf} \varnothing$
$\wedge\ z \in_{nf} V$
$\wedge\ op\ (w,\ x) \in_{nf} V^{2}$
$\wedge\ op\ (w,\ x) \in_{nf} y \restriction^{r} 0$
$\wedge\ (z \in_{nf} x^{c} \Leftrightarrow \neg\ z \in_{nf} x)$
$\wedge\ (z \in_{nf} x \cap_{nf} y \Leftrightarrow z \in_{nf} x \wedge z \in_{nf} y)$
$\wedge\ (z \in_{nf} x \cup_{nf} y \Leftrightarrow z \in_{nf} x \vee z \in_{nf} y)$
$\wedge\ (z \in_{nf} x \setminus_{nf} y \Leftrightarrow z \in_{nf} x \wedge \neg\ z \in_{nf} y)$
$\wedge\ (z \in_{nf} x\ \Delta_{nf}\ y$
$\Leftrightarrow z \in_{nf} x \wedge \neg\ z \in_{nf} y$
$\vee \neg\ z \in_{nf} x \wedge z \in_{nf} y)$
$\wedge\ (z \in_{nf} \iota\ x \Leftrightarrow z = x)$
$\wedge\ (z \in_{nf} pair\ (x,\ y) \Leftrightarrow z = x \vee z = y)$
$\wedge\ (op\ (w,\ x) \in_{nf} y \times_{nf} z \Leftrightarrow w \in_{nf} y \wedge x \in_{nf} z)$
$\wedge\ (op\ (w,\ x) \in_{nf} y^{-1\,nf} \Leftrightarrow op\ (x,\ w) \in_{nf} y)$

***op_ext_clauses*2**

$\vdash \forall\ x\ y\ z\ n$
$\bullet\ (x \in_{nf} \bigcup_{nf}\ y \Leftrightarrow (\exists\ z\bullet\ z \in_{nf} y \wedge x \in_{nf} z))$
$\wedge\ (x \in_{nf} y \times_{nf} z$
$\Leftrightarrow (\exists\ v\ w\bullet\ v \in_{nf} y \wedge w \in_{nf} z \wedge x = op\ (v,\ w)))$
$\wedge\ (x \in_{nf} y^{-1\,nf}$
$\Leftrightarrow (\exists\ v\ w\bullet\ x = op\ (v,\ w) \wedge op\ (w,\ v) \in_{nf} y))$
$\wedge\ (x \in_{nf} y\ \fatsemi_{nf}\ z$
$\Leftrightarrow (\exists\ u\ v\ w$
$\bullet\ x = op\ (u,\ w)$
$\wedge\ op\ (u,\ v) \in_{nf} y$
$\wedge\ op\ (v,\ w) \in_{nf} z))$
$\wedge\ (x \in_{nf} y \restriction^{r} n + 1$
$\Leftrightarrow (\exists\ u\ v\ w$
$\bullet\ x = op\ (u,\ w)$
$\wedge\ op\ (u,\ v) \in_{nf} y \restriction^{r} n$
$\wedge\ op\ (v,\ w) \in_{nf} y))$
$\wedge\ (x \in_{nf} dom\ y \Leftrightarrow (\exists\ u\bullet\ op\ (x,\ u) \in_{nf} y))$
$\wedge\ (x \in_{nf} rng\ y \Leftrightarrow (\exists\ u\bullet\ op\ (u,\ x) \in_{nf} y))$
$\wedge\ (x \in_{nf} field\ y$

$$\Leftrightarrow (\exists\ u\bullet\ op\ (x,\ u) \in_{nf}\ y \vee op\ (u,\ x) \in_{nf}\ y))$$

$\varnothing V\_clauses$ $\vdash \forall\ A$

- $V\ ^{c} = \varnothing$
  - $\wedge\ \varnothing\ ^{c} = V$
  - $\wedge\ A \cap_{nf} \varnothing = \varnothing$
  - $\wedge\ \varnothing \cap_{nf} A = \varnothing$
  - $\wedge\ A\ ^{c} \cap_{nf} A = \varnothing$
  - $\wedge\ A \cap_{nf} A\ ^{c} = \varnothing$
  - $\wedge\ A \cup_{nf} V = V$
  - $\wedge\ V \cup_{nf} A = V$
  - $\wedge\ A\ ^{c} \cup_{nf} A = V$
  - $\wedge\ A \cup_{nf} A\ ^{c} = V$
  - $\wedge\ A \setminus_{nf} A = \varnothing$
  - $\wedge\ A\ \Delta_{nf}\ A = \varnothing$
  - $\wedge\ A\ \Delta_{nf}\ A\ ^{c} = V$
  - $\wedge\ A\ ^{c}\ \Delta_{nf}\ A = V$
  - $\wedge \bigcup_{nf} V = V$
  - $\wedge \bigcup_{nf} \varnothing = \varnothing$
  - $\wedge\ \varnothing\ ^{-1\,nf} = \varnothing$
  - $\wedge\ \varnothing\ \fatsemi_{nf}\ x = \varnothing$
  - $\wedge\ x\ \fatsemi_{nf}\ \varnothing = \varnothing$
  - $\wedge\ dom\ \varnothing = \varnothing$
  - $\wedge\ rng\ \varnothing = \varnothing$
  - $\wedge\ field\ \varnothing = \varnothing$
  - $\wedge\ Rel\ \varnothing$

8 INDEX