A Higher Order Theory of Well-Founded Sets

Roger Bishop Jones

Abstract

An axiomatic development in ProofPower-HOL of a higher order theory of well-founded sets. This is similar to a higher order ZFC strengthened by the assertion that every set is a member of some other set which is a (standard) model of ZFC.

Created 2007/09/25

Last Change Date: 2012/11/24 20:22:24

http://www.rbjones.com/rbjpub/pp/doc/t023.pdf

Id: t023.doc,v 1.18 2012/11/24 20:22:24 rbj Exp

Contents

1 Introduction

This document provides a theory which is used in various foundational studies, for example in my investigations into non well-founded set theories. So its primary purpose has been pragmatic (in the context of some rather bizzarre objectives).

However, central to my philosophy of *metaphysical positivism* is an analytic method and the necessary philosophical and technical material necessary to explain and underpin, possibly to justify, the method and its applications. This has lead me to begin an attempt at explaining this theory as part of that method, which has not progressed far enough as yet to be of much value. This follows.

This document forms a part of a foundational architecture.

In this introduction I will say a few words about the architecture and how this document fits into it, sufficient to give some motivation for the choices I have made here,

The architecture addresses descriptive languages and their use in deductive reasoning. Such languages and methods are seen as enabling, especially when appropriately supported by digital information processing, advances in the ways in which various important objectives may be realised.

To reason deductively using some language, it is important that the language has a definite abstract semantics, and this is best realised by defining suitable languages using a formal semantics. Unfortunately, a semantic regress results.

Multiple strategems are advocated for the termination of semantic regress. We are concerned here with just one of these. This involves termination of formal semantic regress in an informal semantics, which is itself primarily vested in an abstract ontology.

We therefore present:

- an informal description of an abstract ontology
- an inforaml account of the truth conditions of languages suitable for talking and reasoning about this ontology
- an approximation to this language as a computer supported formalised axiomatic theory

2 Ontology

The aim is for a universal abstract ontology. That is, one to which any other kind of abstract ontology may be reduced. This is to provide the subject matter for languages which may be universal in senses to be discussed.

It is seems natural, for many involved in formal semantics, to regard set theory as a metalanguage of last resort, in which the formal semantics of any other language might be given. To be sure negative results on definability are interpreted as forcing us to consider set theories as languages which come in a heirarchy of strengths, and only the family of languages rather than any particular member of that family has any hope of being universal. This does not exhaust the objections which can be raised against the possibility that set theory (or any other language) can be universal.

2.1 Semantic Universality

The practical utility of a foundation for abstract semantics does not rest on its being universal, but it is nevertheless an interesting possibility.

The kind of universality concerned here is universality for defining abstract semantics. A universal language in this sense would be one in which the semantics of any other language could be rendered. To make this notion of unversality precise you would have to make precise the relevant notion of "language" concerned and that of "semantics". I know of no single contender for this.

It may be worth mentioning a couple of examples.

It is conventional wisdom that the truth predicate of a language cannot be defined in that language. This is certainly demonstrable for first order arithmetic, via formalisation of epeminides' paradox. If this could be generalised, it would yield a negative result on the possibility of a universal language for abstract semantics, and it is sometimes construed as doing so.

2.2 Well-Founded Collections

A well-founded set is any definite collection of well-founded sets.

This is intended as an inductive definition, and hence that only those collections whose well-foundedness follows from this definition are well-founded sets. The term "definite" is important, since without some such qualification the definition would yield a contradiction, since the collection of well-founded sets which it defines would then be itself a well-founded set, and would hence not be well-founded.

This tells us that the notion of well-founded set is essentially open ended, that the formation of well-founded sets never comes to an end, that the well-founded sets do not form a definite collection. The meaning of definite is intended here to be very weak. We do require a set to be a definite collection in the sense that every possible member of the set either is or is not a member. This is what we need for membership to be encapsulated in a boolean relationship. The concept of well-founded set could be made definite by giving a stronger meaning to definite in the above definition, e.g. by incorporating a limit on the size or rank of a definite set. But I know of no natural and intuitively plausible strengthening, limitations of size and rank seem quite arbitrary.

3 Axioms

3.1 Introduction

Galactic set theory is a set theory with "galaxies" (previously known as "universes") axiomatised in Higher Order Logic.

3.1.1 Scope

This document is mainly concerned with the axioms of galactic set theory, but includes in addition some definitions and theorems which might easily have been part of the axiomatisation. In the usual first order axiomatisations of set theory, for example, the $Pair_g$ constructor is introduced axiomatically. In this axiomatisation neither the empty set nor $Pair_g$ are primitive, they are introduced by definition once the axioms have been presented. Same goes for separation and intersection. The theory `gst-ax` created by this document, consists of an axiomatic development of a well-founded set theory in ProofPower HOL, and is created as a child of `basic-hol`. This version of the theory is derived from a previous version in which "pseudo-urelements" were available, and in which the standard set theoretic vocabulary was used (which rendered the theory unusable in combination with the usual ProofPower HOL theory of sets). Pseudo-urelements were dropped because I don't need them, and, however slight the complication they introduce, its not necessary. To enable this theory

to be used with the standard set theory (properties in set theoretic clothing) the volcabulary has been systematically subscripted with 'g' (for galactic).

3.1.2 Why Galactic?

This document introduces Galactic Set Theory, which is similar to what has elsewhere been called Universal Set Theory (e.g. by Paul M. Cohn in his "Universal Algebra", but I dare say it came from somewhere else). The "universes" in Cohn, and the galaxies here are mostly models of ZFC, except the smallest in which there is no infinite set. The other main difference is that galactic set theory is formulated in higher order logic.

3.2 Membership

The first thing we do is to introduce a new ProofPower theory and, in that theory, the new TYPE SET together with the membership relation and a psuedo-urelement injection.

3.2.1 The Type GS

The sets under consideration will be the elements of a new type *GS* so the first thing we do is to introduce that new type. GS is a *pure well-founded* set theory. Since the theory will not be conservative, we make no attempt to relate the membership of "GS" to any of the types already available.

SML

```
open_theory "rbjmisc";
force_new_theory "gst-ax";
new_parent "U_orders";
new_parent "wf_relp";
new_parent "wf_recp";
force_new_pc "'gst-ax";
merge_pcs ["'savedthm_cs_∃_proof"] "'gst-ax";
set_merge_pcs ["basic_hol", "'gst-ax"];
new_type ("GS", 0);
```

3.3 Membership

The most important constant is membership, which is a relation over the sets. We can't define this constant (in this context) so it is introduced as a new constant (about which nothing is asserted except its name and type) and its properties are introduced axiomatically.

SML

```
new_const ("∈g", ⌜:GS→GS→BOOL⌝);
declare_infix (230,"∈g");
```

I will possibly be making use of two different treatments of well-foundedness (from the theories *U_orders*, and *wf_relp*) and it may be helpful to establish the connection between them.

The following theorem does the trick:

$$\mathbf{UWellFounded\_well\_founded\_thm} =$$
$$\vdash \forall\ \$<<\bullet\ \mathit{UWellFounded}\ \$<<\ \Leftrightarrow\ \mathit{well\_founded}\ \$<<$$

The axioms of extensionality and well-foundedness may be thought of as telling us what kind of thing a set is (later axioms tell us how many of these sets are to be found in our domain of discourse).

3.3.1 Extensionality

The most fundamental property of membership (or is it of sets?) is *extensionality*, which tells us what kind of thing a set is. The axiom tells us that if two sets have the same elements then they are in fact the same set.

SML

$$\mathit{val}\ \mathbf{gs\_ext\_axiom} = \mathit{new\_axiom}\ (["\mathit{gs\_ext\_axiom}"],$$
$$\ulcorner \forall s\ t{:}GS\bullet\ \ s = t \Leftrightarrow \forall e\bullet\ e \in_g s \Leftrightarrow e \in_g t \urcorner);$$

It follows from the definitions of *IsPue* and *IsSet* and *ue_inj_axiom* that nothing is both a set and a urelement, and that urelements are equal iff the values from which they were obtained under Pue are equal.

It is convenient to have a function which gives the extension of a GS set as a SET of GSs.

HOL Constant

$$\mathbf{X_g} : GS \to GS\ SET$$

$$\forall s\bullet\ X_g\ s = \{t \mid t \in_g s\}$$

3.3.2 Well-Foundedness

Wellfoundedness is asserted using the definition in the theory "U_orders", which is conventional in asserting that each non-empty set has a minimal element.

SML

$$\mathit{val}\ \mathbf{gs\_wf\_axiom} = \mathit{new\_axiom}\ (["\mathit{gs\_wf\_axiom}"], \ulcorner \mathit{UWellFounded}\ \$\in_g \urcorner);$$

$$\mathbf{gs\_wf\_thm1} = \quad \vdash \mathit{well\_founded}\ \$\in_g$$
$$\mathbf{gs\_wf\_min\_thm} = \quad \vdash \forall\ x\bullet\ (\exists\ y\bullet\ y \in_g x) \Rightarrow (\exists\ z\bullet\ z \in_g x\ \wedge\ \neg\ (\exists\ v\bullet\ v \in_g z\ \wedge\ v \in_g x))$$
$$\mathbf{gs\_wftc\_thm} = \quad \vdash \mathit{well\_founded}\ (\mathit{tc}\ \$\in_g)$$

SML

$$\mathit{declare\_infix}\ (230,\ "\in_g{}^+");$$

HOL Constant

$$\$\in_{\mathbf{g}}{}^+ : GS \to GS \to BOOL$$

$$\$\in_g{}^+ = \mathit{tc}\ \$\in_g$$

$$\begin{array}{|ll}
\textbf{gs_wftc_thm2} = & \vdash \mathit{well_founded}\ \$\in_g{}^+ \\
\textbf{tc}\in\textbf{_incr_thm} = & \vdash \forall\ x\ y\bullet\ x \in_g y \Rightarrow x \in_g{}^+ y \\
\textbf{tc}\in\textbf{_cases_thm} = & \vdash \forall\ x\ y\bullet\ x \in_g{}^+ y \Leftrightarrow (x \in_g y \vee (\exists\ z\bullet\ x \in_g{}^+ z \wedge z \in_g y)) \\
\textbf{tc}\in\textbf{_trans_thm} = & \vdash \forall\ s\ t\ u\bullet\ s \in_g{}^+ t \wedge t \in_g{}^+ u \Rightarrow s \in_g{}^+ u
\end{array}$$

The resulting induction principle (sometimes called Neotherian induction) is useful.

$$\begin{array}{|ll}
\textbf{gs_wf_ind_thm} = & \vdash \forall\ p\bullet\ (\forall\ x\bullet\ (\forall\ y\bullet\ y \in_g x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x\bullet\ p\ x) \\
\textbf{gs_cv_ind_thm} = & \vdash \forall\ p\bullet\ (\forall\ x\bullet\ (\forall\ y\bullet\ tc\ \$\in_g\ y\ x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x\bullet\ p\ x) \\
\textbf{gs_cv_ind_thm2} = & \vdash \forall\ p\bullet\ (\forall\ x\bullet\ (\forall\ y\bullet\ y \in_g{}^+ x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x\bullet\ p\ x)
\end{array}$$

But we can get induction tactics directly from the well-foundedness theorems:

SML

$$\begin{array}{|l}
\mathit{val}\ \textbf{GS_INDUCTION_T} = \mathit{WF_INDUCTION_T}\ \mathit{gs_wf_thm1}; \\
\mathit{val}\ \textbf{gs_induction_tac} = \mathit{wf_induction_tac}\ \mathit{gs_wf_thm1}; \\
\mathit{val}\ \textbf{GS_INDUCTION_T2} = \mathit{WF_INDUCTION_T}\ \mathit{gs_wftc_thm2}; \\
\mathit{val}\ \textbf{gs_induction_tac2} = \mathit{wf_induction_tac}\ \mathit{gs_wftc_thm2};
\end{array}$$

$$\begin{array}{|ll}
\textbf{wf_l1} = & \vdash \forall\ x{:}GS\bullet\ \neg\ x \in_g x \\
\textbf{wf_l2} = & \vdash \forall\ x\ y{:}GS\bullet\ \neg\ (x \in_g y \wedge y \in_g x) \\
\textbf{wf_l3} = & \vdash \forall\ x\ y\ z{:}GS\bullet\ \neg\ (x \in_g y \wedge y \in_g z \wedge z \in_g x)
\end{array}$$

3.4 The Ontology Axiom

The remaining axioms are intended to ensure that the subset is a large and well-rounded subset of the cumulative heirarchy. This is to be accomplished by defining a Galaxy as a set satisfying certain closure properties and then asserting that every set is a member of some Galaxy. It is convenient to introduce new constants and axioms for each of the Galactic closure properties before asserting the existence of the Galaxies.

Here we define the subset relation and assert the existence of powersets and unions.

3.4.1 Subsets

A subset s of t is a set all of whose members are also members of t.

SML

$$\begin{array}{|l}
\mathit{declare_infix}\ (230,\text{"}\subseteq_g\text{"}); \\
\mathit{declare_infix}\ (230,\text{"}\subset_g\text{"});
\end{array}$$

HOL Constant

$$\begin{array}{|l}
\$\subseteq_{\boldsymbol{g}} : GS \to GS \to BOOL \\
\hline
\forall s\ t\bullet\ s \subseteq_g t \Leftrightarrow \forall e\bullet\ e \in_g s \Rightarrow e \in_g t
\end{array}$$

HOL Constant

$$\$\subset_{\boldsymbol{g}} : GS \to GS \to BOOL$$

$$\forall s\ t\bullet\ s \subset_g t \Leftrightarrow s \subseteq_g t \wedge \neg\ t \subseteq_g s$$

$$\begin{array}{ll}
\subseteq_{\boldsymbol{g}}\textbf{-}\boldsymbol{thm} = & \vdash \forall\ s\ t\bullet\ s \subseteq_g t \Leftrightarrow (\forall\ e\bullet\ e \in_g s \Rightarrow e \in_g t) \\
\subseteq_{\boldsymbol{g}}\textbf{-}\boldsymbol{eq}\textbf{-}\boldsymbol{thm} = & \vdash \forall\ A\ B\bullet\ A = B \Leftrightarrow A \subseteq_g B \wedge B \subseteq_g A \\
\subseteq_{\boldsymbol{g}}\textbf{-}\boldsymbol{refl}\textbf{-}\boldsymbol{thm} = & \vdash \forall\ A\bullet\ A \subseteq_g A \\
\in_{\boldsymbol{g}}\subseteq_{\boldsymbol{g}}\textbf{-}\boldsymbol{thm} = & \vdash \forall\ e\ A\ B\bullet\ e \in_g A \wedge A \subseteq_g B \Rightarrow e \in_g B \\
\subseteq_{\boldsymbol{g}}\textbf{-}\boldsymbol{trans}\textbf{-}\boldsymbol{thm} = & \vdash \forall\ A\ B\ C\bullet\ A \subseteq_g B \wedge B \subseteq_g C \Rightarrow A \subseteq_g C \\
\boldsymbol{not}\textbf{-}\boldsymbol{psub}\textbf{-}\boldsymbol{thm} = & \vdash \forall\ x\bullet\ \neg\ x \subset_g x
\end{array}$$

HOL Constant

$$\subseteq_{\boldsymbol{g}}\textbf{-}\boldsymbol{closed} : GS \to BOOL$$

$$\forall s\bullet\ \subseteq_g\text{-}closed\ s \Leftrightarrow \forall e\ f\bullet\ e \in_g s \wedge f \subseteq_g e \Rightarrow f \in_g s$$

3.4.2 The Ontology Axiom

We now specify with a single axiom the closure requirements which ensure that our universe is adequately populated. The ontology axiom states that every set is a member of some galaxy which is transitive and closed under formation of powersets and unions and under replacement.

The formulation of replacement only makes membership of a galaxy dependent on the range being contained in the galaxy, it asserts unconditionally the sethood of the image of a set under a functional relation.

SML

```
val Ontology_axiom =
      new_axiom (["Ontology_axiom"],
⌜ ∀s•
      ∃g• s ∈g g
∧
      ∀t• t ∈g g ⇒ t ⊆g g
      ∧ (∃p• (∀v• v ∈g p ⇔ v ⊆g t) ∧ p ∈g g)
      ∧ (∃u• (∀v• v ∈g u ⇔ ∃w• v ∈g w ∧ w ∈g t) ∧ u ∈g g)
      ∧ (∀rel• ManyOne rel ⇒
            (∃r• (∀v• v ∈g r ⇔ ∃w • w ∈g t ∧ rel w v) ∧
                  (r ⊆g g ⇒ r ∈g g)))⌝
);
```

3.5 PowerSets and Union

Here we define the powerset and union operators.

3.5.1 PowerSets

HOL Constant

$\mathbb{P}_{\boldsymbol{g}}$: $GS \to GS$

$\forall s\ t{:}GS \bullet\ t \in_g \mathbb{P}_g\ s \Leftrightarrow t \subseteq_g s$

$\boldsymbol{s}{\in}\mathbb{P}\boldsymbol{s\_thm} = \quad \vdash \forall\ s \bullet\ s \in_g \mathbb{P}_g\ s$

$\boldsymbol{stc}{\in}\mathbb{P}\boldsymbol{s\_thm} = \quad \vdash \forall\ s \bullet\ s \in_g{}^{+} \mathbb{P}_g\ s$

3.5.2 Union

HOL Constant

$\bigcup_{\boldsymbol{g}}$: $GS \to GS$

$\forall s\ t \bullet\ t \in_g \bigcup_g s \Leftrightarrow \exists e \bullet\ t \in_g e \wedge e \in_g s$

$\in_{\boldsymbol{g}}\bigcup_{\boldsymbol{g}}\boldsymbol{\_thm} = \quad \vdash \forall s\ t{:}GS \bullet\ t \in_g s \Rightarrow t \subseteq_g \bigcup_g s$

3.6 Relational Replacement

The constant RelIm is defined for relational replacement.

3.6.1 RelIm

HOL Constant

$\boldsymbol{RelIm}$: $(GS \to GS \to BOOL) \to GS \to GS$

$\forall rel\ s\ t \bullet\ ManyOne\ rel \Rightarrow (t \in_g RelIm\ rel\ s \Leftrightarrow \exists e \bullet\ e \in_g s \wedge rel\ e\ t)$

3.7 Separation

Separation is introduced by conservative extension.

The specification of *Sep* which follows is introduced after proving that it is satisfied by a term involving the use of $RelIm$.

This higher order formulation of separation is accomplished by defining a new constant which takes a property of sets p and a set s and returns the subset of s consisting of those elements which satisfy p.

HOL Constant

$\boldsymbol{Sep}$: $GS \to (GS \to BOOL) \to GS$

$\forall s\ p\ e \bullet\ e \in_g (Sep\ s\ p) \Leftrightarrow e \in_g s \wedge p\ e$

$\boldsymbol{Sep\_sub\_thm} = \quad \vdash \forall\ s\ p \bullet\ Sep\ s\ p \subseteq_g s$

$\boldsymbol{Sep\_sub\_thm2} = \quad \vdash \forall\ s\ p\ e \bullet\ e \in_g Sep\ s\ p \Rightarrow e \in_g s$

$\boldsymbol{Sep\_\in\_\mathbb{P}\_thm} = \vdash \forall\ s\ p \bullet\ Sep\ s\ p \in_g \mathbb{P}_g\ s$

$\boldsymbol{Sep\_\subseteq\_thm} = \quad \vdash \forall\ s\ t \bullet\ t \subseteq_g s \Rightarrow Sep\ s\ (CombC\ \$\in_g\ t) = t$

3.8 Galaxies

A Galaxy is a transitive set closed under powerset formation, union and replacement. The Ontology axioms ensures that every set is a member of some galaxy. Here we define a galaxy constructor and establish some of its properties.

3.8.1 Definition of Galaxy

First we define the property of being a galaxy.

HOL Constant

$$\boldsymbol{galaxy}: GS \to BOOL$$

$$\forall s \bullet$$
$$galaxy\ s \Leftrightarrow (\exists x \bullet\ x \in_g s)$$
$$\wedge\ \forall t \bullet\ t \in_g s$$
$$\Rightarrow\ t \subseteq_g s$$
$$\wedge\ \mathbb{P}_g\ t \in_g s$$
$$\wedge\ \bigcup_g\ t \in_g s$$
$$\wedge\ (\forall rel \bullet\ ManyOne\ rel$$
$$\Rightarrow\ RelIm\ rel\ t \subseteq_g s$$
$$\Rightarrow\ RelIm\ rel\ t \in_g s)$$

$$\boldsymbol{galaxies\_\exists\_thm} =$$
$$\vdash\ \forall s \bullet\ \exists g \bullet\ s \in_g g\ \wedge\ galaxy\ g$$

3.8.2 Definition of Gx

Gx is a function which maps each set onto the smallest galaxy of which it is a member.

HOL Constant

$$\boldsymbol{Gx}: GS \to GS$$

$$\forall s\ t \bullet\ t \in_g Gx\ s \Leftrightarrow \forall g \bullet\ galaxy\ g\ \wedge\ s \in_g g \Rightarrow t \in_g g$$

Each set is in its smallest enclosing galaxy, which is of course a galaxy and is contained in any other galaxy of which that set is a member:

$$\boldsymbol{t\_in\_Gx\_t\_thm} = \quad \vdash\ \forall\ t \bullet\ t \in_g Gx\ t$$
$$\boldsymbol{tc{\in}\_Gx\_thm} = \quad \vdash\ \forall\ t \bullet\ t \in_g{}^{+}\ Gx\ t$$
$$\boldsymbol{galaxy\_Gx} = \quad \vdash\ \forall s \bullet\ galaxy\ (Gx\ s)$$
$$\boldsymbol{Gx\_{\subseteq}_g\_galaxy} = \quad \vdash\ \forall s\ g \bullet\ galaxy\ g\ \wedge\ s \in_g g\ \Rightarrow (Gx\ s) \subseteq_g g$$

3.8.3 Galaxy Closure

The galaxy axiom asserts that a Galaxy is transitive and closed under construction of powersets, distributed union and replacement. Galaxies are also closed under most other ways of constructing sets, and we need to demonstrate these facts systematically as the theory is developed.

HOL Constant

$$\boldsymbol{transitive} : GS \to BOOL$$

$$\forall s\bullet\ transitive\ s \Leftrightarrow \forall e\bullet\ e \in_g s \Rightarrow e \subseteq_g s$$

$$\boldsymbol{GalaxiesTransitive\_thm} = \quad \vdash \forall s\bullet\ galaxy\ s \Rightarrow transitive\ s$$

$$\boldsymbol{GClose\mathbb{P}\_thm} = \quad \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ s\bullet\ s \in_g g \Rightarrow \mathbb{P}_g\ s \in_g g)$$
$$\boldsymbol{GClose\bigcup\_thm} = \quad \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ s\bullet\ s \in_g g \Rightarrow \bigcup_g s \in_g g)$$
$$\boldsymbol{GCloseSep\_thm} = \quad \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall s\bullet\ s \in_g g \Rightarrow \forall p\bullet\ Sep\ s\ p \in_g g)$$
$$\boldsymbol{GClose\_\subseteq\_thm} = \quad \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ s\bullet\ s \in_g g \Rightarrow (\forall\ t\bullet\ t \subseteq_g s \Rightarrow t \in_g g))$$

$$\boldsymbol{GClose\_fc\_clauses} =$$
$$\vdash \forall\ g$$
$$\bullet\ galaxy\ g$$
$$\Rightarrow (\forall\ s$$
$$\bullet\ s \in_g g$$
$$\Rightarrow \mathbb{P}_g\ s \in_g g$$
$$\wedge \bigcup_g s \in_g g$$
$$\wedge (\forall\ p\bullet\ Sep\ s\ p \in_g g)$$
$$\wedge (\forall\ t\bullet\ t \subseteq_g s \Rightarrow t \in_g g))$$

$$\boldsymbol{tc\in_g\_lemma} = \quad \vdash \forall\ s\ e\bullet\ e \in_g{}^+\ s \Rightarrow (\forall\ t\bullet\ transitive\ t \wedge s \subseteq_g t \Rightarrow e \in_g t)$$
$$\boldsymbol{GClose\_tc\in_g\_thm} = \vdash \forall\ s\ g\bullet\ galaxy\ g \Rightarrow s \in_g{}^+\ g \Rightarrow s \in_g g$$

$$\boldsymbol{Gx\_mono\_thm} = \quad \vdash \forall s\ t\bullet\ s \subseteq_g t \Rightarrow Gx\ s \subseteq_g Gx\ t$$
$$\boldsymbol{Gx\_mono\_thm2} = \quad \vdash \forall s\ t\bullet\ s \in_g t \Rightarrow Gx\ s \subseteq_g Gx\ t$$

$$\boldsymbol{Gx\_trans\_thm} = \quad \vdash \forall\ s\bullet\ transitive\ (Gx\ s)$$
$$\boldsymbol{Gx\_trans\_thm2} = \quad \vdash \forall\ s\ t\bullet\ s \in_g t \Rightarrow s \in_g Gx\ t$$
$$\boldsymbol{Gx\_trans\_thm3} = \quad \vdash \forall\ s\ t\ u\bullet\ s \in_g t \wedge t \in_g Gx\ u \Rightarrow s \in_g Gx\ u$$
$$\boldsymbol{t\_sub\_Gx\_t\_thm} = \quad \vdash \forall\ t\bullet\ t \subseteq_g Gx\ t$$

$$\boldsymbol{Gx\_mono\_thm3} = \quad \vdash \forall\ s\ t\bullet\ s \subseteq_g t \Rightarrow s \subseteq_g Gx\ t$$
$$\boldsymbol{Gx\_mono\_thm4} = \quad \vdash \forall\ s\ t\bullet\ s \subseteq_g t \Rightarrow s \in_g Gx\ t$$

3.8.4 The Empty Set

We can now prove that there is an empty set.

So we define $\ulcorner \varnothing_g \urcorner$ as the empty set:

HOL Constant

$$\varnothing_{\boldsymbol{g}} : GS$$

$$\forall s \bullet \neg\ s \in_g \varnothing_g$$

and the empty set is a member of every galaxy:

$$\boldsymbol{G\varnothing_{gC}} = \quad \vdash \forall\ g \bullet\ galaxy\ g \Rightarrow \varnothing_g \in_g g$$
$$\boldsymbol{\varnothing_g \subseteq_g \text{-} thm} = \quad \vdash \forall\ s \bullet\ \varnothing_g \subseteq_g s$$
$$\boldsymbol{\bigcup_g \varnothing_g \text{-} thm} = \quad \vdash \bigcup_g \varnothing_g = \varnothing_g$$

$$\boldsymbol{\varnothing_g \text{-} spec} = \quad \vdash \forall\ s \bullet \neg\ s \in_g \varnothing_g$$
$$\boldsymbol{mem\_not\_empty\_thm} = \quad \vdash \forall\ m\ n \bullet\ m \in_g n \Rightarrow \neg\ n = \varnothing_g$$
$$\boldsymbol{\varnothing_g \text{-} \in_g \text{-} galaxy\_thm} = \quad \vdash \forall\ x \bullet\ galaxy\ x \Rightarrow \varnothing_g \in_g x$$
$$\boldsymbol{\varnothing_g \text{-} \in_g \text{-} Gx\_thm} = \quad \vdash \forall\ x \bullet\ \varnothing_g \in_g Gx\ x$$

3.9 Functional Replacement

The more convenient form of replacement which takes a function rather than a relation (and hence needs no "ManyOne" caveat) is introduced here.

3.9.1 Introduction

Though a functional formulation of replacement is most convenient for most applications, it has a number of small disadvantages which have persuaded me to stay closer to the traditional formulation of replacement in terms of relations. The more convenient functional version will then be introduced by definition (so if you don't know what I'm talking about, look forward to compare the two versions).

For the record the disadvantages of the functional one (if used as an axiom) are:

1. It can't be used to prove the existence of the empty set.
2. When used to define separation it requires the axiom of choice.

3.9.2 Imagep

Now we prove a more convenient version of replacement which takes a HOL function rather than a relation as its argument. It states that the image of any set under a function is also a set.

$\ulcorner Imagep\ f\ s \urcorner$ is the image of s through f.

HOL Constant

$$\boldsymbol{Imagep} : (GS \to GS) \to GS \to GS$$

$$\forall f\ s\ x \bullet\ x \in_g Imagep\ f\ s \Leftrightarrow \exists e \bullet\ e \in_g s \wedge x = f\ e$$

3.9.3 Galaxy Closure

We now show that galaxies are closed under *Imagep*.

$$\begin{array}{|l} \textbf{\textit{GImagepC}} = \quad \vdash \forall g \bullet \mathit{galaxy}\ g \Rightarrow \forall s \bullet s \in_g g \\ \qquad\qquad\qquad\qquad \Rightarrow \forall f \bullet \mathit{Imagep}\ f\ s \subseteq_g g \Rightarrow \mathit{Imagep}\ f\ s \in_g g \end{array}$$

3.10 Pair and Unit sets

$Pair_g$ is defined using replacement, and Sing (because "Unit" is used elsewhere) using $Pair_g$.

Pairs can be defined as the image of some two element set under a function defined by a conditional. A suitable two element set can be constructed from the empty set using the powerset construction a couple of times. However, having proven that this can be done (details omitted), we can introduce the pair constructor by conservative extension as follows. (the ProofPower tool shows that it has accepted my proof by putting this extension into the "definitions" section of the theory listing).

HOL Constant

$$\begin{array}{|l} \textbf{\textit{Pair}}_{\textbf{\textit{g}}} : GS \to GS \to GS \\ \hline \forall s\ t\ e{:}GS \bullet e \in_g \mathit{Pair}_g\ s\ t \Leftrightarrow e = s \vee e = t \end{array}$$

$$\begin{array}{|ll} \textbf{\textit{Pair}}_{\textbf{\textit{g}}}\textbf{\textit{\_}}\in\textbf{\textit{\_thm}} = & \vdash \forall\ x\ y \bullet x \in_g \mathit{Pair}_g\ x\ y \wedge y \in_g \mathit{Pair}_g\ x\ y \\ \textbf{\textit{Pair}}_{\textbf{\textit{g}}}\textbf{\textit{\_tc}}\in\textbf{\textit{\_thm}} = & \vdash \forall\ s\ t \bullet s \in_g{}^+ \mathit{Pair}_g\ s\ t \wedge t \in_g{}^+ \mathit{Pair}_g\ s\ t \\ \textbf{\textit{Pair}}_{\textbf{\textit{g}}}\textbf{\textit{\_eq\_thm}} = & \quad \vdash \forall\ s\ t\ u\ v \bullet \mathit{Pair}_g\ s\ t = \mathit{Pair}_g\ u\ v \\ & \quad \Leftrightarrow s = u \wedge t = v \vee s = v \wedge t = u \end{array}$$

$$\begin{array}{|ll} \textbf{\textit{GClosePair}}_{\textbf{\textit{g}}} = & \vdash \forall g \bullet \mathit{galaxy}\ g \Rightarrow \forall s\ t \bullet s \in_g g \wedge t \in_g g \\ & \Rightarrow \mathit{Pair}_g\ s\ t \in_g g \end{array}$$

3.10.1 Unit Sets

Since "Unit" is used in the theory of groups I use "Sing" for singleton sets.

HOL Constant

$$\begin{array}{|l} \textbf{\textit{Sing}} : GS \to GS \\ \hline \forall s \bullet \mathit{Sing}\ s = \mathit{Pair}_g\ s\ s \end{array}$$

The following theorem tells you what the members of a unit sets are.

$$\begin{array}{|ll} \textbf{\textit{Sing\_thm}} = & \vdash \forall s\ e \bullet e \in_g \mathit{Sing}\ s \Leftrightarrow e = s \\ \textbf{\textit{Sing\_thm2}} = & \vdash \forall x \bullet x \in_g \mathit{Sing}\ x \\ \textbf{\textit{Sing\_tc}}\in\textbf{\textit{\_thm}} = & \vdash \forall x \bullet x \in_g{}^+ \mathit{Sing}\ x \end{array}$$

The following theorem tells you when two unit sets are equal.

$$\textbf{\textit{Sing\_eq\_thm}} = \vdash \forall s\ t \bullet \mathit{Sing}\ s = \mathit{Sing}\ t \Leftrightarrow s = t$$

3.10.2 Galaxy Closure

$$\vdash \forall g \bullet \mathit{galaxy}\ g \Rightarrow \forall s \bullet s \in_g g \Rightarrow \mathit{Sing}\ s \in_g g \quad \textbf{GCloseSing}$$

3.10.3 Sing-Pair equations

The following theorems tell you when Pairs are really Sings.

$$\vdash \forall s\ t\ u \bullet \mathit{Sing}\ s = \mathit{Pair}_g\ t\ u \Leftrightarrow s = t \wedge s = u \quad \textbf{Sing\_Pair}_g\textbf{\_eq\_thm}$$
$$\vdash \forall s\ t\ u \bullet \mathit{Pair}_g\ s\ t = \mathit{Sing}\ u \Leftrightarrow s = u \wedge t = u \quad \textbf{Pair}_g\textbf{\_Sing\_eq\_thm}$$

3.11 Union and Intersection

Binary union and distributed and binary intersection are defined.

3.11.1 Binary Union

HOL Constant

$$\$\cup_g : GS \to GS \to GS$$

$$\forall s\ t\ e \bullet e \in_g (s \cup_g t) \Leftrightarrow e \in_g s \vee e \in_g t$$

$$\vdash \forall\ A\ B \bullet A \subseteq_g A \cup_g B \wedge B \subseteq_g A \cup_g B \quad \textbf{}\subseteq_g\cup_g\textbf{thm}$$
$$\vdash \forall\ A\ B\ C \bullet A \subseteq_g C \wedge B \subseteq_g C \Rightarrow A \cup_g B \subseteq_g C \quad \cup_g\subseteq_g\textbf{thm1}$$
$$\vdash \forall\ A\ B\ C\ D \bullet A \subseteq_g C \wedge B \subseteq_g D \Rightarrow A \cup_g B \subseteq_g C \cup_g D \quad \cup_g\subseteq_g\textbf{thm2}$$
$$\vdash \forall\ A \bullet A \cup_g \varnothing_g = A \wedge \varnothing_g \cup_g A = A \quad \cup_g\varnothing_g\textbf{clauses}$$

3.11.2 Galaxy Closure

$$\vdash \forall g \bullet \mathit{galaxy}\ g \Rightarrow \forall s\ t \bullet s \in_g g \wedge t \in_g g \Rightarrow s \cup_g t \in_g g \quad \textbf{GClose}\cup_g$$

3.11.3 Distributed Intersection

Distributed intersection doesn't really make sense for the empty set, but under this definition it maps the empty set onto itself.

HOL Constant

$$\bigcap_g : GS \to GS$$

$$\forall s \bullet \bigcap_g s = \mathit{Sep}\ (\bigcup_g s)\ (\lambda x \bullet \forall t \bullet t \in_g s \Rightarrow x \in_g t)$$

$$\vdash \forall x\ s\ e \bullet x \in_g s \Rightarrow (e \in_g \bigcap_g s \Leftrightarrow \forall y \bullet y \in_g s \Rightarrow e \in_g y) \quad \textbf{}\bigcap_g\subseteq_g\textbf{thm}$$
$$\vdash \forall A\ B \bullet A \in_g B \Rightarrow \forall C \bullet (\forall D \bullet D \in_g B \Rightarrow C \subseteq_g D) \Rightarrow C \subseteq_g \bigcap_g B \quad \textbf{}\subseteq_g\bigcap_g\textbf{thm}$$
$$\vdash \bigcap_g \varnothing_g = \varnothing_g \quad \bigcap_g\varnothing_g\textbf{thm}$$

3.11.4 Binary Intersection

Binary intersection could be defined in terms of distributed intersection, but its easier not to.

SML

$$declare\_infix\ (240,\ "\cap_g");$$

HOL Constant

$$\$\cap_{\boldsymbol{g}} : GS \to GS \to GS$$

$$\forall s\ t\bullet\ s\ \cap_g\ t = Sep\ s\ (\lambda x\bullet\ x\ \in_g\ t)$$

3.11.5 Galaxy Closure

$$\boldsymbol{GClose}\bigcap_{\boldsymbol{g}} = \quad \vdash \forall g\bullet\ galaxy\ g \Rightarrow \forall s\bullet\ s\ \in_g\ g \Rightarrow \bigcap_g\ s\ \in_g\ g$$
$$\boldsymbol{GClose}\cap_{\boldsymbol{g}} = \quad \vdash \forall g\bullet\ galaxy\ g \Rightarrow \forall s\ t\bullet\ s\ \in_g\ g\ \wedge\ t\ \in_g\ g \Rightarrow s\ \cap_g\ t\ \in_g\ g$$

$$\cap_{\boldsymbol{g}}\boldsymbol{\_thm} = \quad \vdash \forall s\ t\ e\bullet\ e\ \in_g\ s\ \cap_g\ t \Leftrightarrow e\ \in_g\ s\ \wedge\ e\ \in_g\ t$$
$$\cap_{\boldsymbol{g}}\boldsymbol{\_thm} = \quad \vdash \forall s\ t\ e\bullet\ \ e\ \in_g\ s\ \cap_g\ t \Leftrightarrow e\ \in_g\ s\ \wedge\ e\ \in_g\ t$$

$$\subseteq_{\boldsymbol{g}}\cap_{\boldsymbol{g}}\boldsymbol{\_thm} = \quad \vdash \forall A\ B\bullet\ A\ \cap_g\ B\ \subseteq_g\ A\ \wedge\ A\ \cap_g\ B\ \subseteq_g\ B$$
$$\cap_{\boldsymbol{g}}\subseteq_{\boldsymbol{g}}\boldsymbol{\_thm1} = \quad \vdash \forall A\ B\ C\bullet\ A\ \subseteq_g\ C\ \wedge\ B\ \subseteq_g\ C \Rightarrow A\ \cap_g\ B\ \subseteq_g\ C$$
$$\cap_{\boldsymbol{g}}\subseteq_{\boldsymbol{g}}\boldsymbol{\_thm2} = \quad \vdash \forall A\ B\ C\ D\bullet\ A\ \subseteq_g\ C\ \wedge\ B\ \subseteq_g\ D \Rightarrow (A\ \cap_g\ B)\ \subseteq_g\ (C\ \cap_g\ D)$$
$$\cap_{\boldsymbol{g}}\subseteq_{\boldsymbol{g}}\boldsymbol{\_thm3} = \quad \vdash \forall A\ B\ C\bullet\ C\ \subseteq_g\ A\ \wedge\ C\ \subseteq_g\ B \Rightarrow C\ \subseteq_g\ A\ \cap_g\ B$$

3.11.6 Consequences of Well-Foundedness

$$\boldsymbol{not\_x\_in\_x\_thm} = \quad \vdash \neg\ (\exists\ x\bullet\ x\ \in_g\ x)$$

3.12 Galaxy Closure Clauses

$$\begin{aligned}
&\boldsymbol{GClose_fc_clauses2} = \\
&\quad \vdash \forall\ g \\
&\qquad \bullet\ galaxy\ g \\
&\qquad\quad \Rightarrow (\forall\ s\ t\bullet\ s\ \in_g\ g\ \wedge\ t\ \in_g\ g \Rightarrow Pair_g\ s\ t\ \in_g\ g) \\
&\qquad\qquad \wedge\ (\forall\ s\bullet\ s\ \in_g\ g \Rightarrow Sing\ s\ \in_g\ g) \\
&\qquad\qquad \wedge\ (\forall\ s\ t\bullet\ s\ \in_g\ g\ \wedge\ t\ \in_g\ g \Rightarrow s\ \cup_g\ t\ \in_g\ g) \\
&\qquad\qquad \wedge\ (\forall\ s\bullet\ s\ \in_g\ g \Rightarrow \bigcap_g\ s\ \in_g\ g) \\
&\qquad\qquad \wedge\ (\forall\ s\ t\bullet\ s\ \in_g\ g\ \wedge\ t\ \in_g\ g \Rightarrow s\ \cap_g\ t\ \in_g\ g)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{tc}\in\boldsymbol{_clauses} = \quad & \vdash \forall\ s\bullet\ \ \ s\ \in_g{}^+\ Sing\ s \\
& \wedge\ \ \ s\ \in_g{}^+\ \mathbb{P}_g\ s \\
& \wedge\ \forall t\bullet\ \ \ t\ \in_g{}^+\ Pair_g\ s\ t \\
& \wedge\ \ \ s\ \in_g{}^+\ Pair_g\ s\ t
\end{aligned}$$

3.13 Proof Context

To simplify subsequent proofs a new "proof context" is created enabling automatic use of the results now available.

3.13.1 Principles

The only principle I know of which assists with elementary proofs in set theory is the principle that set theoretic conjectures can be reduced to the predicate calculus by using extensional rules for relations and for operators.

Too hasty a reduction can be overkill and may convert a simple conjecture into an unintelligible morass. We have sometimes in the past used made available two proof contexts, an aggressive extensional one, and a milder non-extensional one. However, the extensional rules for the operators are fairly harmless, expansion is triggered by the extensional rules for the relations (equality and subset), so a proof context containing the former together with a suitable theorem for the latter gives good control.

3.13.2 Theorems Used Recklessly

This is pretty much guesswork, only time will tell whether this is the best collection.

SML

```
val gst_ax_thms = [
        ∅_g_spec,
        get_spec ⌜ℙ_g⌝,
        get_spec ⌜⋃_g⌝,
        Imagep_spec,
        Pair_g_eq_thm,
        get_spec ⌜Pair_g⌝,
        Sing_eq_thm,
        Sing_thm,
        Pair_g_Sing_eq_thm,
        Sing_Pair_g_eq_thm,
        Sep_thm,
        ∪_g_thm,
        ∩_g_thm
];

val gst_opext_clauses =
        (all_∀_intro
        o list_∧_intro
        o (map all_∀_elim))
        gst_ax_thms;
save_thm ("gst_opext_clauses", gst_opext_clauses);
```

3.13.3 Theorems Used Cautiously

The following theorems are too aggressive for general use in the proof context but are needed when attempting automatic proof. When an extensional proof is appropriate it can be initiated by a cautious (i.e. a "once") rewrite using the following clauses, after which the extensional rules in the proof context will be triggered.

SML

```
val gst_relext_clauses =
        (all_∀_intro
        o list_∧_intro
        o (map all_∀_elim))
        [gs_ext_axiom,
        get_spec⌜$⊆g⌝];
save_thm ("gst_relext_clauses", gst_relext_clauses);
```

There are a number of important theorems, such as well-foundedness and galaxy closure which have not been mentioned in this context. The character of these theorems makes them unsuitable for the proof context, their application requires thought.

3.13.4 Automatic Proof

The basic proof automation is augmented by adding a preliminary rewrite with the relational extensionality clauses.

SML

```
fun gst_ax_prove_conv thl =
        TRY_C (pure_rewrite_conv [gst_relext_clauses])
        THEN_C (basic_prove_conv thl);
```

3.13.5 Proof Context 'gst-ax

SML

```
val nost_thms = [galaxy_Gx, t_in_Gx_t_thm];

add_rw_thms (gst_ax_thms @ nost_thms) "'gst−ax";
add_sc_thms (gst_ax_thms @ nost_thms) "'gst−ax";
add_st_thms gst_ax_thms "'gst−ax";
set_pr_conv gst_ax_prove_conv "'gst−ax";
set_pr_tac
        (conv_tac o gst_ax_prove_conv)
        "'gst−ax";
commit_pc "'gst−ax";
```

Using the proof context "'gst-ax" elementary results in gst are now provable automatically on demand. For this reason it is not necessary to prove in advance of needing them results such as the associativity of intersection, since they can be proven when required by an expression of the form "prove rule[] term" which proves *term* and returns it as a theorem. If the required proof context for doing this is not in place the form " *merge_pcs_rule* ["*basic_hol*", *'gst − ax*"] (*prove_rule* []) *term*" may be used. Since this is a little cumbersome we define the function *gst_ax_rule* and illustrate its use as follows:

SML

```
val gst_ax_rule =
      (merge_pcs_rule1
      ["basic_hol", "'gst-ax"]
      prove_rule) [];
val gst_ax_conv =
      MERGE_PCS_C1
      ["basic_hol", "'gst-ax"]
      prove_conv;
val gst_ax_tac =
      conv_tac o gst_ax_conv;
```

3.13.6 Examples

The following are examples of the elementary results which are now proven automatically:
SML

```
gst_ax_rule ⌜
      a ∩g (b ∩g c)
      = (a ∩g b) ∩g c⌝;
gst_ax_rule ⌜a ∩g b ⊆g b⌝;
gst_ax_rule ⌜∅g ∪g b = b⌝;
gst_ax_rule ⌜
      a ⊆g b ∧ c ⊆g d
      ⇒ a ∩g c ⊆g b ∩g d⌝;
gst_ax_rule ⌜Sep b p ⊆g b⌝;
gst_ax_rule ⌜a ⊆g b ⇒
      Imagep f a ⊆g Imagep f b⌝;
```

4 Products and Sums

A new "gst-fun" theory is created as a child of "gst-ax". The theory will contain the definitions of ordered pairs, cartesian product, relations and functions, dependent products (functions), dependent sums (disjoint unions) and related material for general use.
SML

```
open_theory "gst-ax";
force_new_theory "gst-fun";
force_new_pc "'gst-fun";
merge_pcs ["'savedthm_cs_∃_proof"] "'gst-fun";
set_merge_pcs ["basic_hol", "'gst-ax", "'gst-fun"];
```

4.1 Ordered Pairs

SML

```
declare_infix (240,"↦g");
```

I first attempted to define ordered pairs in a more abstract way than by explicit use of the Wiener-Kuratovski representation, but this gace me problems so I eventually switched to the explicit definition.

This influences the development of the theory, since the first thing I do is to replicate the previously used defining properties.

HOL Constant

$$\$\mapsto_{\boldsymbol{g}} : GS \to GS \to GS$$

$$\forall s\ t\bullet\ (s \mapsto_g t) = Pair_g\ (Sing\ s)\ (Pair_g\ s\ t)$$

$$\mapsto_{\boldsymbol{g}}\textbf{\_eq\_thm} = \quad \vdash \forall\ s\ t\ u\ v\bullet\ (s \mapsto_g t = u \mapsto_g v) = (s = u \wedge t = v)$$

$$\boldsymbol{Pair_g}\textbf{\_}\in\textbf{\_}\mapsto_{\boldsymbol{g}}\textbf{\_thm} = \quad \vdash \forall s\ t\bullet\ Pair_g\ s\ t \in_g s \mapsto_g t$$

$$\boldsymbol{Pair_g}\textbf{\_}\in_{\boldsymbol{g}}\textbf{\_}\boldsymbol{Gx}\textbf{\_}\mapsto_{\boldsymbol{g}}\textbf{\_thm} = \quad \vdash \forall\ s\ t\bullet\ Pair_g\ s\ t \in_g Gx\ (s \mapsto_g t)$$

$$\begin{aligned}\mapsto_{\boldsymbol{g}}\textbf{\_spec\_thm} = \quad & \vdash (\forall\ s\ t\ u\ v\bullet\ (s \mapsto_g t = u \mapsto_g v) = (s = u \wedge t = v))\\ & \wedge (\forall\ s\ t\bullet\ Pair_g\ s\ t \in_g s \mapsto_g t)\\ & \wedge (\forall\ s\ t\bullet\ Pair_g\ s\ t \in_g Gx\ (s \mapsto_g t))\end{aligned}$$

$$\mapsto_{\boldsymbol{g}}\textbf{\_}\in_{\boldsymbol{g}}\textbf{\_}\boldsymbol{Gx}\textbf{\_}\boldsymbol{Pair_g}\textbf{\_thm} = \quad \vdash \forall\ s\ t\bullet\ s \mapsto_g t \in_g Gx\ (Pair_g\ s\ t)$$

$$\neg\mapsto_{\boldsymbol{g}}\varnothing_{\boldsymbol{g}}\textbf{\_thm} = \vdash \forall\ x\ y\bullet\ \neg\ x \mapsto_g y = \varnothing_g$$

$$\neg\varnothing_{\boldsymbol{g}}\mapsto_{\boldsymbol{g}}\textbf{\_thm} = \vdash \forall\ x\ y\bullet\ \neg\ \varnothing_g = x \mapsto_g y$$

$$\boldsymbol{GClose}\mapsto_{\boldsymbol{g}}\textbf{\_thm} = \quad \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow s \mapsto_g t \in_g g)$$

$$\boldsymbol{tc}\in\textbf{\_}\mapsto\textbf{\_}\boldsymbol{left}\textbf{\_thm} = \quad \vdash \forall\ s\ t\bullet\ s \in_g{}^+\ s \mapsto_g t$$

$$\boldsymbol{tc}\in\textbf{\_}\mapsto\textbf{\_}\boldsymbol{right}\textbf{\_thm} = \quad \vdash \forall\ s\ t\bullet\ t \in_g{}^+\ s \mapsto_g t$$

4.1.1 Projections

The following functions may be used for extracting the components of ordered pairs.

HOL Constant

$$\boldsymbol{fst\ snd} : GS \to GS$$

$$\forall s\ t\bullet\ fst(s \mapsto_g t) = s \wedge snd(s \mapsto_g t) = t$$

$$\mapsto\textbf{\_}\boldsymbol{tc}\textbf{\_thm} = \quad \vdash \forall\ x\ y\bullet\ tc\ \$\in_g x\ (x \mapsto_g y) \wedge tc\ \$\in_g y\ (x \mapsto_g y)$$

4.1.2 MkPair and MkTriple

It proves convenient to have constructors which take HOL pairs and triples as parameters.

HOL Constant

$$\boldsymbol{MkPair_g} : GS \times GS \to GS$$

$$\forall lr\bullet\ MkPair_g\ lr = (Fst\ lr) \mapsto_g (Snd\ lr)$$

HOL Constant

$$\boldsymbol{MkTriple_g} : GS \times GS \times GS \to GS$$

$$\forall t\bullet\ MkTriple_g\ t = (Fst\ t) \mapsto_g (MkPair_g\ (Snd\ t))$$

4.2 Relations

HOL Constant

$$\boldsymbol{rel} : GS \to BOOL$$

$$\forall x\bullet\ rel\ x \Leftrightarrow \forall y\bullet\ y \in_g x \Rightarrow \exists s\ t\bullet\ y = s \mapsto_g t$$

$$\boldsymbol{rel\_\varnothing_g\_thm} = \vdash rel\ \varnothing_g$$

The domain is the set of elements which are related to something under the relationship.

HOL Constant

$$\boldsymbol{dom} : GS \to GS$$

$$\forall x\bullet\ dom\ x = Sep\ (Gx\ x)\ (\lambda w\bullet\ \exists v\bullet\ w \mapsto_g v \in_g x)$$

$$\begin{aligned}
&\boldsymbol{dom_\varnothing_g_thm} = && \vdash dom\ \varnothing_g = \varnothing_g\\
&\boldsymbol{dom_thm} = && \vdash \forall\ r\ y\bullet\ y \in_g dom\ r \Leftrightarrow (\exists\ x\bullet\ y \mapsto_g x \in_g r)\\
&\boldsymbol{dom_Gx_thm} = && \vdash \forall\ r\bullet\ dom\ r \in_g Gx\ r\\
&\boldsymbol{GClose_dom_thm} = && \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ r\bullet\ r \in_g g \Rightarrow dom\ r \in_g g)
\end{aligned}$$

HOL Constant

$$\boldsymbol{ran} : GS \to GS$$

$$\forall x\bullet\ ran\ x = Sep\ (Gx\ x)\ (\lambda w\bullet\ \exists v\bullet\ v \mapsto_g w \in_g x)$$

$$\begin{aligned}
&\boldsymbol{ran_\varnothing_g_thm} = && \vdash ran\ \varnothing_g = \varnothing_g\\
&\boldsymbol{ran_thm} = && \vdash \forall r\ y\bullet\ y \in_g ran\ r \Leftrightarrow \exists\ x\bullet\ x \mapsto_g y \in_g r\\
&\boldsymbol{GClose_ran_thm} = && \vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ r\bullet\ r \in_g g \Rightarrow ran\ r \in_g g)\\
&\boldsymbol{tc{\in}_ran_thm} = && \vdash \forall\ x\ y\bullet\ x \in_g{}^{+} ran\ y \Rightarrow x \in_g{}^{+} y
\end{aligned}$$

HOL Constant

$$\boldsymbol{field}: GS \to GS$$

$$\forall s\ e\bullet\ e \in_g (field\ s) \Leftrightarrow e \in_g (dom\ s) \vee e \in_g (ran\ s)$$

$$\boldsymbol{field\_\varnothing_g\_thm} = \vdash field\ \varnothing_g = \varnothing_g$$

4.3 Domain and Range Restrictions

SML

```
declare_infix (300, "◁g");
declare_infix (300, "▷g");
declare_infix (300, "⩤g");
declare_infix (300, "⩥g");
```

HOL Constant

$$\$\lhd_{\boldsymbol{g}}\colon GS \to GS \to GS$$

$$\forall s\ r\bullet\ s \lhd_g r = Sep\ r\ (\lambda p\bullet\ fst\ p \in_g s)$$

HOL Constant

$$\$\rhd_{\boldsymbol{g}}\colon GS \to GS \to GS$$

$$\forall s\ r\bullet\ r \rhd_g s = Sep\ r\ (\lambda p\bullet\ snd\ p \in_g s)$$

HOL Constant

$$\$\lhd_{\boldsymbol{g}}\colon GS \to GS \to GS$$

$$\forall s\ r\bullet\ s \lhd_g r = Sep\ r\ (\lambda p\bullet\ \neg\ fst\ p \in_g s)$$

HOL Constant

$$\$\rhd_{\boldsymbol{g}}\colon GS \to GS \to GS$$

$$\forall s\ r\bullet\ r \rhd_g s = Sep\ r\ (\lambda p\bullet\ \neg\ snd\ p \in_g s)$$

SML

```
declare_alias ("◁", ⌜$◁g⌝);
declare_alias ("▷", ⌜$▷g⌝);
declare_alias ("⩤", ⌜$⩤g⌝);
declare_alias ("⩥", ⌜$⩥g⌝);
```

4.4 Dependent Types

Any relation may be regarded as a dependent sum type. When so regarded, each ordered pair in the relation consist with a type-index and a value whose type is that associated with the type.

The indexed set of types, relative to which every pair in the relation is well-typed may be retrieved from the relation as follows.

HOL Constant

$$\boldsymbol{Rel2DepType_g} : GS \to GS$$

$$\begin{array}{l}\forall r\bullet\ Rel2DepType_g\ r = Sep \\ \qquad (Gx\ r) \\ \qquad (\lambda e\bullet\ \exists i\ t{:}GS\bullet \\ \qquad\quad e = i \mapsto_g t \\ \qquad\quad \wedge\ i \in_g dom\ r \\ \qquad\quad \wedge\ (\forall j\bullet\ j \in_g t \Leftrightarrow i \mapsto_g j \in_g r))\end{array}$$

Any similar indexed collection of sets, determines a set of ordered pairs and a set of functions according to the following definitions.

The dependent sums are as follows:

HOL Constant

$$\boldsymbol{DepSum_g} : GS \to GS$$

$$\begin{array}{l}\forall t\bullet\ DepSum_g\ t = Sep \\ \qquad (Gx\ t) \\ \qquad (\lambda e\bullet\ \exists i\ t2\ v{:}GS\bullet \\ \qquad\quad e = i \mapsto_g v \\ \qquad\quad \wedge\ v \in_g t2 \\ \qquad\quad \wedge\ i \mapsto_g t2 \in_g t)\end{array}$$

4.5 Dependent Sums and Cartesian Products

SML

```
declare_binder "Σ_g";
```

HOL Constant

$$\$\boldsymbol{\Sigma_g} : (GS \to GS) \to GS \to GS$$

$$\begin{array}{l}\forall f\ s\bullet\ \ \$\Sigma_g\ f\ s = \bigcup_g\ (\\ \qquad Imagep\quad (\lambda e\bullet\ Imagep\ (\lambda x\bullet\ e \mapsto_g x)\ (f\ e)) \\ \qquad\quad s \\ )\end{array}$$

SML

```
declare_infix(240,"×_g");
```

HOL Constant

$$\$\times_{\boldsymbol{g}} : GS \to GS \to GS$$

$$\forall s\ t\bullet\ s\ \times_g\ t = \bigcup_g\ (\ Imagep\ (\lambda se\bullet\ (Imagep\ (\lambda te\bullet\ se \mapsto_g\ te)\ t))\ s)$$

$$\boldsymbol{f{\mapsto_{gs}}\_thm} = \vdash\ \forall\ s\ t\ p\bullet\ p \in_g\ s\ \times_g\ t \Rightarrow fst\ p \mapsto_g\ snd\ p = p$$

$$\boldsymbol{v{\in_g}{\times_g}\_thm} = \vdash\ \forall\ p\ s\ t\bullet\ p \in_g\ s\ \times_g\ t \Rightarrow fst\ p \in_g\ s\ \wedge\ snd\ p \in_g\ t$$

$$\boldsymbol{{\mapsto_g}{\in_g}{\times_g}\_thm} = \vdash\ \forall\ l\ r\ s\ t\bullet\ l \mapsto_g\ r \in_g\ s\ \times_g\ t \Leftrightarrow (l \in_g\ s\ \wedge\ r \in_g\ t)$$

4.5.1 Relation Space

This is the set of all relations over some domain and codomain, i.e. the power set of the cartesian product.

SML

```
declare_infix(240,"↔g");
```

HOL Constant

$$\$\leftrightarrow_g : GS \to GS \to GS$$

$$\forall s\ t\bullet\ s\ \leftrightarrow_g\ t = \mathbb{P}_g(s\ \times_g\ t)$$

$$\leftrightarrow_g{\subseteq_g}{\times_g}\_thm = \vdash\ \forall s\ t\ r\bullet\ r \in_g\ s\ \leftrightarrow_g\ t \Leftrightarrow r \subseteq_g\ (s\ \times_g\ t)$$

$$\varnothing_g{\in_g}{\leftrightarrow_g}\_thm = \vdash\ \forall s\ t\bullet\ \varnothing_g \in_g\ s\ \leftrightarrow_g\ t$$

4.5.2 Another Pair-Projection Inverse Theorem

Couched in terms of membership of relation spaces.

SML

```
set_goal ([], ⌜∀p r s t•
        p ∈g r ∧
        r ∈g s ↔g t ⇒
        fst(p) ↦g snd(p) = p⌝);
a (prove_tac[
        get_spec ⌜$↔g⌝,
        ⊆g_thm]);
a (REPEAT
        (asm_fc_tac[f↦gs_thm]));
val f↦gs_thm1 =
        save_pop_thm "f↦gs_thm1";
```

4.5.3 Member of Relation Theorem

SML

```
set_goal ([],⌜∀p r s t•
        p ∈g r ∧
        r ∈g s ↔g t ⇒
        fst(p) ∈g s ∧
        snd(p) ∈g t⌝);
a (prove_tac[
        get_spec ⌜$↔g⌝,
        ⊆g_thm]);
a (asm_fc_tac[]);
a (fc_tac[v∈g×g_thm]);
a (asm_fc_tac[]);
a (fc_tac[v∈g×g_thm]);
val ∈g↔g_thm =
        save_pop_thm "∈g↔g_thm";
```

4.5.4 Relational Composition

SML

```
declare_infix (250,"og");
```

HOL Constant

```
$og : GS → GS → GS
```

```
∀f g• f og g =
        Imagep
        (λp• (fst(fst p) ↦g snd(snd p)))
        (Sep (g ×g f) λp• ∃q r s• p = (q ↦g r) ↦g (r ↦g s))
```

o_g_thm =
$\vdash \forall f\ g\ x \bullet\ x \in_g f\ o_g\ g \Leftrightarrow$
$\exists q\ r\ s \bullet\ q \mapsto_g r \in_g g \wedge r \mapsto_g s \in_g f$
$\wedge\ x = q \mapsto_g s$

o_g_thm2 =
$\vdash \forall f\ g\ x\ y \bullet\ x \mapsto_g y \in_g f\ o_g\ g$
$\Leftrightarrow (\exists z \bullet\ x \mapsto_g z \in_g g \wedge z \mapsto_g y \in_g f)$

$o_g\_associative\_thm$ =
$\vdash \forall f\ g\ h \bullet\ (f\ o_g\ g)\ o_g\ h = f\ o_g\ g\ o_g\ h$

$o_g\_rel\_thm$ =
$\vdash \forall r\ s \bullet\ rel\ r \wedge rel\ s \Rightarrow rel\ (r\ o_g\ s)$

4.5.5 Relation Subset of Cartesian Product

$rel\_sub\_cp\_thm$ =
$\vdash \forall x \bullet\ rel\ x \Leftrightarrow (\exists s\ t \bullet\ x \subseteq_g s \times_g t)$

4.6 Functions

Definition of partial and total functions and the corresponding function spaces.

4.6.1 fun

HOL Constant
$fun : GS \to BOOL$

$\forall x \bullet\ fun\ x \Leftrightarrow rel\ x \wedge$
$\forall s\ t\ u \bullet\ s \mapsto_g u \in_g x$
$\wedge\ s \mapsto_g t \in_g x$
$\Rightarrow u = t$

4.6.2 lemmas

$fun\_\varnothing_g\_thm$ =
$\vdash fun\ \varnothing_g$
$o_g\_fun\_thm$ =
$\vdash \forall f\ g \bullet\ fun\ f \wedge fun\ g \Rightarrow fun\ (f\ o_g\ g)$
$ran\_o_g\_thm$ =
$\vdash \forall f\ g \bullet\ ran\ (f\ o_g\ g) \subseteq_g ran\ f$
$dom\_o_g\_thm$ =
$\vdash \forall f\ g \bullet\ dom\ (f\ o_g\ g) \subseteq_g dom\ g$
$dom\_o_g\_thm2$ =
$\vdash \forall f\ g \bullet\ ran\ g \subseteq_g dom\ f \Rightarrow dom\ (f\ o_g\ g) = dom\ g$

4.6.3 Partial Function Space

This is the set of all partial functions (i.e. many one mapings) over some domain and codomain.

SML

```
declare_infix (240, "↦g");
```

HOL Constant

```
$↦g : GS → GS → GS
```

```
∀s t• s ↦g t = Sep (s ↔g t) fun
```

4.6.4 Partial Function Space Non-Empty

First the theorem that the empty set is a partial function over any domain and codomain.

SML

```
set_goal([],
      ⌜∀s t• ∅g ∈g s ↦g t⌝);
a (prove_tac[
      get_spec ⌜$↦g⌝,
      fun_∅g_thm]);
val ∅g∈g↦g_thm =
      save_pop_thm "∅g∈g↦g_thm";
```

And then that every partial function space is non-empty.

SML

```
set_goal([],
      ⌜∀s t• ∃ f• f ∈g s ↦g t⌝);
a (REPEAT strip_tac
      THEN ∃_tac ⌜∅g⌝
      THEN
      rewrite_tac [∅g∈g↦g_thm]);
val ∃↦g_thm =
      save_pop_thm "∃↦g_thm";
```

4.6.5 Function Space

This is the set of all total functions over some domain and codomain.

SML

```
declare_infix (240, "→g");
```

HOL Constant

```
$→g : GS → GS → GS
```

```
∀s t• s →g t = Sep (s ↦g t)
      λr• dom r = s
```

4.6.6 Function Space Non-Empty

First, for the special case of function spaces with empty domain we prove the theorem that the empty set is a member:

SML

```
set_goal([],⌜∀s t• ∅g ∈g ∅g →g t⌝);
a (prove_tac[get_spec ⌜$→g⌝,
        fun_∅g_thm,
        ∅g∈g↠g_thm]);
val ∅g∈g∅g→g_thm =
        save_pop_thm "∅g∈g∅g→g_thm";
```

Then that whenever the codomain is non-empty the function space is non-empty.

```
∃→g_thm =
   ⊢ ∀ s t• (∃ x• x ∈g t) ⇒ (∃ f• f ∈g s →g t)
```

HOL Constant

```
→g_closed : GS → BOOL
```

$$\forall s\bullet \rightarrow_{g\_}closed\ s \Leftrightarrow \forall d\ c\bullet\ d \in_g s \wedge c \in_g s \Rightarrow d \rightarrow_g c \in_g s$$

4.7 Functional Abstraction

Functional abstraction is defined as a new variable binding construct yeilding a functional set.

4.7.1 Abstraction

Because of the closeness to lambda abstraction λ_g is used as the name of a new binder for set theoretic functional abstraction.

SML

```
declare_binder "λg";
```

To define a functional set we need a HOL function over sets together with a set which is to be the domain of the function. Specification of the range is not needed. The binding therefore yields a function which maps sets to sets (maps the domain to the function).

The following definition is a placeholder, a more abstract definition might eventually be substituted. The function is defined as that subset of the cartesian product of the set s and its image under the function f which coincides with the graph of f over s.

HOL Constant

```
$λg: (GS → GS) → GS → GS
```

$$\forall f\ s\bullet\ \$\lambda_g\ f\ s = Sep\ (s \times_g (Imagep\ f\ s))\ (\lambda p\bullet\ snd\ p = f\ (fst\ p))$$

4.8 Application and Extensionality

In this section we define function application and show that functions are extensional.

4.8.1 Application

Application by juxtaposition cannot be overloaded and is used for application of HOL functions. Application of functional sets is therefore defined as an infix operator whose name is the empty name subscripted by "g".

SML

$$declare\_infix\ (250,"_g");$$

The particular form shown here is innovative in the value specified for applications of functions to values outside their domain. The merit of the particular value chosen is that it makes true an extensionality theorem which quantifies over all sets as arguments to the function, which might not otherwise be the case. Whether this form is useful I don't know. Generally a result with fewer conditionals is harder to prove but easier to use, but in this case I'm not so sure of the benefit.

It may be noted that it may also be used to apply a non-functional relation, if what you want it some arbitrary value (selected by the choice function) to which some object relates.

HOL Constant

$$\$_g : GS \to GS \to GS$$

$$\forall f\ x\bullet f\ _g\ x = \\ \quad if\ \exists y\bullet x \mapsto_g y \in_g f \\ \quad then\ \epsilon y\bullet x \mapsto_g y \in_g f \\ \quad else\ f$$

$$app\_thm1 = \\ \quad \vdash \forall f\ x\bullet (\exists_1 y\bullet x \mapsto_g y \in_g f) \\ \quad\quad \Rightarrow x \mapsto_g (f\ _g\ x) \in_g f$$

$$app\_thm2 = \\ \quad \vdash \forall f\ x\ y\bullet fun\ f \wedge (x \mapsto_g y \in_g f) \\ \quad\quad \Rightarrow f\ _g\ x = y$$

$$app\_thm3 = \\ \quad \vdash \forall f\ x\bullet fun\ f \wedge x \in_g dom\ f \\ \quad\quad \Rightarrow x \mapsto_g f\ _g\ x \in_g f$$

$$o_{g\text{-}g}\_thm = \\ \quad \vdash \forall f\ g\ x\bullet fun\ f \wedge fun\ g \wedge x \in_g dom\ g \wedge ran\ g \subseteq_g dom\ f \\ \quad\quad \Rightarrow (f\ o_g\ g)\ _g\ x = f\ _g\ g\ _g\ x$$

4.8.2 The "Type" of an Application (1)

The following theorem states that the result of applying a partial function to a value in its domain is a value in its codomain.

SML

```
set_goal([],
        ⌜∀f s t u• f ∈g s ↛g t ∧
        u ∈g dom f ⇒
        f g u ∈g t⌝);
a (prove_tac[
        get_spec ⌜$↛g⌝,
        get_spec ⌜dom⌝]);
a (all_fc_tac [app_thm2] THEN asm_rewrite_tac[]);
a (all_fc_tac [f↦gs_thm1]);
a (all_fc_tac [∈g↔g_thm]);
a (POP_ASM_T ante_tac THEN asm_rewrite_tac []);
val g∈g_thm = save_pop_thm "g∈g_thm";
```

4.8.3 The "Type" of an Application (2)

The following theorem states that the result of applying a total function to a value in its domain is a value in its codomain.

4.8.4 Partial functions are total

Every partial function is total over its domain. (there is an ambiguity in the use of the term "domain" for a partial function. It might mean the left hand operand of some partial function space construction within which the partial function concerned may be found, or it might mean the set of values over which the function is defined. Here we are saying that if f is a partial function over A, then its domain is some subset of A and f is a total function over that subset of A.)

```
∈g↛g⇒∈g→g_thm =
        ⊢ ∀f s t u• f ∈g s ↛g t ⇒ f ∈g dom f →g t
```

4.9 The Identity Function

4.9.1 specification

HOL Constant

```
id : GS → GS
```

```
∀s• id s = Sep
        (s ×g s)
        λx• fst x = snd x
```

4.9.2 lemmas

$$
\begin{array}{l}
id\_thm1 = \\
\quad \vdash \forall s\ x \bullet\ x \in_g id\ s \\
\quad\quad \Leftrightarrow \exists y \bullet\ y \in_g s \wedge x = y \mapsto_g y
\end{array}
$$

$$
\begin{array}{l}
id\_ap\_thm = \\
\quad \vdash \forall s\ x \bullet\ x \in_g s \\
\quad\quad \Rightarrow (id\ s)\ _g\ x = x
\end{array}
$$

$$
\begin{array}{l}
id \in_g \twoheadrightarrow_g \_thm1 = \\
\quad \vdash \forall s\ t\ u \bullet\ s \subseteq_g t \cap_g u \\
\quad\quad \Rightarrow id\ s \in_g t \twoheadrightarrow_g u
\end{array}
$$

$$
\begin{array}{l}
id \in_g \twoheadrightarrow_g \_thm2 = \\
\quad \vdash \forall s\ t\ u \bullet\ s \subseteq_g t \\
\quad\quad \Rightarrow id\ s \in_g t \twoheadrightarrow_g t
\end{array}
$$

$$
\begin{array}{l}
id\_clauses = \\
\quad \vdash \forall s \bullet\ rel(id\ s) \wedge fun\ (id\ s) \\
\quad\quad \wedge\ dom(id\ s) = s \wedge ran(id\ s) = s
\end{array}
$$

4.10 Override

Override is an operator over sets which is intended primarily for use with functions. It may be used to change the value of the function over any part of its domain by overriding it with a function which is defined only for those values.

SML

$$declare\_infix\ (250,"\oplus_g");$$

HOL Constant

$$\$\oplus_{\boldsymbol{g}} : GS \to GS \to GS$$

$$
\begin{array}{l}
\forall s\ t \bullet\ s \oplus_g t = Sep\ (s \cup_g t) \\
\quad\quad \lambda x \bullet\ if\ fst\ x \in_g dom\ t\ then\ x \in_g t\ else\ x \in_g s
\end{array}
$$

$$
\begin{array}{l}
\in_{\boldsymbol{g}} \oplus_{\boldsymbol{g}} \boldsymbol{\_thm} = \\
\quad \vdash \forall\ s\ t\ x \bullet\ x \in_g s \oplus_g t = (if\ fst\ x \in_g dom\ t\ then\ x \in_g t\ else\ x \in_g s)
\end{array}
$$

$$
\begin{array}{l}
\mapsto_{\boldsymbol{g}} \in_{\boldsymbol{g}} \oplus_{\boldsymbol{g}} \boldsymbol{\_thm} = \\
\quad \vdash \forall\ s\ t\ x\ y \\
\quad\quad \bullet\ x \mapsto_g y \in_g s \oplus_g t = (x \mapsto_g y \in_g t \vee \neg\ x \in_g dom\ t \wedge x \mapsto_g y \in_g s)
\end{array}
$$

$$
\begin{array}{l}
\oplus_{\boldsymbol{g}} \boldsymbol{\_rel\_thm} = \\
\quad \vdash \forall\ s\ t \bullet\ rel\ s \wedge rel\ t \Rightarrow rel\ (s \oplus_g t)
\end{array}
$$

```
|
|⊕_g_fun_thm =
|   ⊢ ∀ s t• fun s ∧ fun t ⇒ fun (s ⊕_g t)
```

4.11 Proof Contexts

Finalisation of a proof context.

4.11.1 Proof Context

SML

```
|add_pc_thms "'gst-fun" ([
|       field_∅_g_thm,
|       fun_∅_g_thm,
|       ∅_g∈_g↦_g_thm]);
|set_merge_pcs ["basic_hol", "'gst-ax", "'gst-fun"];
|commit_pc "'gst-fun";
```

5 Ordinals

A new "gst-ord" theory is created as a child of "gst-ax". The theory will contain the definitions of ordinals and related material for general use, roughly following "Set Theory" by Frank Drake, chapter 2 section 2. The subsections in this document correspondend to the subsections in the book.

5.0.2 Motivation

This is really motivated purely by interest and self-education. Since its so fundamental I think it likely to turn out handy. Some of the material required is not specific to set theory and is quite widely applicable (in which case I actually develop it elsewhere and then just use it here. Well-foundedness and induction over well-founded relations is the obvious case relevant to this part of Drake. The recursion theorem is the important more general result which appears in the next section in Drake. "more general" means "can be developed as a polymorphic theory in HOL and applied outside the context of set theory". In fact these things have to be developed in the more general context to be used in the ways they are required in the development of set theory, since, for example, one wants to do definitions by recursion over the set membership relation where neither the function defined nor the relevant well-founded relation are actually sets.

5.0.3 Divergence

I have not followed Drake slavishly. More or less, I follow him where it works out OK and looks reasonable and doesn't trigger any of my prejudices.

Sometimes the context in which I am doing the work makes some divergence desirable or necessary. For example, I am doing the work in the context of a slightly eccentric set theory ("Galactic Set Theory") which mainly makes no difference, but has a non-standard formulation of the axiom of foundation. Mainly this is covered by deriving the standard formulation and its consequences and using them where this is used by Drake (in proving the trichotomy theorem). However, the machinery

for dealing with well-foundedness makes a difference to how induction principles are best formulated and derived.

Sometimes I look at what he has done and I think, "no way am I going to do that". Not necessarily big things, for example, I couldn't use his definition of successor ordinal which he pretty much admits himself is what we nerds call a kludge.

5.0.4 The Theory ord

The new theory is first created, together with a proof context which we will build up as we develop the theory.

SML

```
open_theory "gst-ax";
force_new_theory "gst-ord";
(* new_parent "wf_recp"; *)
force_new_pc "'gst-ord";
merge_pcs ["'savedthm_cs_∃_proof"] "'gst-ord";
set_merge_pcs ["basic_hol", "'gst-ax", "'gst-ord"];
```

5.1 Definitions 2.1 and 2.3

An ordinal is defined as a transitive and connected set. The usual ordering over the ordinals is defined and also the successor function.

5.1.1 The Definition

The concept of transitive set has already been defined in theory *gst-ax*. The concepts *connected* and *ordinal* are now defined.

HOL Constant

$$\boldsymbol{connected} : GS \to BOOL$$

$$\forall s : GS \bullet\ connected\ s \Leftrightarrow \forall t\ u : GS \bullet\ t \in_g s \wedge u \in_g s \Rightarrow t \in_g u \vee t = u \vee u \in_g t$$

HOL Constant

$$\boldsymbol{ordinal} : GS \to BOOL$$

$$\forall s : GS \bullet\ ordinal\ s \Leftrightarrow transitive\ s \wedge connected\ s$$

We now introduce infix ordering relations over ordinals.

SML

```
declare_infix(240,"<o");
declare_infix(240,"≤o");
```

HOL Constant

$\$<_o : GS \to GS \to BOOL$

$\forall x\ y{:}GS\bullet\ x <_o y \Leftrightarrow ordinal\ x \wedge ordinal\ y \wedge x \in_g y$

less_mem_thm =

$\vdash \forall\ \alpha\ \beta\bullet\ \alpha <_o \beta \Rightarrow ordinal\ \alpha \wedge ordinal\ \beta \wedge \alpha \in_g \beta$

mem_less_thm =

$\vdash \forall\ \alpha\ \beta\bullet\ ordinal\ \alpha \wedge ordinal\ \beta \wedge \alpha \in_g \beta \Rightarrow \alpha <_o \beta$

ord_mem_psub_thm =

$\vdash \forall\ \alpha\bullet\ ordinal\ \alpha \Rightarrow (\forall\ \beta\bullet\ \beta \in_g \alpha \Rightarrow \beta \subset_g \alpha)$

lto_psub_thm =

$\vdash \forall\ \alpha\ \beta\bullet\ \alpha <_o \beta \Rightarrow \alpha \subset_g \beta$

lo_trans_thm =

$\vdash \forall\ \alpha\ \beta\ \gamma\bullet\ \alpha <_o \beta \wedge \beta <_o \gamma \Rightarrow \alpha <_o \gamma$

HOL Constant

$\$\leqslant_o : GS \to GS \to BOOL$

$\forall x\ y{:}GS\bullet\ x \leqslant_o y \Leftrightarrow ordinal\ x \wedge ordinal\ y \wedge (x \in_g y \vee x = y)$

leo_lo_thm =

$\vdash \forall\ x\ y\bullet\ x \leqslant_o y \Leftrightarrow ordinal\ x \wedge ordinal\ y \wedge (x <_o y \vee x = y)$

leo_sub_thm =

$\vdash \forall\ \alpha\ \beta\bullet\ \alpha \leqslant_o \beta \Rightarrow \alpha \subseteq_g \beta$

leo_trans_thm =

$\vdash \forall\ \alpha\ \beta\ \gamma\bullet\ \alpha \leqslant_o \beta \wedge \beta \leqslant_o \gamma \Rightarrow \alpha \leqslant_o \gamma$

leo_lo_trans_thm =

$\vdash \forall\ \alpha\ \beta\ \gamma\bullet\ \alpha \leqslant_o \beta \wedge \beta <_o \gamma \Rightarrow \alpha <_o \gamma$

lo_leo_trans_thm =

$\vdash \forall\ \alpha\ \beta\ \gamma\bullet\ \alpha <_o \beta \wedge \beta \leqslant_o \gamma \Rightarrow \alpha <_o \gamma$

The following definition gives the successor function over the ordinals (this appears later in Drake).

HOL Constant

$\boldsymbol{suc_o} : GS \to GS$

$\forall x{:}GS\bullet\ suc_o\ x = x \cup_g (Sing\ x)$

5.2 Theorem 2.2

We prove that the empty set is an ordinal, and that the members of an ordinal and the successor of an ordinal are ordinals.

5.2.1 The Empty Set is an Ordinal

First we prove that the empty set is an ordinal, which requires only rewriting with the relevant definitions.

5.2.2 The Successor of an Ordinal is an Ordinal

Next we prove that the successor of an ordinal is an ordinal. This is done in two parts, transitivity and connectedness.

```
SML
set_goal([], ⌜     ∀ x:GS• transitive x ⇒ transitive (suc_o x)     ⌝);
```

```
SML
set_goal([],⌜∀ x:GS•
      connected x ⇒ connected (suc_o x)
⌝);
```

These together enable us to prove:

```
SML
set_goal([], ⌜∀ x:GS• ordinal x ⇒ ordinal (suc_o x)⌝);
```

The proof expands using the definition of ordinal, strips the goal and reasons forward from the resulting assumptions using the two lemmas proved above.

```
SML
a (rewrite_tac[get_spec ⌜ordinal⌝]
      THEN REPEAT strip_tac
      THEN fc_tac [trans_suc_trans, conn_suc_conn]);
val ord_suc_ord_thm = save_pop_thm "ord_suc_ord_thm";
```

5.2.3 The Ordinal Zero is not a successor

```
∅_g_not_suc_o_thm =
      ⊢ ¬ (∃ α• suc_o α = ∅_g)

not_in_suco_thm =
      ⊢ ∀ α• ¬ α = suc_o α

leo_suc_thm =
      ⊢ ∀ α• ordinal α ⇒ α ≤_o suc_o α

lo_suc_thm =
      ⊢ ∀ α• ordinal α ⇒ α <_o suc_o α
```

5.2.4 The members of an Ordinal are Ordinals

We now aim to prove that the members of an ordinal are ordinals. We do this by proving first that they are connected and then that they are transitive. First however, we show that any subset of a connected set is connected.

SML
```
set_goal([],⌜
      ∀ x:GS• connected x ⇒ ∀ y:GS• y ⊆g x ⇒ connected y
⌝);
```

The proof consists of expanding appropriate definitions, stripping the goal and then reasoning forward from the assumptions.

SML
```
a (rewrite_tac (map get_spec [⌜connected⌝, ⌜$⊆g⌝])
      THEN REPEAT_N 7 strip_tac);
```

```
(* *** Goal "" *** *)

(*  4 *) ⌜∀ t u• t ∈g x ∧ u ∈g x ⇒ t ∈g u ∨ t = u ∨ u ∈g t⌝
(*  3 *) ⌜∀ e• e ∈g y ⇒ e ∈g x⌝
(*  2 *) ⌜t ∈g y⌝
(*  1 *) ⌜u ∈g y⌝

(* ?⊢ *) ⌜t ∈g u ∨ t = u ∨ u ∈g t⌝
```

SML
```
a (all_asm_fc_tac[]);
a (REPEAT_N 2 (asm_fc_tac[]) THEN REPEAT strip_tac);
val conn_sub_conn = save_pop_thm "conn_sub_conn";
```

Now we show that any member of an ordinal is an ordinal.

SML
```
set_goal([],⌜
      ∀ x:GS• ordinal x ⇒ ∀ y:GS• y ∈g x ⇒ connected y
⌝);
```

Expanding the definition of ordinal and making use of transitivity enables us to infer that members of an ordinals are subsets and permits application of the previous result to obtain connectedness.

SML
```
a (rewrite_tac (map get_spec [⌜ordinal⌝, ⌜transitive⌝])
      THEN REPEAT strip_tac);
a (all_asm_fc_tac []);
a (all_asm_fc_tac [conn_sub_conn]);
val conn_mem_ord = save_pop_thm "conn_mem_ord";
```

To prove that the members of an ordinal are transitive, well-foundedness is needed. Now we are ready to prove that the members of an ordinal are transitive.

SML
$|set\_goal([], \ulcorner \forall\ x{:}GS\bullet\ ordinal\ x \Rightarrow \forall\ y{:}GS\bullet\ y \in_g x \Rightarrow transitive\ y\urcorner);$

Finally we prove that all members of an ordinal are ordinals.

SML
$|set\_goal([], \ulcorner \forall\ x{:}GS\bullet\ ordinal\ x \Rightarrow \forall\ y{:}GS\bullet\ y \in_g x \Rightarrow ordinal\ y\urcorner);$

5.2.5 Galaxies are Closed under suc

$|GCloseSuc = \vdash \forall g\bullet\ galaxy\ g \Rightarrow \forall x\bullet\ x \in_g g \Rightarrow suc_o\ x \in_g g$

5.3 Theorem 2.4

We prove that the ordinals are linearly ordered by $<_o$.

5.3.1

First we prove some lemmas:

SML
$|set\_goal([], \ulcorner \forall\ x\ y{:}GS\bullet\ transitive\ x \wedge transitive\ y \Rightarrow transitive\ (x \cap_g y)\urcorner);$

SML
$|set\_goal([], \ulcorner \forall\ x\ y{:}GS\bullet\ transitive\ x \wedge transitive\ y \Rightarrow transitive\ (x \cup_g y)\urcorner);$

SML
$|set\_goal([], \ulcorner \forall\ x\ y{:}GS\bullet\ connected\ x \wedge connected\ y \Rightarrow connected\ (x \cap_g y)\urcorner);$

SML
$|set\_goal([], \ulcorner \forall\ x\ y{:}GS\bullet\ ordinal\ x \wedge ordinal\ y \Rightarrow ordinal\ (x \cap_g y)\urcorner);$

SML
$|set\_goal([], \ulcorner \forall\ x\ y{:}GS\bullet\ ordinal\ x \wedge ordinal\ y \wedge x \subseteq_g y \wedge \neg\ x = y \Rightarrow x \in_g y\urcorner);$

$|\boldsymbol{trich\_for\_ords\_thm} =$
$|\quad \vdash \forall\ x\ y\bullet\ ordinal\ x \wedge ordinal\ y \Rightarrow x <_o y \vee x = y \vee y <_o x$
$|$
$|\boldsymbol{sub\_leo\_thm} =$
$|\quad \vdash \forall\ x\ y\bullet\ ordinal\ x \wedge ordinal\ y \Rightarrow (x \subseteq_g y \Leftrightarrow x \leqslant_o y)$
$|$
$|\boldsymbol{sub\_leo\_thm1} =$
$|\quad \vdash \forall\ x\ y\bullet\ ordinal\ x \wedge ordinal\ y \wedge x \subseteq_g y \Rightarrow x \leqslant_o y$

5.4 Definition 2.6

Successor and limit ordinals are defined. Natural numbers are defined.

5.4.1

These definitions are not the ones used by Drake, and not only the names but the concepts differ. My successor predicate does not hold of the empty set. I use the name "natural number" where he talks of integers, and generally I'm chosing longer names.

HOL Constant

$$\boldsymbol{successor} : GS \to BOOL$$

$$\forall s : GS \bullet\ successor\ s \Leftrightarrow \exists t \bullet\ ordinal\ t \wedge s = suc_o\ t$$

HOL Constant

$$\boldsymbol{limit\_ordinal} : GS \to BOOL$$

$$\forall s : GS \bullet\ limit\_ordinal\ s \Leftrightarrow ordinal\ s \wedge \neg\ successor\ s \wedge \neg\ s = \varnothing_g$$

5.5 Theorem 2.7

Induction theorems over ordinals.

5.5.1 Successors are Ordinals

SML

```
set_goal([],⌜ ∀ x:GS• successor x ⇒ ordinal x ⌝);
a (rewrite_tac[get_spec ⌜successor⌝]
      THEN REPEAT strip_tac
      THEN fc_tac [ord_suc_ord_thm]
      THEN asm_rewrite_tac[]);
val successor_ord_thm = save_pop_thm "successor_ord_thm";
```

5.5.2 Well-foundedness of the ordinals

First we prove that $<_o$ is well-founded.

5.5.3 An Ordinal is Zero, a successor or a limit

$$\boldsymbol{ordinal\_kind\_thm} =$$
$$\forall n \bullet\ ordinal\ n \Rightarrow n = \varnothing_g \vee successor\ n \vee limit\_ordinal\ n$$

5.6 Supremum and Strict Supremum

The supremum of a set of ordinals is the smallest ordinal greater than or equal to every ordinal in the set. With the Von Neumann representation of ordinals this is just the union of the set of ordinals.

SML

```
declare_infix (200, "ub");
declare_infix (200, "sub");
```

HOL Constant

$$\$\boldsymbol{ub} : GS \to GS \to BOOL$$

$$\forall \alpha\ \beta \bullet\ \alpha\ ub\ \beta \Leftrightarrow \forall \gamma \bullet\ \gamma \in_g \alpha \Rightarrow \gamma \leqslant_o \beta$$

HOL Constant

$$\boldsymbol{sup} : GS \to GS$$

$$\forall \alpha \bullet\ sup\ \alpha = \bigcup_g \alpha$$

$$\boldsymbol{ordinal\_limit\_thm} =$$
$$\vdash\ \forall\ \alpha \bullet\ (\forall\ \beta \bullet\ \beta \in_g \alpha \Rightarrow ordinal\ \beta) \Rightarrow ordinal\ (\bigcup_g \alpha)$$

$$\boldsymbol{sup\_lub\_thm} =$$
$$\vdash\ \forall\ \alpha \bullet\quad (\forall\ \beta \bullet\ \beta \in_g \alpha \Rightarrow ordinal\ \beta)$$
$$\Rightarrow \alpha\ ub\ sup\ \alpha$$
$$\wedge\ (\forall\ \gamma \bullet\ ordinal\ \gamma \wedge \alpha\ ub\ \gamma \Rightarrow sup\ \alpha \leqslant_o \gamma)$$

The operand here is intended to be an arbitrary set of ordinals and the result is the smallest ordinal strictly greater than any in the set.

HOL Constant

$$\$\boldsymbol{sub} : GS \to GS \to BOOL$$

$$\forall \alpha\ \beta \bullet\ \alpha\ sub\ \beta \Leftrightarrow \forall \gamma \bullet\ \gamma \in_g \alpha \Rightarrow \gamma <_o \beta$$

HOL Constant

$$\boldsymbol{ssup} : GS \to GS$$

$$\forall \alpha \bullet\ ssup\ \alpha = \bigcup_g (Imagep\ suc_o\ \alpha)$$

$$\boldsymbol{ordinal\_ssup\_thm} =$$
$$\vdash\ \forall\ \alpha \bullet\ (\forall\ \beta \bullet\ \beta \in_g \alpha \Rightarrow ordinal\ \beta) \Rightarrow ordinal\ (ssup\ \alpha)$$

5.7 Rank

We define the rank of a set.

5.7.1 The Consistency Proof

Before introducing the definition of rank we undertake the proof necessary to establish that the definition is conservative. The key lemma in this proof is the proof that the relevant functional "respects" the membership relation.

respect_lemma =
$\vdash (\lambda\ f\ x\bullet \bigcup_g\ (Imagep\ (suc_o\ o\ f)\ x))\ respects\ \$\in_g$

Armed with that lemma we can now prove that the function which we will call "rank" exists.

HOL Constant
rank : $GS \to GS$

$\forall x\bullet\ rank\ x = \bigcup_g\ (Imagep\ (suc_o\ o\ rank)\ x)$

5.8 Ordinal Arithmetic

5.8.1 Addition

The following lemma is used to demonstrate well-foundedness of the definition of ordinal addition:

plus_o_respect_lemma =
$\vdash\ \forall\ x\bullet\ (\lambda\ x\_+\ y\bullet\ if\ y = \varnothing_g\ then\ x\ else\ ssup\ (Imagep\ x\_+\ y))\ respects\ \$\in_g$

HOL Constant
$\$+_{\boldsymbol{o}} : GS \to GS \to GS$

$\forall\alpha\ \beta\bullet\ \alpha +_o \beta = if\ \beta = \varnothing_g\ then\ \alpha\ else\ ssup\ (Imagep\ (\$+_o\ \alpha)\ \beta)$

ord_plus_thm =
$\vdash\ \forall\ \alpha\ \beta\bullet\ ordinal\ \alpha\ \wedge\ ordinal\ \beta \Rightarrow ordinal\ (\alpha +_o \beta)$

5.8.2 Subtraction

The following definition is of reverse subtraction, i.e. the right operand is subtracted from the left and is taken from the left of that operand so that the following lemma (as yet unproven) obtains:

--_o_lemma =
$\forall\alpha\ \beta\bullet\ \alpha \leqslant_o \beta \Rightarrow \alpha +_o (\beta --_o \alpha) = \beta$

HOL Constant
$\$--_{\boldsymbol{o}} : GS \to GS \to GS$

T

5.9 Proof Context

In this section we define a proof context for ordinals.

5.9.1 Proof Context

SML

```
add_pc_thms "'gst-ord" ([]);
set_merge_pcs ["basic_hol", "'gst-ax", "'gst-ord"];
commit_pc "'gst-ord";
```

6 Natural Numbers

SML

```
open_theory "gst-ord";
force_new_theory "gst-nat";
force_new_pc "'gst-nat";
merge_pcs ["'savedthm_cs_∃_proof"] "'gst-nat";
set_merge_pcs ["basic_hol", "'gst-ax", "'gst-ord", "'gst-nat"];
```

HOL Constant

$$\boldsymbol{natural\_number} : GS \to BOOL$$

$$\forall s : GS \bullet natural\_number\ s \Leftrightarrow s = \varnothing_g \vee (successor\ s \wedge \forall t \bullet t \in_g s \Rightarrow t = \varnothing_g \vee successor\ t)$$

6.0.2 Ordering the Natural Numbers

To get an induction principle for the natural numbers we first define a well-founded ordering over them. Since I don't plan to use this a lot I use the name $<_{gn}$ (less than over the natural numbers defined in galactic set theory).

SML

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

HOL Constant

$$\$<_{\boldsymbol{gn}} : GS \to GS \to BOOL$$

$$\forall x\ y : GS \bullet x <_{gn} y \Leftrightarrow natural\_number\ x \wedge natural\_number\ y \wedge x \in_g y$$

Now we try to find a better proof that the one above that this is well-founded. And fail, this is just a more compact rendition of the same proof.

SML

```
set_goal([],⌜well_founded $<gn⌝);
a (asm_tac gs_wf_thm1);
a (fc_tac [wf_restrict_wf_thm]);
a (SPEC_NTH_ASM_T 1 ⌜λx y• natural_number x ∧ natural_number y⌝ ante_tac
      THEN rewrite_tac[]);
a (lemma_tac ⌜$<gn = (λ x y• (natural_number x ∧ natural_number y) ∧ x ∈g y)⌝
      THEN1 (REPEAT_N 2 (once_rewrite_tac [ext_thm])
            THEN prove_tac[get_spec ⌜$<gn⌝]));
a (asm_rewrite_tac[]);
val wf_nat_thm = save_pop_thm "wf_nat_thm";
```

This allows us to do well-founded induction over the natural number which the way I have implemented it is "course-of-values" induction. However, for the sake of form I will prove that induction principle as an explicit theorem. This is just what you get by expanding the definition of well-foundedness in the above theorem.

SML

```
val nat_induct_thm = save_thm ("nat_induct_thm",
      (rewrite_rule [get_spec ⌜well_founded⌝] wf_nat_thm));
```

Note that this theorem can only be used to prove properties which are true of all sets, so you have to make it conditional (*natural_number n ⇒ whatever*)I suppose I'd better do another one.

SML

```
set_goal([], ⌜∀ p• (∀ x• natural_number x ∧ (∀ y• y <gn x ⇒ p y) ⇒ p x)
      ⇒ (∀ x• natural_number x ⇒ p x)⌝);
a (asm_tac (rewrite_rule []
      (all_∀_intro (∀_elim ⌜λx• natural_number x ⇒ p x⌝ nat_induct_thm))));
a (rewrite_tac [all_∀_intro (taut_rule ⌜a ∧ b ⇒ c ⇔ b ⇒ a ⇒ c⌝)]);
a (lemma_tac ⌜∀ p x• (∀ y• y <gn x ⇒ p y)
      ⇔ (∀ y• y <gn x ⇒ natural_number y ⇒ p y)⌝);
(* *** Goal "1" *** *)
a (rewrite_tac [get_spec ⌜$<gn⌝]);
a (REPEAT strip_tac THEN all_asm_fc_tac[]);
(* *** Goal "2" *** *)
a (asm_rewrite_tac[]);
val nat_induct_thm2 = save_pop_thm "nat_induct_thm2";
```

I've tried using that principle and it too has disadvantages. Because $<_{gn}$ is used the induction hypothesis is more awkward to use (weaker) than it would have been if $\in_g$ had been used. Unfortunately the proof of an induction theorem using plain set membership is not entirely trivial, so its proof has to be left til later.

SML

```
set_goal([], ⌜∀ p• (∀ x• natural_number x ∧ (∀ y• y ∈g x ⇒ p y) ⇒ p x)
      ⇒ (∀ x• natural_number x ⇒ p x)⌝);
```

6.1 Theorem 2.8

The set of natural numbers.

6.1.1 Natural Numbers are Ordinals

SML

```
set_goal ([], ⌜∀n• natural_number n ⇒ ordinal n⌝);
a (rewrite_tac [get_spec ⌜natural_number⌝, get_spec ⌜successor⌝]);
a (REPEAT strip_tac THEN_TRY asm_rewrite_tac[ordinal_∅g]);
a (all_fc_tac [ord_suc_ord_thm]);
val ord_nat_thm = save_pop_thm "ord_nat_thm";
```

6.1.2 Members of Natural Numbers are Ordinals

SML

```
set_goal ([], ⌜∀n• natural_number n ⇒ ∀m• m ∈g n ⇒ ordinal m⌝);
a (REPEAT strip_tac);
a (REPEAT (all_fc_tac[ord_nat_thm, ord_mem_ord]));
val mem_nat_ord_thm = save_pop_thm "mem_nat_ord_thm";
```

6.1.3 A Natural Number is not a Limit Ordinal

SML

```
set_goal ([], ⌜∀n• natural_number n ⇒ ¬ limit_ordinal n⌝);
a (rewrite_tac [get_spec ⌜limit_ordinal⌝, get_spec ⌜natural_number⌝]);
a (REPEAT strip_tac);
val nat_not_lim_thm = save_pop_thm "nat_not_lim_thm";
```

6.1.4 A Natural Number is zero or a successor

SML

```
set_goal ([], ⌜∀n• natural_number n ⇒ successor n ∨ n = ∅g⌝);
a (rewrite_tac [get_spec ⌜natural_number⌝]);
a (REPEAT strip_tac);
val nat_zero_or_suc_thm = save_pop_thm "nat_zero_or_suc_thm";
```

6.1.5 A Natural Number does not contain a Limit Ordinal

SML

```
set_goal ([], ⌜∀m n• natural_number n ∧ m ∈g n ⇒ ¬ limit_ordinal m⌝);
a (rewrite_tac [get_spec ⌜limit_ordinal⌝, get_spec ⌜natural_number⌝]);
a (REPEAT strip_tac);
(* *** Goal "1" *** *)
a (all_fc_tac [mem_not_empty_thm]);
(* *** Goal "2" *** *)
a (all_asm_fc_tac[]);
val mem_nat_not_lim_thm = save_pop_thm "mem_nat_not_lim_thm";
```

6.1.6 All Members of Natural Numbers are Natural Numbers

SML

```
set_goal ([], ⌜∀n• natural_number n ⇒ ∀m• m ∈g n ⇒ natural_number m⌝);
a (rewrite_tac [get_spec ⌜natural_number⌝]);
a (REPEAT strip_tac THEN_TRY all_asm_fc_tac [mem_not_empty_thm]);
a (lemma_tac ⌜transitive n⌝ THEN1
      (REPEAT (all_fc_tac [get_spec ⌜ordinal⌝, successor_ord_thm])));
a (lemma_tac ⌜t ∈g n⌝ THEN1 (EVERY [all_fc_tac [get_spec ⌜transitive⌝],
      POP_ASM_T ante_tac, rewrite_tac [gst_relext_clauses], asm_prove_tac[]]));
a (all_asm_fc_tac[]);
val mem_nat_nat_thm = save_pop_thm "mem_nat_nat_thm";
```

6.1.7 Natural Numbers are in the Smallest Galaxy

SML

```
set_goal ([], ⌜∀n• natural_number n ⇒ n ∈g Gx ∅g⌝);
a (strip_tac THEN gen_induction_tac1 nat_induct_thm2);
a (fc_tac [nat_zero_or_suc_thm]);
(* *** Goal "1" *** *)
a (fc_tac [get_spec ⌜successor⌝]);
a (lemma_tac ⌜t <gn n⌝
      THEN1 asm_rewrite_tac [get_spec ⌜$<gn⌝, get_spec ⌜suco⌝]);
(* *** Goal "1.1" *** *)
a (lemma_tac ⌜t ∈g n⌝
      THEN1 asm_rewrite_tac [get_spec ⌜suco⌝]);
a (all_fc_tac [mem_nat_nat_thm]);
(* *** Goal "1.2" *** *)
a (asm_tac (∀_elim ⌜∅g⌝ galaxy_Gx));
a (asm_rewrite_tac[]);
a (REPEAT (all_asm_fc_tac[GCloseSuc]));
(* *** Goal "2" *** *)
a (asm_rewrite_tac[]);
val nat_in_G∅g_thm = save_pop_thm "nat_in_G∅g_thm";
```

6.1.8 The Existence of w

This comes from the axiom of infinity, however, in galactic set theory we get that from the existence of galaxies, so the following proof is a little unusual.

SML

```
set_goal ([], ⌜∃w• ∀z• z ∈g w ⇔ natural_number z⌝);
a (∃_tac ⌜Sep (Gx ∅g) natural_number⌝
      THEN rewrite_tac [gst_opext_clauses]);
a (rewrite_tac [all_∀_intro (taut_rule ⌜(a ∧ b ⇔ b) ⇔ b ⇒ a⌝)]);
a strip_tac;
a (gen_induction_tac1 nat_induct_thm2);
a (fc_tac [nat_zero_or_suc_thm]);
```

```
(* *** Goal "1" *** *)
a (fc_tac [get_spec ⌜successor⌝, nat_in_G∅g_thm]);
(* *** Goal "2" *** *)
a (asm_rewrite_tac []);
val w_exists_thm = save_pop_thm "w_exists_thm";
```

6.2 Naming the Natural Numbers

It will be useful to be able to have names for the finite ordinals, which are used as tags in the syntax:

HOL Constant

```
Natg: ℕ → GS
```

```
    Natg 0 = ∅g
∧ ∀n• Natg (n+1) = suco (Natg n)
```

We will need to know that these are all distinct ordinals.

```
ord_nat_thm2 =
    ⊢ ∀ n• ordinal (Natg n)

not_suc_nat_zero_thm =
    ⊢ ∀ n• ¬ suco (Natg n) = ∅g

less_sum_thm =
    ⊢ ∀ x y• x ⩽ y ⇒ (∃ z• x + z = y)

natg_mono_thm =
    ⊢ ∀ x y• Natg x ⩽o Natg (x + y)

natg_one_one_thm =
    ⊢ ∀ x y• Natg x = Natg y ⇒ x = y

natg_one_one_thm2 =
    ⊢ ∀ x y• Natg x = Natg y ⇔ x = y
```

6.3 Proof Context

In this section we define a proof context for natural numbers.

6.3.1 Proof Context

SML

```
add_pc_thms "'gst-nat" ([natg_one_one_thm2]);
set_merge_pcs ["basic_hol", "'gst-ax", "'gst-ord", "'gst-nat"];
commit_pc "'gst-nat";
```

7 Closing

SML

```
open_theory "gst-fun";
force_new_theory "GS";
(* new_parent "gst-sumprod";
new_parent "gst-fixp"; *)
new_parent "gst-ord";
new_parent "gst-nat";
force_new_pc "GS";
force_new_pc "'GS1";
val rewrite_thms = ref ([]:THM list);

merge_pcs ["'gst-ax", "'gst-fun"(*, "'gst-sumprod", "'gst-fixp", "'gst-lists"*),"'gst-ord", "'gst-nat'
      "'GS1";
commit_pc "'GS1";
merge_pcs ["basic_hol", "'GS1"] "GS";
commit_pc "GS";
```

8 The Theory gst-ax

8.1 Parents

wf_recp *wf_relp* *U_orders* *rbjmisc*

8.2 Children

gst−ord *gst−fun*

8.3 Constants

$\$\in_{\boldsymbol{g}}$	$GS \to GS \to BOOL$
$\boldsymbol{X_g}$	$GS \to GS\ \mathbb{P}$
$\$\in_{\boldsymbol{g}}{}^{+}$	$GS \to GS \to BOOL$
$\$\subseteq_{\boldsymbol{g}}$	$GS \to GS \to BOOL$
$\$\subset_{\boldsymbol{g}}$	$GS \to GS \to BOOL$
$\subseteq_{\boldsymbol{g}}\textbf{-closed}$	$GS \to BOOL$
$\mathbb{P}_{\boldsymbol{g}}$	$GS \to GS$
$\bigcup_{\boldsymbol{g}}$	$GS \to GS$
$\boldsymbol{RelIm}$	$(GS \to GS \to BOOL) \to GS \to GS$
$\boldsymbol{Sep}$	$GS \to (GS \to BOOL) \to GS$
$\boldsymbol{galaxy}$	$GS \to BOOL$
$\boldsymbol{Gx}$	$GS \to GS$
$\boldsymbol{transitive}$	$GS \to BOOL$
$\varnothing_{\boldsymbol{g}}$	GS
$\boldsymbol{Imagep}$	$(GS \to GS) \to GS \to GS$
$\boldsymbol{Pair_g}$	$GS \to GS \to GS$
$\boldsymbol{Sing}$	$GS \to GS$
$\$\cup_{\boldsymbol{g}}$	$GS \to GS \to GS$
$\bigcap_{\boldsymbol{g}}$	$GS \to GS$
$\$\cap_{\boldsymbol{g}}$	$GS \to GS \to GS$

8.4 Types

GS

8.5 Fixity

Right Infix 230:

$\subseteq_{\boldsymbol{g}}$ $\in_{\boldsymbol{g}}$ $\in_{\boldsymbol{g}}{}^{+}$ $\subset_{\boldsymbol{g}}$

Right Infix 240:

$\cap_{\boldsymbol{g}}$ $\cup_{\boldsymbol{g}}$

8.6 Axioms

$$
\begin{array}{ll}
\boldsymbol{gs\_ext\_axiom} & \vdash \forall\ s\ t\bullet\ s = t \Leftrightarrow (\forall\ e\bullet\ e \in_g s \Leftrightarrow e \in_g t) \\
\boldsymbol{gs\_wf\_axiom} & \vdash \mathit{UWellFounded}\ \$\in_g \\
\boldsymbol{Ontology\_axiom} & \\
 & \vdash \forall\ s \\
 & \quad \bullet \exists\ g \\
 & \quad\quad \bullet\ s \in_g g \\
 & \quad\quad\quad \wedge\ (\forall\ t \\
 & \quad\quad\quad \bullet\ t \in_g g \\
 & \quad\quad\quad\quad \Rightarrow t \subseteq_g g \\
 & \quad\quad\quad\quad\quad \wedge\ (\exists\ p \\
 & \quad\quad\quad\quad\quad \bullet\ (\forall\ v\bullet\ v \in_g p \Leftrightarrow v \subseteq_g t) \wedge p \in_g g) \\
 & \quad\quad\quad\quad\quad \wedge\ (\exists\ u \\
 & \quad\quad\quad\quad\quad \bullet\ (\forall\ v \\
 & \quad\quad\quad\quad\quad\quad \bullet\ v \in_g u \Leftrightarrow (\exists\ w\bullet\ v \in_g w \wedge w \in_g t)) \\
 & \quad\quad\quad\quad\quad\quad \wedge\ u \in_g g) \\
 & \quad\quad\quad\quad\quad \wedge\ (\forall\ \mathit{rel} \\
 & \quad\quad\quad\quad\quad \bullet\ \mathit{ManyOne}\ \mathit{rel} \\
 & \quad\quad\quad\quad\quad\quad \Rightarrow (\exists\ r \\
 & \quad\quad\quad\quad\quad\quad \bullet\ (\forall\ v \\
 & \quad\quad\quad\quad\quad\quad\quad \bullet\ v \in_g r \\
 & \quad\quad\quad\quad\quad\quad\quad\quad \Leftrightarrow (\exists\ w\bullet\ w \in_g t \wedge \mathit{rel}\ w\ v)) \\
 & \quad\quad\quad\quad\quad\quad\quad \wedge\ (r \subseteq_g g \Rightarrow r \in_g g))))
\end{array}
$$

8.7 Definitions

$$
\begin{array}{ll}
\boldsymbol{X_g} & \vdash \forall\ s\bullet\ X_g\ s = \{t | t \in_g s\} \\
{\in_{\boldsymbol{g}}}^+ & \vdash \$\in_g{}^+ = tc\ \$\in_g \\
\subseteq_{\boldsymbol{g}} & \vdash \forall\ s\ t\bullet\ s \subseteq_g t \Leftrightarrow (\forall\ e\bullet\ e \in_g s \Rightarrow e \in_g t) \\
\subset_{\boldsymbol{g}} & \vdash \forall\ s\ t\bullet\ s \subset_g t \Leftrightarrow s \subseteq_g t \wedge \neg\ t \subseteq_g s \\
\subseteq_{\boldsymbol{g}}\boldsymbol{\text{-}closed} & \vdash \forall\ s \\
 & \quad \bullet \subseteq_g\text{-}\mathit{closed}\ s \Leftrightarrow (\forall\ e\ f\bullet\ e \in_g s \wedge f \subseteq_g e \Rightarrow f \in_g s) \\
\mathbb{P}_{\boldsymbol{g}} & \vdash \forall\ s\ t\bullet\ t \in_g \mathbb{P}_g\ s \Leftrightarrow t \subseteq_g s \\
\bigcup_{\boldsymbol{g}} & \vdash \forall\ s\ t\bullet\ t \in_g \bigcup_g s \Leftrightarrow (\exists\ e\bullet\ t \in_g e \wedge e \in_g s) \\
\boldsymbol{RelIm} & \vdash \forall\ \mathit{rel}\ s\ t \\
 & \quad \bullet\ \mathit{ManyOne}\ \mathit{rel} \\
 & \quad\quad \Rightarrow (t \in_g \mathit{RelIm}\ \mathit{rel}\ s \Leftrightarrow (\exists\ e\bullet\ e \in_g s \wedge \mathit{rel}\ e\ t)) \\
\boldsymbol{Sep} & \vdash \forall\ s\ p\ e\bullet\ e \in_g \mathit{Sep}\ s\ p \Leftrightarrow e \in_g s \wedge p\ e \\
\boldsymbol{galaxy} & \vdash \forall\ s \\
 & \quad \bullet\ \mathit{galaxy}\ s \\
 & \quad\quad \Leftrightarrow (\exists\ x\bullet\ x \in_g s) \\
 & \quad\quad\quad \wedge\ (\forall\ t \\
 & \quad\quad\quad \bullet\ t \in_g s \\
 & \quad\quad\quad\quad \Rightarrow t \subseteq_g s \\
 & \quad\quad\quad\quad\quad \wedge\ \mathbb{P}_g\ t \in_g s \\
 & \quad\quad\quad\quad\quad \wedge\ \bigcup_g t \in_g s \\
 & \quad\quad\quad\quad\quad \wedge\ (\forall\ \mathit{rel} \\
 & \quad\quad\quad\quad\quad \bullet\ \mathit{ManyOne}\ \mathit{rel} \\
 & \quad\quad\quad\quad\quad\quad \Rightarrow \mathit{RelIm}\ \mathit{rel}\ t \subseteq_g s \\
 & \quad\quad\quad\quad\quad\quad \Rightarrow \mathit{RelIm}\ \mathit{rel}\ t \in_g s))
\end{array}
$$

Gx $\vdash \forall\ s\ t$
$\quad\bullet\ t \in_g Gx\ s \Leftrightarrow (\forall\ g\bullet\ galaxy\ g \wedge s \in_g g \Rightarrow t \in_g g)$

transitive $\vdash \forall\ s\bullet\ transitive\ s \Leftrightarrow (\forall\ e\bullet\ e \in_g s \Rightarrow e \subseteq_g s)$

$\emptyset_g$ $\vdash \forall\ s\bullet \neg\ s \in_g \emptyset_g$

Imagep $\vdash \forall\ f\ s\ x\bullet\ x \in_g Imagep\ f\ s \Leftrightarrow (\exists\ e\bullet\ e \in_g s \wedge x = f\ e)$

$Pair_g$ $\vdash \forall\ s\ t\ e\bullet\ e \in_g Pair_g\ s\ t \Leftrightarrow e = s \vee e = t$

Sing $\vdash \forall\ s\bullet\ Sing\ s = Pair_g\ s\ s$

$\cup_g$ $\vdash \forall\ s\ t\ e\bullet\ e \in_g s \cup_g t \Leftrightarrow e \in_g s \vee e \in_g t$

$\bigcap_g$ $\vdash \forall\ s$
$\quad\bullet \bigcap_g s = Sep\ (\bigcup_g s)\ (\lambda\ x\bullet \forall\ t\bullet\ t \in_g s \Rightarrow x \in_g t)$

$\cap_g$ $\vdash \forall\ s\ t\bullet\ s \cap_g t = Sep\ s\ (\lambda\ x\bullet\ x \in_g t)$

8.8 Theorems

UWellFounded_well_founded_thm
$\vdash \forall\ \$<<\bullet\ UWellFounded\ \$<< \Leftrightarrow well\_founded\ \$<<$

gs_wf_thm1 $\vdash well\_founded\ \$\in_g$

gs_wftc_thm $\vdash well\_founded\ (tc\ \$\in_g)$

gs_wf_min_thm
$\vdash \forall\ x$
$\quad\bullet (\exists\ y\bullet\ y \in_g x)$
$\qquad \Rightarrow (\exists\ z\bullet\ z \in_g x \wedge \neg\ (\exists\ v\bullet\ v \in_g z \wedge v \in_g x))$

gs_wftc_thm2 $\vdash well\_founded\ \$\in_g{}^+$

tc∈_incr_thm $\vdash \forall\ x\ y\bullet\ x \in_g y \Rightarrow x \in_g{}^+ y$

tc∈_cases_thm
$\vdash \forall\ x\ y$
$\quad\bullet\ x \in_g{}^+ y \Leftrightarrow x \in_g y \vee (\exists\ z\bullet\ x \in_g{}^+ z \wedge z \in_g y)$

tc∈_trans_thm
$\vdash \forall\ s\ t\ u\bullet\ s \in_g{}^+ t \wedge t \in_g{}^+ u \Rightarrow s \in_g{}^+ u$

gs_wf_ind_thm
$\vdash \forall\ p\bullet\ (\forall\ x\bullet\ (\forall\ y\bullet\ y \in_g x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x\bullet\ p\ x)$

gs_cv_ind_thm
$\vdash \forall\ p$
$\quad\bullet\ (\forall\ x\bullet\ (\forall\ y\bullet\ tc\ \$\in_g\ y\ x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x\bullet\ p\ x)$

gs_cv_ind_thm2
$\vdash \forall\ p\bullet\ (\forall\ x\bullet\ (\forall\ y\bullet\ y \in_g{}^+ x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x\bullet\ p\ x)$

wf_l1 $\vdash \forall\ x\bullet \neg\ x \in_g x$

wf_l2 $\vdash \forall\ x\ y\bullet \neg\ (x \in_g y \wedge y \in_g x)$

wf_l3 $\vdash \forall\ x\ y\ z\bullet \neg\ (x \in_g y \wedge y \in_g z \wedge z \in_g x)$

⊆g_eq_thm $\vdash \forall\ A\ B\bullet\ A = B \Leftrightarrow A \subseteq_g B \wedge B \subseteq_g A$

⊆g_refl_thm $\vdash \forall\ A\bullet\ A \subseteq_g A$

∈g⊆g_thm $\vdash \forall\ e\ A\ B\bullet\ e \in_g A \wedge A \subseteq_g B \Rightarrow e \in_g B$

⊆g_trans_thm
$\vdash \forall\ A\ B\ C\bullet\ A \subseteq_g B \wedge B \subseteq_g C \Rightarrow A \subseteq_g C$

not_psub_thm $\vdash \forall\ x\bullet \neg\ x \subset_g x$

s∈ℙs_thm $\vdash \forall\ s\bullet\ s \in_g \mathbb{P}_g\ s$

stc∈ℙs_thm $\vdash \forall\ s\bullet\ s \in_g{}^+ \mathbb{P}_g\ s$

∈g⋃g_thm $\vdash \forall\ s\ t\bullet\ t \in_g s \Rightarrow t \subseteq_g \bigcup_g s$

Sep_sub_thm $\vdash \forall\ s\ p\bullet\ Sep\ s\ p \subseteq_g s$

Sep_sub_thm2 $\vdash \forall\ s\ p\ e\bullet\ e \in_g Sep\ s\ p \Rightarrow e \in_g s$

Sep_∈_ℙ_thm $\vdash \forall\ s\ p\bullet\ Sep\ s\ p \in_g \mathbb{P}_g\ s$

Sep_⊆_thm $\vdash \forall\ s\ t\bullet\ t \subseteq_g s \Rightarrow Sep\ s\ (CombC\ \$\in_g\ t) = t$

galaxies_∃_thm

$\vdash \forall\ s\bullet\ \exists\ g\bullet\ s \in_g g \wedge galaxy\ g$

t_in_Gx_t_thm

$\vdash \forall\ t\bullet\ t \in_g Gx\ t$

tc∈_Gx_thm $\vdash \forall\ t\bullet\ t \in_g{}^{+} Gx\ t$

Gx_⊆g_galaxy

$\vdash \forall\ s\ g\bullet\ galaxy\ g \wedge s \in_g g \Rightarrow Gx\ s \subseteq_g g$

galaxy_Gx $\vdash \forall\ s\bullet\ galaxy\ (Gx\ s)$

GalaxiesTransitive_thm

$\vdash \forall\ s\bullet\ galaxy\ s \Rightarrow transitive\ s$

GClose_fc_clauses

$\vdash \forall\ g$
$\bullet\ galaxy\ g$
$\Rightarrow (\forall\ s$
$\bullet\ s \in_g g$
$\Rightarrow \mathbb{P}_g\ s \in_g g$
$\wedge \bigcup_g\ s \in_g g$
$\wedge (\forall\ p\bullet\ Sep\ s\ p \in_g g)$
$\wedge (\forall\ t\bullet\ t \subseteq_g s \Rightarrow t \in_g g))$

GClose_tc∈g_thm

$\vdash \forall\ s\ g\bullet\ galaxy\ g \Rightarrow s \in_g{}^{+} g \Rightarrow s \in_g g$

Gx_mono_thm $\vdash \forall\ s\ t\bullet\ s \subseteq_g t \Rightarrow Gx\ s \subseteq_g Gx\ t$

Gx_mono_thm2 $\vdash \forall\ s\ t\bullet\ s \in_g t \Rightarrow Gx\ s \subseteq_g Gx\ t$

Gx_trans_thm $\vdash \forall\ s\bullet\ transitive\ (Gx\ s)$

t_sub_Gx_t_thm

$\vdash \forall\ t\bullet\ t \subseteq_g Gx\ t$

Gx_mono_thm3 $\vdash \forall\ s\ t\bullet\ s \subseteq_g t \Rightarrow s \subseteq_g Gx\ t$

Gx_mono_thm4 $\vdash \forall\ s\ t\bullet\ s \subseteq_g t \Rightarrow s \in_g Gx\ t$

Gx_trans_thm2

$\vdash \forall\ s\ t\bullet\ s \in_g t \Rightarrow s \in_g Gx\ t$

Gx_trans_thm3

$\vdash \forall\ s\ t\ u\bullet\ s \in_g t \wedge t \in_g Gx\ u \Rightarrow s \in_g Gx\ u$

G∅gC $\vdash \forall\ g\bullet\ galaxy\ g \Rightarrow \varnothing_g \in_g g$

∅g⊆g_thm $\vdash \forall\ s\bullet\ \varnothing_g \subseteq_g s$

⋃g∅g_thm $\vdash \bigcup_g \varnothing_g = \varnothing_g$

mem_not_empty_thm

$\vdash \forall\ m\ n\bullet\ m \in_g n \Rightarrow \neg\ n = \varnothing_g$

∅g_∈g_galaxy_thm

$\vdash \forall\ x\bullet\ galaxy\ x \Rightarrow \varnothing_g \in_g x$

∅g_∈g_Gx_thm

$\vdash \forall\ x\bullet\ \varnothing_g \in_g Gx\ x$

GImagepC $\vdash \forall\ g$
$\bullet\ galaxy\ g$
$\Rightarrow (\forall\ s$
$\bullet\ s \in_g g$
$\Rightarrow (\forall\ f\bullet\ Imagep\ f\ s \subseteq_g g \Rightarrow Imagep\ f\ s \in_g g))$

Pairg_∈_thm $\vdash \forall\ x\ y\bullet\ x \in_g Pair_g\ x\ y \wedge y \in_g Pair_g\ x\ y$

Pairg_tc∈_thm

$\vdash \forall\ s\ t\bullet\ s \in_g{}^{+} Pair_g\ s\ t \wedge t \in_g{}^{+} Pair_g\ s\ t$

Pairg_eq_thm

$\vdash \forall\ s\ t\ u\ v$
• $Pair_g\ s\ t = Pair_g\ u\ v$
$\Leftrightarrow s = u \wedge t = v \vee s = v \wedge t = u$

GClosePair$_g$ $\vdash \forall\ g$
• $galaxy\ g$
$\Rightarrow (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow Pair_g\ s\ t \in_g g)$

Sing_thm2 $\vdash \forall\ x\bullet\ x \in_g Sing\ x$

Sing_tc$\in$_thm $\vdash \forall\ x\bullet\ x \in_g{}^+ Sing\ x$

GCloseSing $\vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ s\bullet\ s \in_g g \Rightarrow Sing\ s \in_g g)$

$\subseteq_g\cup_g$_thm $\vdash \forall\ A\ B\bullet\ A \subseteq_g A \cup_g B \wedge B \subseteq_g A \cup_g B$

$\cup_g\subseteq_g$_thm1 $\vdash \forall\ A\ B\ C\bullet\ A \subseteq_g C \wedge B \subseteq_g C \Rightarrow A \cup_g B \subseteq_g C$

$\cup_g\subseteq_g$_thm2 $\vdash \forall\ A\ B\ C\ D\bullet\ A \subseteq_g C \wedge B \subseteq_g D \Rightarrow A \cup_g B \subseteq_g C \cup_g D$

$\cup_g\varnothing_g$_clauses
$\vdash \forall\ A\bullet\ A \cup_g \varnothing_g = A \wedge \varnothing_g \cup_g A = A$

GClose$\cup_g$ $\vdash \forall\ g$
• $galaxy\ g$
$\Rightarrow (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow s \cup_g t \in_g g)$

$\bigcap_g$_thm $\vdash \forall\ x\ s\ e$
• $x \in_g s \Rightarrow (e \in_g \bigcap_g s \Leftrightarrow (\forall\ y\bullet\ y \in_g s \Rightarrow e \in_g y))$

$\bigcap_g\subseteq_g$_thm $\vdash \forall\ s\ t\bullet\ s \in_g t \Rightarrow \bigcap_g t \subseteq_g s$

$\subseteq_g\bigcap_g$_thm $\vdash \forall\ A\ B$
• $A \in_g B$
$\Rightarrow (\forall\ C\bullet\ (\forall\ D\bullet\ D \in_g B \Rightarrow C \subseteq_g D) \Rightarrow C \subseteq_g \bigcap_g B)$

$\bigcap_g\varnothing_g$_thm $\vdash \bigcap_g \varnothing_g = \varnothing_g$

GClose$\bigcap_g$ $\vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ s\bullet\ s \in_g g \Rightarrow \bigcap_g s \in_g g)$

GClose$\cap_g$ $\vdash \forall\ g$
• $galaxy\ g$
$\Rightarrow (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow s \cap_g t \in_g g)$

$\cap_g$_thm $\vdash \forall\ s\ t\ e\bullet\ e \in_g s \cap_g t \Leftrightarrow e \in_g s \wedge e \in_g t$

$\subseteq_g\cap_g$_thm $\vdash \forall\ A\ B\bullet\ A \cap_g B \subseteq_g A \wedge A \cap_g B \subseteq_g B$

$\cap_g\subseteq_g$_thm1 $\vdash \forall\ A\ B\ C\bullet\ A \subseteq_g C \wedge B \subseteq_g C \Rightarrow A \cap_g B \subseteq_g C$

$\cap_g\subseteq_g$_thm2 $\vdash \forall\ A\ B\ C\ D\bullet\ A \subseteq_g C \wedge B \subseteq_g D \Rightarrow A \cap_g B \subseteq_g C \cap_g D$

$\cap_g\subseteq_g$_thm3 $\vdash \forall\ A\ B\ C\bullet\ C \subseteq_g A \wedge C \subseteq_g B \Rightarrow C \subseteq_g A \cap_g B$

not_x_in_x_thm
$\vdash \neg\ (\exists\ x\bullet\ x \in_g x)$

GClose_fc_clauses2 $\vdash \forall\ g$
• $galaxy\ g$
$\Rightarrow (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow Pair_g\ s\ t \in_g g)$
$\wedge\ (\forall\ s\bullet\ s \in_g g \Rightarrow Sing\ s \in_g g)$
$\wedge\ (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow s \cup_g t \in_g g)$
$\wedge\ (\forall\ s\bullet\ s \in_g g \Rightarrow \bigcap_g s \in_g g)$
$\wedge\ (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow s \cap_g t \in_g g)$

tc$\in$_clauses $\vdash \forall\ s$
• $s \in_g{}^+ Sing\ s$
$\wedge\ s \in_g{}^+ \mathbb{P}_g\ s$
$\wedge\ (\forall\ t\bullet\ t \in_g{}^+ Pair_g\ s\ t \wedge s \in_g{}^+ Pair_g\ s\ t)$

gst_opext_clauses
$\vdash \forall\ s\ t\ x\ f\ u\ v\ e\ p$
• $\neg\ s \in_g \varnothing_g$
$\wedge\ (t \in_g \mathbb{P}_g\ s \Leftrightarrow t \subseteq_g s)$

$$\wedge\ (t \in_g \bigcup_g s \Leftrightarrow (\exists\ e\bullet\ t \in_g e \wedge e \in_g s))$$
$$\wedge\ (x \in_g Imagep\ f\ s \Leftrightarrow (\exists\ e\bullet\ e \in_g s \wedge x = f\ e))$$
$$\wedge\ (Pair_g\ s\ t = Pair_g\ u\ v$$
$$\Leftrightarrow s = u \wedge t = v \vee s = v \wedge t = u)$$
$$\wedge\ (e \in_g Pair_g\ s\ t \Leftrightarrow e = s \vee e = t)$$
$$\wedge\ (Sing\ s = Sing\ t \Leftrightarrow s = t)$$
$$\wedge\ (e \in_g Sing\ s \Leftrightarrow e = s)$$
$$\wedge\ (Pair_g\ s\ t = Sing\ u \Leftrightarrow s = u \wedge t = u)$$
$$\wedge\ (Sing\ s = Pair_g\ t\ u \Leftrightarrow s = t \wedge s = u)$$
$$\wedge\ (e \in_g Sep\ s\ p \Leftrightarrow e \in_g s \wedge p\ e)$$
$$\wedge\ (e \in_g s \cup_g t \Leftrightarrow e \in_g s \vee e \in_g t)$$
$$\wedge\ (e \in_g s \cap_g t \Leftrightarrow e \in_g s \wedge e \in_g t)$$

gst_relext_clauses

$$\vdash \forall\ s\ t$$
$$\bullet\ (s = t \Leftrightarrow (\forall\ e\bullet\ e \in_g s \Leftrightarrow e \in_g t))$$
$$\wedge\ (s \subseteq_g t \Leftrightarrow (\forall\ e\bullet\ e \in_g s \Rightarrow e \in_g t))$$

9 The Theory gst-fun

9.1 Parents

$gst-ax$

9.2 Children

GS

9.3 Constants

$\$\mapsto_g$	$GS \to GS \to GS$
$\boldsymbol{snd}$	$GS \to GS$
$\boldsymbol{fst}$	$GS \to GS$
$\boldsymbol{MkPair_g}$	$GS \times GS \to GS$
$\boldsymbol{MkTriple_g}$	$GS \times GS \times GS \to GS$
$\boldsymbol{rel}$	$GS \to BOOL$
$\boldsymbol{dom}$	$GS \to GS$
$\boldsymbol{ran}$	$GS \to GS$
$\boldsymbol{field}$	$GS \to GS$
$\$\lhd_g$	$GS \to GS \to GS$
$\$\rhd_g$	$GS \to GS \to GS$
$\$\trianglelefteq_g$	$GS \to GS \to GS$
$\$\trianglerighteq_g$	$GS \to GS \to GS$
$\boldsymbol{Rel2DepType_g}$	$GS \to GS$
$\boldsymbol{DepSum_g}$	$GS \to GS$
$\$\boldsymbol{\Sigma}_g$	$(GS \to GS) \to GS \to GS$
$\$\times_g$	$GS \to GS \to GS$
$\$\leftrightarrow_g$	$GS \to GS \to GS$
$\$\boldsymbol{o}_g$	$GS \to GS \to GS$
$\boldsymbol{fun}$	$GS \to BOOL$
$\$\nrightarrow_g$	$GS \to GS \to GS$
$\$\to_g$	$GS \to GS \to GS$
$\boldsymbol{\to_g\text{-}closed}$	$GS \to BOOL$
$\$\boldsymbol{\lambda}_g$	$(GS \to GS) \to GS \to GS$
$\$_g$	$GS \to GS \to GS$
$\boldsymbol{id}$	$GS \to GS$
$\$\oplus_g$	$GS \to GS \to GS$

9.4 Aliases

$\lhd$	$\$\lhd_g : GS \to GS \to GS$
$\rhd$	$\$\rhd_g : GS \to GS \to GS$
$\trianglelefteq$	$\$\trianglelefteq_g : GS \to GS \to GS$
$\trianglerighteq$	$\$\trianglerighteq_g : GS \to GS \to GS$

9.5 Fixity

Binder: $\boldsymbol{\Sigma_g}$ $\boldsymbol{\lambda_g}$

Right Infix 240:

$\boldsymbol{\leftrightarrow_g}$ $\boldsymbol{\rightarrow_g}$ $\boldsymbol{\times_g}$ $\boldsymbol{\nrightarrow_g}$ $\boldsymbol{\mapsto_g}$

Right Infix 250:

$\boldsymbol{o_g}$ $\boldsymbol{;_g}$ $\boldsymbol{\oplus_g}$

Right Infix 300:

$\boldsymbol{\rhd_g}$ $\boldsymbol{\triangleright_g}$ $\boldsymbol{\lhd_g}$ $\boldsymbol{\triangleleft_g}$

9.6 Definitions

$\boldsymbol{\mapsto_g}$ $\vdash \forall\ s\ t \bullet\ s \mapsto_g t = Pair_g\ (Sing\ s)\ (Pair_g\ s\ t)$

$\boldsymbol{fst}$

$\boldsymbol{snd}$ $\vdash \forall\ s\ t \bullet\ fst\ (s \mapsto_g t) = s \wedge snd\ (s \mapsto_g t) = t$

$\boldsymbol{MkPair_g}$ $\vdash \forall\ lr \bullet\ MkPair_g\ lr = Fst\ lr \mapsto_g Snd\ lr$

$\boldsymbol{MkTriple_g}$ $\vdash \forall\ t \bullet\ MkTriple_g\ t = Fst\ t \mapsto_g MkPair_g\ (Snd\ t)$

$\boldsymbol{rel}$ $\vdash \forall\ x \bullet\ rel\ x \Leftrightarrow (\forall\ y \bullet\ y \in_g x \Rightarrow (\exists\ s\ t \bullet\ y = s \mapsto_g t))$

$\boldsymbol{dom}$ $\vdash \forall\ x \bullet\ dom\ x = Sep\ (Gx\ x)\ (\lambda\ w \bullet\ \exists\ v \bullet\ w \mapsto_g v \in_g x)$

$\boldsymbol{ran}$ $\vdash \forall\ x \bullet\ ran\ x = Sep\ (Gx\ x)\ (\lambda\ w \bullet\ \exists\ v \bullet\ v \mapsto_g w \in_g x)$

$\boldsymbol{field}$ $\vdash \forall\ s\ e \bullet\ e \in_g field\ s \Leftrightarrow e \in_g dom\ s \vee e \in_g ran\ s$

$\boldsymbol{\lhd_g}$ $\vdash \forall\ s\ r \bullet\ s \lhd r = Sep\ r\ (\lambda\ p \bullet\ fst\ p \in_g s)$

$\boldsymbol{\rhd_g}$ $\vdash \forall\ s\ r \bullet\ r \rhd s = Sep\ r\ (\lambda\ p \bullet\ snd\ p \in_g s)$

$\boldsymbol{\nlhd_g}$ $\vdash \forall\ s\ r \bullet\ s \nlhd r = Sep\ r\ (\lambda\ p \bullet\ \neg\ fst\ p \in_g s)$

$\boldsymbol{\nrhd_g}$ $\vdash \forall\ s\ r \bullet\ r \nrhd s = Sep\ r\ (\lambda\ p \bullet\ \neg\ snd\ p \in_g s)$

$\boldsymbol{Rel2DepType_g}$

$\vdash \forall\ r$
- $Rel2DepType_g\ r$
  $= Sep$
  $(Gx\ r)$
  $(\lambda\ e$
  - $\exists\ i\ t$
    - $e = i \mapsto_g t$
      $\wedge\ i \in_g dom\ r$
      $\wedge\ (\forall\ j \bullet\ j \in_g t \Leftrightarrow i \mapsto_g j \in_g r))$

$\boldsymbol{DepSum_g}$

$\vdash \forall\ t$
- $DepSum_g\ t$
  $= Sep$
  $(Gx\ t)$
  $(\lambda\ e$
  - $\exists\ i\ t2\ v$
    - $e = i \mapsto_g v \wedge v \in_g t2 \wedge i \mapsto_g t2 \in_g t)$

$\boldsymbol{\Sigma_g}$

$\vdash \forall\ f\ s$
- $\$\Sigma_g\ f\ s$
  $= \bigcup_g$
  $(Imagep\ (\lambda\ e \bullet\ Imagep\ (\lambda\ x \bullet\ e \mapsto_g x)\ (f\ e))\ s)$

$\boldsymbol{\times_g}$

$\vdash \forall\ s\ t$
- $s \times_g t$
  $= \bigcup_g$
  $(Imagep\ (\lambda\ se \bullet\ Imagep\ (\lambda\ te \bullet\ se \mapsto_g te)\ t)\ s)$

$\boldsymbol{\leftrightarrow_g}$ $\vdash \forall\ s\ t \bullet\ s \leftrightarrow_g t = \mathbb{P}_g\ (s \times_g t)$

o_g $\vdash \forall\ f\ g$
$\bullet\ f\ o_g\ g$
$=\ Imagep$
$(\lambda\ p\bullet\ fst\ (fst\ p) \mapsto_g snd\ (snd\ p))$
$(Sep$
$(g \times_g f)$
$(\lambda\ p\bullet \exists\ q\ r\ s\bullet\ p = (q \mapsto_g r) \mapsto_g r \mapsto_g s))$

fun $\vdash \forall\ x$
$\bullet\ fun\ x$
$\Leftrightarrow\ rel\ x$
$\wedge\ (\forall\ s\ t\ u$
$\bullet\ s \mapsto_g u \in_g x \wedge s \mapsto_g t \in_g x \Rightarrow u = t)$

$\twoheadrightarrow_g$ $\vdash \forall\ s\ t\bullet\ s \twoheadrightarrow_g t = Sep\ (s \leftrightarrow_g t)\ fun$

$\rightarrow_g$ $\vdash \forall\ s\ t\bullet\ s \rightarrow_g t = Sep\ (s \twoheadrightarrow_g t)\ (\lambda\ r\bullet\ dom\ r = s)$

$\rightarrow_g\_closed$ $\vdash \forall\ s$
$\bullet \rightarrow_g\_closed\ s$
$\Leftrightarrow (\forall\ d\ c\bullet\ d \in_g s \wedge c \in_g s \Rightarrow d \rightarrow_g c \in_g s)$

λ_g $\vdash \forall\ f\ s$
$\bullet\ \$\lambda_g\ f\ s$
$=\ Sep\ (s \times_g Imagep\ f\ s)\ (\lambda\ p\bullet\ snd\ p = f\ (fst\ p))$

g $\vdash \forall\ f\ x$
$\bullet\ f\ _g\ x$
$=\ (if\ \exists\ y\bullet\ x \mapsto_g y \in_g f$
$then\ \epsilon\ y\bullet\ x \mapsto_g y \in_g f$
$else\ f)$

id $\vdash \forall\ s\bullet\ id\ s = Sep\ (s \times_g s)\ (\lambda\ x\bullet\ fst\ x = snd\ x)$

$\oplus_g$ $\vdash \forall\ s\ t$
$\bullet\ s \oplus_g t$
$=\ Sep$
$(s \cup_g t)$
$(\lambda\ x$
$\bullet\ if\ fst\ x \in_g dom\ t$
$then\ x \in_g t$
$else\ x \in_g s)$

9.7 Theorems

$\mapsto_g\_eq\_thm$ $\vdash \forall\ s\ t\ u\ v\bullet\ s \mapsto_g t = u \mapsto_g v \Leftrightarrow s = u \wedge t = v$

$Pair_g\_\in\_\mapsto_g\_thm$
$\vdash \forall\ s\ t\bullet\ Pair_g\ s\ t \in_g s \mapsto_g t$

$Pair_g\_\in_g\_Gx\_\mapsto_g\_thm$
$\vdash \forall\ s\ t\bullet\ Pair_g\ s\ t \in_g Gx\ (s \mapsto_g t)$

$\mapsto_g\_\in_g\_Gx\_Pair_g\_thm$
$\vdash \forall\ s\ t\bullet\ s \mapsto_g t \in_g Gx\ (Pair_g\ s\ t)$

$\neg\mapsto_g\varnothing_g\_thm$ $\vdash \forall\ x\ y\bullet\ \neg\ x \mapsto_g y = \varnothing_g$

$\neg\varnothing_g\mapsto_g\_thm$ $\vdash \forall\ x\ y\bullet\ \neg\ \varnothing_g = x \mapsto_g y$

$GClose\mapsto_g\_thm$
$\vdash \forall\ g$
$\bullet\ galaxy\ g$
$\Rightarrow (\forall\ s\ t\bullet\ s \in_g g \wedge t \in_g g \Rightarrow s \mapsto_g t \in_g g)$

$tc\in\_\mapsto\_left\_thm$

$\vdash \forall\ s\ t\bullet\ s \in_g{}^+\ s \mapsto_g t$

tc∈_↦_right_thm

$\vdash \forall\ s\ t\bullet\ t \in_g{}^+\ s \mapsto_g t$

↦_tc_thm $\vdash \forall\ x\ y\bullet\ tc\ \$\in_g\ x\ (x \mapsto_g y) \wedge tc\ \$\in_g\ y\ (x \mapsto_g y)$

rel_∅_g_thm $\vdash rel\ \varnothing_g$

dom_∅_g_thm $\vdash dom\ \varnothing_g = \varnothing_g$

dom_thm $\vdash \forall\ r\ y\bullet\ y \in_g dom\ r \Leftrightarrow (\exists\ x\bullet\ y \mapsto_g x \in_g r)$

dom_Gx_thm $\vdash \forall\ r\bullet\ dom\ r \in_g Gx\ r$

GClose_dom_thm

$\vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ r\bullet\ r \in_g g \Rightarrow dom\ r \in_g g)$

ran_∅_g_thm $\vdash ran\ \varnothing_g = \varnothing_g$

ran_thm $\vdash \forall\ r\ y\bullet\ y \in_g ran\ r \Leftrightarrow (\exists\ x\bullet\ x \mapsto_g y \in_g r)$

GClose_ran_thm

$\vdash \forall\ g\bullet\ galaxy\ g \Rightarrow (\forall\ r\bullet\ r \in_g g \Rightarrow ran\ r \in_g g)$

tc∈_ran_thm $\vdash \forall\ x\ y\bullet\ x \in_g{}^+\ ran\ y \Rightarrow x \in_g{}^+\ y$

field_∅_g_thm

$\vdash field\ \varnothing_g = \varnothing_g$

×_g_spec $\vdash \forall\ s\ t\ e$

$\bullet\ e \in_g s \times_g t$

$\Leftrightarrow (\exists\ l\ r\bullet\ l \in_g s \wedge r \in_g t \wedge e = l \mapsto_g r)$

f↦gs_thm $\vdash \forall\ s\ t\ p\bullet\ p \in_g s \times_g t \Rightarrow fst\ p \mapsto_g snd\ p = p$

v∈g×g_thm $\vdash \forall\ p\ s\ t\bullet\ p \in_g s \times_g t \Rightarrow fst\ p \in_g s \wedge snd\ p \in_g t$

↦g∈g×g_thm

$\vdash \forall\ l\ r\ s\ t\bullet\ l \mapsto_g r \in_g s \times_g t \Leftrightarrow l \in_g s \wedge r \in_g t$

↔g⊆g×g_thm

$\vdash \forall\ s\ t\ r\bullet\ r \in_g s \leftrightarrow_g t \Leftrightarrow r \subseteq_g s \times_g t$

∅g∈g↔g_thm

$\vdash \forall\ s\ t\bullet\ \varnothing_g \in_g s \leftrightarrow_g t$

f↦gs_thm1 $\vdash \forall\ p\ r\ s\ t$

$\bullet\ p \in_g r \wedge r \in_g s \leftrightarrow_g t \Rightarrow fst\ p \mapsto_g snd\ p = p$

∈g↔g_thm $\vdash \forall\ p\ r\ s\ t$

$\bullet\ p \in_g r \wedge r \in_g s \leftrightarrow_g t \Rightarrow fst\ p \in_g s \wedge snd\ p \in_g t$

o_g_thm $\vdash \forall\ f\ g\ x$

$\bullet\ x \in_g f\ o_g\ g$

$\Leftrightarrow (\exists\ q\ r\ s$

$\bullet\ q \mapsto_g r \in_g g \wedge r \mapsto_g s \in_g f \wedge x = q \mapsto_g s)$

o_g_thm2 $\vdash \forall\ f\ g\ x\ y$

$\bullet\ x \mapsto_g y \in_g f\ o_g\ g$

$\Leftrightarrow (\exists\ z\bullet\ x \mapsto_g z \in_g g \wedge z \mapsto_g y \in_g f)$

o_g_rel_thm $\vdash \forall\ r\ s\bullet\ rel\ r \wedge rel\ s \Rightarrow rel\ (r\ o_g\ s)$

o_g_associative_thm

$\vdash \forall\ f\ g\ h\bullet\ (f\ o_g\ g)\ o_g\ h = f\ o_g\ g\ o_g\ h$

rel_sub_cp_thm

$\vdash \forall\ x\bullet\ rel\ x \Leftrightarrow (\exists\ s\ t\bullet\ x \subseteq_g s \times_g t)$

fun_∅_g_thm $\vdash fun\ \varnothing_g$

o_g_fun_thm $\vdash \forall\ f\ g\bullet\ fun\ f \wedge fun\ g \Rightarrow fun\ (f\ o_g\ g)$

ran_o_g_thm $\vdash \forall\ f\ g\bullet\ ran\ (f\ o_g\ g) \subseteq_g ran\ f$

dom_o_g_thm $\vdash \forall\ f\ g\bullet\ dom\ (f\ o_g\ g) \subseteq_g dom\ g$

dom_o_g_thm2 $\vdash \forall\ f\ g\bullet\ ran\ g \subseteq_g dom\ f \Rightarrow dom\ (f\ o_g\ g) = dom\ g$

∅g∈g↛g_thm

$\vdash \forall\ s\ t\bullet\ \varnothing_g \in_g s \nrightarrow_g t$

$\exists \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g$**-thm** $\vdash \forall\ s\ t \bullet \exists\ f \bullet\ f \in_g s \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g t$

$\varnothing_g \in_g \varnothing_g \rightarrow_g$**-thm**
$\vdash \forall\ s\ t \bullet\ \varnothing_g \in_g \varnothing_g \rightarrow_g t$

$\exists \rightarrow_g$**-thm** $\vdash \forall\ s\ t \bullet\ (\exists\ x \bullet\ x \in_g t) \Rightarrow (\exists\ f \bullet\ f \in_g s \rightarrow_g t)$

app-thm1 $\vdash \forall\ f\ x \bullet\ (\exists_1\ y \bullet\ x \mapsto_g y \in_g f) \Rightarrow x \mapsto_g f\ _g\ x \in_g f$

app-thm2 $\vdash \forall\ f\ x\ y \bullet\ fun\ f \wedge x \mapsto_g y \in_g f \Rightarrow f\ _g\ x = y$

app-thm3 $\vdash \forall\ f\ x \bullet\ fun\ f \wedge x \in_g dom\ f \Rightarrow x \mapsto_g f\ _g\ x \in_g f$

o_g**-**g**-thm** $\vdash \forall\ f\ g\ x$
$\bullet\ fun\ f \wedge fun\ g \wedge x \in_g dom\ g \wedge ran\ g \subseteq_g dom\ f$
$\Rightarrow (f\ o_g\ g)\ _g\ x = f\ _g\ g\ _g\ x$

$_g \in_g$**-thm** $\vdash \forall\ f\ s\ t\ u \bullet\ f \in_g s \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g t \wedge u \in_g dom\ f \Rightarrow f\ _g\ u \in_g t$

$_g \in_g$**-thm1** $\vdash \forall\ f\ s\ t\ u \bullet\ f \in_g s \rightarrow_g t \wedge u \in_g s \Rightarrow f\ _g\ u \in_g t$

$\in_g \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g \Rightarrow \in_g \rightarrow_g$**-thm**
$\vdash \forall\ f\ s\ t\ u \bullet\ f \in_g s \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g t \Rightarrow f \in_g dom\ f \rightarrow_g t$

id-thm1 $\vdash \forall\ s\ x \bullet\ x \in_g id\ s \Leftrightarrow (\exists\ y \bullet\ y \in_g s \wedge x = y \mapsto_g y)$

id-ap-thm $\vdash \forall\ s\ x \bullet\ x \in_g s \Rightarrow id\ s\ _g\ x = x$

id$\in_g \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g$**-thm1**
$\vdash \forall\ s\ t\ u \bullet\ s \subseteq_g t \cap_g u \Rightarrow id\ s \in_g t \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g u$

id$\in_g \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g$**-thm2**
$\vdash \forall\ s\ t\ u \bullet\ s \subseteq_g t \Rightarrow id\ s \in_g t \mathbin{\rightarrow\!\!\!\!\!\!\cdot}_g t$

id-clauses $\vdash \forall\ s$
$\bullet\ rel\ (id\ s)$
$\wedge\ fun\ (id\ s)$
$\wedge\ dom\ (id\ s) = s$
$\wedge\ ran\ (id\ s) = s$

$\in_g \oplus_g$**-thm** $\vdash \forall\ s\ t\ x$
$\bullet\ x \in_g s \oplus_g t$
$\Leftrightarrow (if\ fst\ x \in_g dom\ t\ then\ x \in_g t\ else\ x \in_g s)$

$\mapsto_g \in_g \oplus_g$**-thm**
$\vdash \forall\ s\ t\ x\ y$
$\bullet\ x \mapsto_g y \in_g s \oplus_g t$
$\Leftrightarrow x \mapsto_g y \in_g t \vee \neg\ x \in_g dom\ t \wedge x \mapsto_g y \in_g s$

$\oplus_g$**-rel-thm** $\vdash \forall\ s\ t \bullet\ rel\ s \wedge rel\ t \Rightarrow rel\ (s \oplus_g t)$

$\oplus_g$**-fun-thm** $\vdash \forall\ s\ t \bullet\ fun\ s \wedge fun\ t \Rightarrow fun\ (s \oplus_g t)$

10 The Theory gst-ord

10.1 Parents

$gst-ax$

10.2 Children

GS $gst-nat$

10.3 Constants

connected	$GS \to BOOL$
ordinal	$GS \to BOOL$
$\$<_{\boldsymbol{o}}$	$GS \to GS \to BOOL$
$\$\leqslant_{\boldsymbol{o}}$	$GS \to GS \to BOOL$
$\boldsymbol{suc_o}$	$GS \to GS$
successor	$GS \to BOOL$
limit_ordinal	$GS \to BOOL$
$\$\boldsymbol{ub}$	$GS \to GS \to BOOL$
sup	$GS \to GS$
$\$\boldsymbol{sub}$	$GS \to GS \to BOOL$
ssup	$GS \to GS$
rank	$GS \to GS$
$\$+_{\boldsymbol{o}}$	$GS \to GS \to GS$
$\$--_{\boldsymbol{o}}$	$GS \to GS \to GS$

10.4 Fixity

Right Infix 200:

$\boldsymbol{sub}$ $\boldsymbol{ub}$

Right Infix 240:

$<_{\boldsymbol{o}}$ $\leqslant_{\boldsymbol{o}}$

Right Infix 300:

$+_{\boldsymbol{o}}$ $--_{\boldsymbol{o}}$

10.5 Definitions

connected	$\vdash \forall\ s$
	$\bullet$ *connected* s
	$\Leftrightarrow (\forall\ t\ u$
	$\bullet\ t \in_g s \wedge u \in_g s \Rightarrow t \in_g u \vee t = u \vee u \in_g t)$
ordinal	$\vdash \forall\ s\bullet$ *ordinal* $s \Leftrightarrow$ *transitive* $s \wedge$ *connected* s
$<_{\boldsymbol{o}}$	$\vdash \forall\ x\ y\bullet\ x <_o y \Leftrightarrow$ *ordinal* $x \wedge$ *ordinal* $y \wedge x \in_g y$
$\leqslant_{\boldsymbol{o}}$	$\vdash \forall\ x\ y$
	$\bullet\ x \leqslant_o y \Leftrightarrow$ *ordinal* $x \wedge$ *ordinal* $y \wedge (x \in_g y \vee x = y)$
$\boldsymbol{suc_o}$	$\vdash \forall\ x\bullet\ suc_o\ x = x \cup_g Sing\ x$
successor	$\vdash \forall\ s\bullet$ *successor* $s \Leftrightarrow (\exists\ t\bullet$ *ordinal* $t \wedge s = suc_o\ t)$
limit_ordinal	

$\vdash \forall\ s$
$\bullet\ limit\_ordinal\ s$
$\Leftrightarrow ordinal\ s \wedge \neg\ successor\ s \wedge \neg\ s = \varnothing_g$

ub $\vdash \forall\ \alpha\ \beta \bullet\ \alpha\ ub\ \beta \Leftrightarrow (\forall\ \gamma \bullet\ \gamma \in_g \alpha \Rightarrow \gamma \leqslant_o \beta)$

sup $\vdash \forall\ \alpha \bullet\ sup\ \alpha = \bigcup_g \alpha$

sub $\vdash \forall\ \alpha\ \beta \bullet\ \alpha\ sub\ \beta \Leftrightarrow (\forall\ \gamma \bullet\ \gamma \in_g \alpha \Rightarrow \gamma <_o \beta)$

ssup $\vdash \forall\ \alpha \bullet\ ssup\ \alpha = \bigcup_g (Imagep\ suc_o\ \alpha)$

rank $\vdash \forall\ x \bullet\ rank\ x = \bigcup_g (Imagep\ (suc_o\ o\ rank)\ x)$

$+_o$ $\vdash \forall\ \alpha\ \beta$
$\bullet\ \alpha +_o \beta$
$= (if\ \beta = \varnothing_g$
$then\ \alpha$
$else\ ssup\ (Imagep\ (\$+_o\ \alpha)\ \beta))$

$--_o$ $\vdash T$

10.6 Theorems

mem_less_thm $\vdash \forall\ \alpha\ \beta \bullet\ ordinal\ \alpha \wedge ordinal\ \beta \wedge \alpha \in_g \beta \Rightarrow \alpha <_o \beta$

less_mem_thm $\vdash \forall\ \alpha\ \beta \bullet\ \alpha <_o \beta \Rightarrow ordinal\ \alpha \wedge ordinal\ \beta \wedge \alpha \in_g \beta$

ord_mem_psub_thm
$\vdash \forall\ \alpha \bullet\ ordinal\ \alpha \Rightarrow (\forall\ \beta \bullet\ \beta \in_g \alpha \Rightarrow \beta \subset_g \alpha)$

lto_psub_thm $\vdash \forall\ \alpha\ \beta \bullet\ \alpha <_o \beta \Rightarrow \alpha \subset_g \beta$

lo_trans_thm $\vdash \forall\ \alpha\ \beta\ \gamma \bullet\ \alpha <_o \beta \wedge \beta <_o \gamma \Rightarrow \alpha <_o \gamma$

leo_lo_thm $\vdash \forall\ x\ y$
$\bullet\ x \leqslant_o y \Leftrightarrow ordinal\ x \wedge ordinal\ y \wedge (x <_o y \vee x = y)$

leo_sub_thm $\vdash \forall\ \alpha\ \beta \bullet\ \alpha \leqslant_o \beta \Rightarrow \alpha \subseteq_g \beta$

leo_trans_thm
$\vdash \forall\ \alpha\ \beta\ \gamma \bullet\ \alpha \leqslant_o \beta \wedge \beta \leqslant_o \gamma \Rightarrow \alpha \leqslant_o \gamma$

leo_lo_trans_thm
$\vdash \forall\ \alpha\ \beta\ \gamma \bullet\ \alpha \leqslant_o \beta \wedge \beta <_o \gamma \Rightarrow \alpha <_o \gamma$

lo_leo_trans_thm
$\vdash \forall\ \alpha\ \beta\ \gamma \bullet\ \alpha <_o \beta \wedge \beta \leqslant_o \gamma \Rightarrow \alpha <_o \gamma$

ordinal_$\varnothing_g$ $\vdash ordinal\ \varnothing_g$

trans_suc_trans
$\vdash \forall\ x \bullet\ transitive\ x \Rightarrow transitive\ (suc_o\ x)$

conn_suc_conn
$\vdash \forall\ x \bullet\ connected\ x \Rightarrow connected\ (suc_o\ x)$

ord_suc_ord_thm
$\vdash \forall\ x \bullet\ ordinal\ x \Rightarrow ordinal\ (suc_o\ x)$

$\varnothing_g$_not_suc$_o$_thm
$\vdash \neg\ (\exists\ \alpha \bullet\ suc_o\ \alpha = \varnothing_g)$

not_in_suco_thm
$\vdash \forall\ \alpha \bullet\ \neg\ \alpha = suc_o\ \alpha$

leo_suc_thm $\vdash \forall\ \alpha \bullet\ ordinal\ \alpha \Rightarrow \alpha \leqslant_o suc_o\ \alpha$

lo_suc_thm $\vdash \forall\ \alpha \bullet\ ordinal\ \alpha \Rightarrow \alpha <_o suc_o\ \alpha$

conn_sub_conn
$\vdash \forall\ x \bullet\ connected\ x \Rightarrow (\forall\ y \bullet\ y \subseteq_g x \Rightarrow connected\ y)$

conn_mem_ord $\vdash \forall\ x \bullet\ ordinal\ x \Rightarrow (\forall\ y \bullet\ y \in_g x \Rightarrow connected\ y)$

tran_mem_ord $\vdash \forall\ x \bullet\ ordinal\ x \Rightarrow (\forall\ y \bullet\ y \in_g x \Rightarrow transitive\ y)$

ord_mem_ord $\vdash \forall\ x \bullet\ ordinal\ x \Rightarrow (\forall\ y \bullet\ y \in_g x \Rightarrow ordinal\ y)$

GCloseSuc $\vdash \forall\ g \bullet\ galaxy\ g \Rightarrow (\forall\ x \bullet\ x \in_g g \Rightarrow suc_o\ x \in_g g)$

tran_∩_thm $\vdash \forall\ x\ y$
$\bullet\ transitive\ x\ \wedge\ transitive\ y \Rightarrow transitive\ (x\ \cap_g\ y)$

tran_∪_thm $\vdash \forall\ x\ y$
$\bullet\ transitive\ x\ \wedge\ transitive\ y \Rightarrow transitive\ (x\ \cup_g\ y)$

conn_∩_thm $\vdash \forall\ x\ y$
$\bullet\ connected\ x\ \wedge\ connected\ y \Rightarrow connected\ (x\ \cap_g\ y)$

ord_∩_thm $\vdash \forall\ x\ y\bullet\ ordinal\ x\ \wedge\ ordinal\ y \Rightarrow ordinal\ (x\ \cap_g\ y)$

trich_for_ords_thm
$\vdash \forall\ x\ y$
$\bullet\ ordinal\ x\ \wedge\ ordinal\ y \Rightarrow x <_o y\ \vee\ x = y\ \vee\ y <_o x$

sub_leo_thm $\vdash \forall\ x\ y\bullet\ ordinal\ x\ \wedge\ ordinal\ y \Rightarrow (x \subseteq_g y \Leftrightarrow x \leqslant_o y)$

sub_leo_thm1 $\vdash \forall\ x\ y\bullet\ ordinal\ x\ \wedge\ ordinal\ y\ \wedge\ x \subseteq_g y \Rightarrow x \leqslant_o y$

successor_ord_thm
$\vdash \forall\ x\bullet\ successor\ x \Rightarrow ordinal\ x$

wf_ordinals_thm
$\vdash well\_founded\ \$<_o$

ordinal_kind_thm
$\vdash \forall\ n$
$\bullet\ ordinal\ n \Rightarrow n = \varnothing_g\ \vee\ successor\ n\ \vee\ limit\_ordinal\ n$

ordinal_limit_thm
$\vdash \forall\ \alpha\bullet\ (\forall\ \beta\bullet\ \beta \in_g \alpha \Rightarrow ordinal\ \beta) \Rightarrow ordinal\ (\bigcup_g\ \alpha)$

sup_lub_thm $\vdash \forall\ \alpha$
$\bullet\ (\forall\ \beta\bullet\ \beta \in_g \alpha \Rightarrow ordinal\ \beta)$
$\Rightarrow \alpha\ ub\ sup\ \alpha$
$\wedge\ (\forall\ \gamma\bullet\ ordinal\ \gamma\ \wedge\ \alpha\ ub\ \gamma \Rightarrow sup\ \alpha \leqslant_o \gamma)$

ordinal_ssup_thm
$\vdash \forall\ \alpha\bullet\ (\forall\ \beta\bullet\ \beta \in_g \alpha \Rightarrow ordinal\ \beta) \Rightarrow ordinal\ (ssup\ \alpha)$

ord_plus_thm $\vdash \forall\ \alpha\ \beta\bullet\ ordinal\ \alpha\ \wedge\ ordinal\ \beta \Rightarrow ordinal\ (\alpha +_o \beta)$

11 The Theory gst-nat

11.1 Parents

$gst-ord$

11.2 Children

GS

11.3 Constants

natural_number
$GS \to BOOL$

$\$<_{gn}$ $GS \to GS \to BOOL$

$\boldsymbol{Nat_g}$ $\mathbb{N} \to GS$

11.4 Fixity

Right Infix 240:
$<_{gn}$

11.5 Definitions

natural_number
$$\vdash \forall\ s \bullet\ natural\_number\ s \Leftrightarrow s = \varnothing_g \vee successor\ s \wedge (\forall\ t \bullet\ t \in_g s \Rightarrow t = \varnothing_g \vee successor\ t)$$

$<_{gn}$
$$\vdash \forall\ x\ y \bullet\ x <_{gn} y \Leftrightarrow natural\_number\ x \wedge natural\_number\ y \wedge x \in_g y$$

$\boldsymbol{Nat_g}$
$$\vdash Nat_g\ 0 = \varnothing_g \wedge (\forall\ n \bullet\ Nat_g\ (n + 1) = suc_o\ (Nat_g\ n))$$

11.6 Theorems

wf_nat_thm
$$\vdash well\_founded\ \$<_{gn}$$

nat_induct_thm
$$\vdash \forall\ s \bullet\ (\forall\ x \bullet\ (\forall\ y \bullet\ y <_{gn} x \Rightarrow s\ y) \Rightarrow s\ x) \Rightarrow (\forall\ x \bullet\ s\ x)$$

nat_induct_thm2
$$\vdash \forall\ p \bullet\ (\forall\ x \bullet\ natural\_number\ x \wedge (\forall\ y \bullet\ y <_{gn} x \Rightarrow p\ y) \Rightarrow p\ x) \Rightarrow (\forall\ x \bullet\ natural\_number\ x \Rightarrow p\ x)$$

ord_nat_thm
$$\vdash \forall\ n \bullet\ natural\_number\ n \Rightarrow ordinal\ n$$

mem_nat_ord_thm
$$\vdash \forall\ n \bullet\ natural\_number\ n \Rightarrow (\forall\ m \bullet\ m \in_g n \Rightarrow ordinal\ m)$$

nat_not_lim_thm

$$\vdash \forall\ n\bullet\ natural\_number\ n \Rightarrow \neg\ limit\_ordinal\ n$$

nat_zero_or_suc_thm

$$\vdash \forall\ n\bullet\ natural\_number\ n \Rightarrow successor\ n \vee n = \varnothing_g$$

mem_nat_not_lim_thm

$$\vdash \forall\ m\ n\bullet\ natural\_number\ n \wedge m \in_g n \Rightarrow \neg\ limit\_ordinal\ m$$

mem_nat_nat_thm

$$\vdash \forall\ n \bullet\ natural\_number\ n \Rightarrow (\forall\ m\bullet\ m \in_g n \Rightarrow natural\_number\ m)$$

nat_in_G$\varnothing_g$_thm

$$\vdash \forall\ n\bullet\ natural\_number\ n \Rightarrow n \in_g Gx\ \varnothing_g$$

w_exists_thm $\vdash \exists\ w\bullet \forall\ z\bullet\ z \in_g w \Leftrightarrow natural\_number\ z$

ord_nat_thm2 $\vdash \forall\ n\bullet\ ordinal\ (Nat_g\ n)$

not_suc_nat_zero_thm

$$\vdash \forall\ n\bullet \neg\ suc_o\ (Nat_g\ n) = \varnothing_g$$

less_sum_thm $\vdash \forall\ x\ y\bullet\ x \leqslant y \Rightarrow (\exists\ z\bullet\ x + z = y)$

natg_mono_thm

$$\vdash \forall\ x\ y\bullet\ Nat_g\ x \leqslant_o Nat_g\ (x + y)$$

natg_one_one_thm

$$\vdash \forall\ x\ y\bullet\ Nat_g\ x = Nat_g\ y \Rightarrow x = y$$

natg_one_one_thm2

$$\vdash \forall\ x\ y\bullet\ Nat_g\ x = Nat_g\ y \Leftrightarrow x = y$$

12 The Theory GS

12.1 Parents

gst−nat *gst−ord* *gst−fun*

12.2 Children

ICsynmisc2

674

13 INDEX