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.
## Contents

1 Introduction  

2 Ontology  
    2.1 Semantic Universality  
    2.2 Well-Founded Collections  

3 Axioms  
    3.1 Introduction  
    3.1.1 Scope  
    3.1.2 Why Galactic?  
    3.2 Membership  
    3.2.1 The Type GS  
    3.3 Membership  
    3.3.1 Extensionality  
    3.3.2 Well-Foundedness  
    3.4 The Ontology Axiom  
    3.4.1 Subsets  
    3.4.2 The Ontology Axiom  
    3.5 PowerSets and Union  
    3.5.1 PowerSets  
    3.5.2 Union  
    3.6 Relational Replacement  
    3.6.1 RelIm  
    3.7 Separation  
    3.8 Galaxies  
    3.8.1 Definition of Galaxy  
    3.8.2 Definition of Gx  
    3.8.3 Galaxy Closure  
    3.8.4 The Empty Set  
    3.9 Functional Replacement  
    3.9.1 Introduction  
    3.9.2 Imagep  
    3.9.3 Galaxy Closure  
    3.10 Pair and Unit sets  
    3.10.1 Unit Sets  
    3.10.2 Galaxy Closure  
    3.10.3 Sing-Pair equations  
    3.11 Union and Intersection  
    3.11.1 Binary Union  
    3.11.2 Galaxy Closure  
    3.11.3 Distributed Intersection  
    3.11.4 Binary Intersection  
    3.11.5 Galaxy Closure  
    3.11.6 Consequences of Well-Foundedness  
    3.12 Galaxy Closure Clauses  
    3.13 Proof Context  
    3.13.1 Principles  
    3.13.2 Theorems Used Recklessly  
    3.13.3 Theorems Used Cautiously  
    3.13.4 Automatic Proof
### 4 Products and Sums

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Ordered Pairs</td>
<td>21</td>
</tr>
<tr>
<td>4.1.1 Projections</td>
<td>22</td>
</tr>
<tr>
<td>4.1.2 MkPair and MkTriple</td>
<td>22</td>
</tr>
<tr>
<td>4.2 Relations</td>
<td>23</td>
</tr>
<tr>
<td>4.3 Domain and Range Restrictions</td>
<td>24</td>
</tr>
<tr>
<td>4.4 Dependent Types</td>
<td>24</td>
</tr>
<tr>
<td>4.5 Dependent Sums and Cartesian Products</td>
<td>25</td>
</tr>
<tr>
<td>4.5.1 Relation Space</td>
<td>26</td>
</tr>
<tr>
<td>4.5.2 Another Pair-Projection Inverse Theorem</td>
<td>26</td>
</tr>
<tr>
<td>4.5.3 Member of Relation Theorem</td>
<td>27</td>
</tr>
<tr>
<td>4.5.4 Relational Composition</td>
<td>27</td>
</tr>
<tr>
<td>4.5.5 Relation Subset of Cartesian Product</td>
<td>28</td>
</tr>
<tr>
<td>4.6 Functions</td>
<td>28</td>
</tr>
<tr>
<td>4.6.1 fun</td>
<td>28</td>
</tr>
<tr>
<td>4.6.2 lemmas</td>
<td>28</td>
</tr>
<tr>
<td>4.6.3 Partial Function Space</td>
<td>29</td>
</tr>
<tr>
<td>4.6.4 Partial Function Space Non-Empty</td>
<td>29</td>
</tr>
<tr>
<td>4.6.5 Function Space</td>
<td>29</td>
</tr>
<tr>
<td>4.6.6 Function Space Non-Empty</td>
<td>30</td>
</tr>
<tr>
<td>4.7 Functional Abstraction</td>
<td>30</td>
</tr>
<tr>
<td>4.7.1 Abstraction</td>
<td>30</td>
</tr>
<tr>
<td>4.8 Application and Extensionality</td>
<td>31</td>
</tr>
<tr>
<td>4.8.1 Application</td>
<td>31</td>
</tr>
<tr>
<td>4.8.2 The &quot;Type&quot; of an Application (1)</td>
<td>32</td>
</tr>
<tr>
<td>4.8.3 The &quot;Type&quot; of an Application (2)</td>
<td>32</td>
</tr>
<tr>
<td>4.8.4 Partial functions are total</td>
<td>32</td>
</tr>
<tr>
<td>4.9 The Identity Function</td>
<td>32</td>
</tr>
<tr>
<td>4.9.1 specification</td>
<td>32</td>
</tr>
<tr>
<td>4.9.2 lemmas</td>
<td>33</td>
</tr>
<tr>
<td>4.10 Override</td>
<td>33</td>
</tr>
<tr>
<td>4.11 Proof Contexts</td>
<td>34</td>
</tr>
<tr>
<td>4.11.1 Proof Context</td>
<td>34</td>
</tr>
</tbody>
</table>

### 5 Ordinals

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Definitions 2.1 and 2.3</td>
<td>35</td>
</tr>
<tr>
<td>5.1.1 The Definition</td>
<td>35</td>
</tr>
<tr>
<td>5.2 Theorem 2.2</td>
<td>37</td>
</tr>
<tr>
<td>5.2.1 The Empty Set is an Ordinal</td>
<td>37</td>
</tr>
<tr>
<td>5.2.2 The Successor of an Ordinal is an Ordinal</td>
<td>37</td>
</tr>
<tr>
<td>5.2.3 The Ordinal Zero is not a successor</td>
<td>37</td>
</tr>
<tr>
<td>5.2.4 The members of an Ordinal are Ordinals</td>
<td>38</td>
</tr>
<tr>
<td>5.2.5 Galaxies are Closed under suc</td>
<td>39</td>
</tr>
<tr>
<td>5.3 Theorem 2.4</td>
<td>39</td>
</tr>
<tr>
<td>5.3.1</td>
<td>39</td>
</tr>
<tr>
<td>5.4 Definition 2.6</td>
<td>40</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>5.4.1</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Theorem 2.7</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Successors are Ordinals</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Well-foundedness of the ordinals</td>
</tr>
<tr>
<td>5.5.3</td>
<td>An Ordinal is Zero, a successor or a limit</td>
</tr>
<tr>
<td>5.6</td>
<td>Supremum and Strict Supremum</td>
</tr>
<tr>
<td>5.7</td>
<td>Rank</td>
</tr>
<tr>
<td>5.7.1</td>
<td>The Consistency Proof</td>
</tr>
<tr>
<td>5.8</td>
<td>Ordinal Arithmetic</td>
</tr>
<tr>
<td>5.8.1</td>
<td>Addition</td>
</tr>
<tr>
<td>5.8.2</td>
<td>Subtraction</td>
</tr>
<tr>
<td>5.9</td>
<td>Proof Context</td>
</tr>
<tr>
<td>5.9.1</td>
<td>Proof Context</td>
</tr>
<tr>
<td>6</td>
<td>Natural Numbers</td>
</tr>
<tr>
<td>6.0.2</td>
<td>Ordering the Natural Numbers</td>
</tr>
<tr>
<td>6.1</td>
<td>Theorem 2.8</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Natural Numbers are Ordinals</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Members of Natural Numbers are Ordinals</td>
</tr>
<tr>
<td>6.1.3</td>
<td>A Natural Number is not a Limit Ordinal</td>
</tr>
<tr>
<td>6.1.4</td>
<td>A Natural Number is zero or a successor</td>
</tr>
<tr>
<td>6.1.5</td>
<td>A Natural Number does not contain a Limit Ordinal</td>
</tr>
<tr>
<td>6.1.6</td>
<td>All Members of Natural Numbers are Natural Numbers</td>
</tr>
<tr>
<td>6.1.7</td>
<td>Natural Numbers are in the Smallest Galaxy</td>
</tr>
<tr>
<td>6.1.8</td>
<td>The Existence of w</td>
</tr>
<tr>
<td>6.2</td>
<td>Naming the Natural Numbers</td>
</tr>
<tr>
<td>6.3</td>
<td>Proof Context</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Proof Context</td>
</tr>
<tr>
<td>7</td>
<td>Closing</td>
</tr>
<tr>
<td>8</td>
<td>The Theory gst-ax</td>
</tr>
<tr>
<td>8.1</td>
<td>Parents</td>
</tr>
<tr>
<td>8.2</td>
<td>Children</td>
</tr>
<tr>
<td>8.3</td>
<td>Constants</td>
</tr>
<tr>
<td>8.4</td>
<td>Types</td>
</tr>
<tr>
<td>8.5</td>
<td>Fixity</td>
</tr>
<tr>
<td>8.6</td>
<td>Axioms</td>
</tr>
<tr>
<td>8.7</td>
<td>Definitions</td>
</tr>
<tr>
<td>8.8</td>
<td>Theorems</td>
</tr>
<tr>
<td>9</td>
<td>The Theory gst-fun</td>
</tr>
<tr>
<td>9.1</td>
<td>Parents</td>
</tr>
<tr>
<td>9.2</td>
<td>Children</td>
</tr>
<tr>
<td>9.3</td>
<td>Constants</td>
</tr>
<tr>
<td>9.4</td>
<td>Aliases</td>
</tr>
<tr>
<td>9.5</td>
<td>Fixity</td>
</tr>
<tr>
<td>9.6</td>
<td>Definitions</td>
</tr>
<tr>
<td>9.7</td>
<td>Theorems</td>
</tr>
<tr>
<td>10</td>
<td>The Theory gst-ord</td>
</tr>
<tr>
<td>10.1</td>
<td>Parents</td>
</tr>
</tbody>
</table>
10.2 Children ......................................................... 60
10.3 Constants ......................................................... 60
10.4 Fixity ............................................................. 60
10.5 Definitions ......................................................... 60
10.6 Theorems ......................................................... 61

11 The Theory gst-nat ................................................. 63
  11.1 Parents ......................................................... 63
  11.2 Children ......................................................... 63
  11.3 Constants ......................................................... 63
  11.4 Fixity ............................................................. 63
  11.5 Definitions ......................................................... 63
  11.6 Theorems ......................................................... 63

12 The Theory GS ....................................................... 65
  12.1 Parents ......................................................... 65
  12.2 Children ......................................................... 65

13 INDEX ............................................................... 67
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 bizzare 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 informal 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 universality 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 epimenides’ 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 $\text{Pair}_g$ constructor is introduced axiomatically. In this axiomatisation neither the empty set nor $\text{Pair}_g$ are primitive, they are introduced by definition once the axioms have been presented. Same goes for separation and intersection. The theory $\text{gst-ax}$ 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 $\text{basic-hol}$. 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 vocabulary 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 pseudo-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.3-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:
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
\[
\text{val } gs\_ext\_axiom = \text{new}\_axiom ("gs\_ext\_axiom"),
\]
\[
\forall s\;t:\text{GS} \cdot s = t \iff \forall e \cdot e \in s \iff e \in t
\]

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
\[
X_g : \text{GS} \to \text{GS SET}
\]
\[
\forall s \cdot X_g s = \{ t \mid t \in s \}
\]

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
\[
\text{val } gs\_wf\_axiom = \text{new}\_axiom ("gs\_wf\_axiom"), \quad UWellFounded \$\in_g $
\]
\[
gs\_wf\_thm1 = \quad \vdash \text{well-founded } \$\in_g $
\]
\[
gs\_wf\_min\_thm = \quad \vdash \forall x \cdot (\exists y \cdot y \in_g x) \Rightarrow (\exists z \cdot z \in_g x \land \neg (\exists v \cdot v \in_g z \land v \in_g x))
\]
\[
gs\_wf\_tc\_thm = \quad \vdash \text{well-founded } (\text{tc } \$\in_g $
\]

SML
\[
\text{declare}_\text{infix} (230, "\in_g +")$
\]

HOL Constant
\[
\$\in_g + : \text{GS} \to \text{GS} \to \text{BOOL}
\]
\[
\$\in_g + = \text{tc } \$\in_g$
\]
The resulting induction principle (sometimes called Neotherian induction) is useful.

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

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 hierarchy. 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.

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 hierarchy. 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.
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"],
  \forall s • \exists g • s \in_g g
  \land
  \forall t • t \in_g g \Rightarrow t \subseteq_g t
  \land (\exists p • (\forall v • v \in_g p \Leftrightarrow v \subseteq_g t) \land p \subseteq_g g)
  \land (\exists u • (\forall v • v \in_g u \Leftrightarrow \exists w • v \in_g w \land w \subseteq_g t) \land u \subseteq_g g)
  \land (\forall rel • ManyOne rel \Rightarrow
    (\exists r • (\forall v • v \in_g r \Leftrightarrow \exists w • w \in_g t \land rel w v) \land
     (r \subseteq_g g \Rightarrow r \subseteq_g g)))
```

3.5 PowerSets and Union

Here we define the powerset and union operators.
3.5.1 PowerSets

HOL Constant

\[ \mathbb{P}_g : GS \rightarrow GS \]

\[ \forall s \ t : GS \bullet t \in_g \mathbb{P}_g s \iff t \subseteq_g s \]

\[ sps_{\text{thm}} = \vdash \forall s \bullet s \in_g \mathbb{P}_g s \]

\[ stc\mathbb{P}_g s_{\text{thm}} = \vdash \forall s \bullet s \in_g^+ \mathbb{P}_g s \]

3.5.2 Union

HOL Constant

\[ \bigcup_g : GS \rightarrow GS \]

\[ \forall s \ t : \bigcup_g s \iff \exists e : t \in_g e \wedge e \in_g s \]

\[ eg_{\bigcup_g s_{\text{thm}}} = \vdash \forall s \bullet GS t \in_g s \Rightarrow t \subseteq_g \bigcup_g s \]

3.6 Relational Replacement

The constant RelIm is defined for relational replacement.

3.6.1 RelIm

HOL Constant

\[ \text{RelIm} : (GS \rightarrow GS \rightarrow \text{BOOL}) \rightarrow GS \rightarrow GS \]

\[ \forall rel \ s \ t : \text{ManyOne rel} \Rightarrow (t \in_g \text{RelIm rel} s \iff \exists e : e \in_g s \wedge \text{rel e t}) \]

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

\[ \text{Sep} : GS \rightarrow (GS \rightarrow \text{BOOL}) \rightarrow GS \]

\[ \forall s \ p : e \in_g (\text{Sep s p}) \iff e \in_g s \wedge p e \]

\[ \text{Sep}\_\text{sub}\_\text{thm} = \vdash \forall s \bullet \text{Sep s p} \subseteq_g s \]

\[ \text{Sep}\_\text{sub}\_\text{thm2} = \vdash \forall s \ p : e \bullet e \in_g \text{Sep s p} \Rightarrow e \in_g s \]

\[ \text{Sep}\_\subseteq\_\text{thm} = \vdash \forall s \bullet t \subseteq_g s \Rightarrow \text{Sep s (CombC \$e_g t)} = t \]
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.

\[
galaxy: GS \rightarrow BOOL
\]

\[
\forall s \bullet
\]

\[
galaxy s \iff (\exists x \bullet x \in_g s)
\]

\[
\land \forall t \bullet t \in_g s
\]

\[
\Rightarrow t \subseteq_g s
\]

\[
\land \forall_g t \in_g s
\]

\[
\land \bigcup_g t \in_g s
\]

\[
\land (\forall rel \bullet \text{ManyOne } rel
\]

\[
\Rightarrow \text{RelIm } rel t \subseteq_g s
\]

\[
\Rightarrow \text{RelIm } rel t \in_g s)
\]

\[
galaxies_\exists_thm =
\]

\[
\vdash \forall s \bullet \exists g \bullet s \in_g g \land galaxy g
\]

3.8.2 Definition of Gx

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

\[
Gx: GS \rightarrow GS
\]

\[
\forall s t \bullet t \in_g Gx s \iff \forall g \bullet galaxy g \land s \in_g g \Rightarrow t \in_g g
\]

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:

\[
t_in_Gx_t_thm =
\]

\[
t \in Gx_t
\]

\[
t \subseteq Gx_t
\]

\[
galaxy_Gx =
\]

\[
\vdash \forall s \bullet galaxy (Gx s)
\]

\[
Gx \subseteq_g galaxy =
\]

\[
\vdash \forall s \bullet galaxy g \land s \in_g g \Rightarrow (Gx s) \subseteq_g g
\]
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

\[ \text{transitive} : GS \rightarrow BOOL \]

\[ \forall s \bullet \text{transitive } s \leftrightarrow \forall e \bullet e \in_g s \Rightarrow e \subseteq_g s \]

\[ \text{GalaxiesTransitive.thm} = \quad \vdash \forall s \bullet \text{galaxy } s \Rightarrow \text{transitive } s \]

\[ \begin{align*}
\text{GClose}_{\mathbb{P}} \text{.thm} &= \quad \vdash \forall g \bullet \text{galaxy } g \Rightarrow (\forall s \bullet s \in_g g \Rightarrow \mathbb{P}_g s \in_g g) \\
\text{GClose}_{\bigcup} \text{.thm} &= \quad \vdash \forall g \bullet \text{galaxy } g \Rightarrow (\forall s \bullet s \in_g g \Rightarrow \bigcup_g s \in_g g) \\
\text{GClose}_{\text{Sep}} \text{.thm} &= \quad \vdash \forall g \bullet \text{galaxy } g \Rightarrow (\forall s \bullet s \in_g g \Rightarrow \forall p \bullet \text{Sep } s p \in_g g) \\
\text{GClose}_{\subseteq} \text{.thm} &= \quad \vdash \forall g \bullet \text{galaxy } g \Rightarrow (\forall s \bullet s \in_g g \Rightarrow (\forall t \bullet t \subseteq_g s \Rightarrow t \in_g g))
\end{align*} \]

\[ \begin{align*}
\text{GClose}_{-} \text{fc clauses} = \quad \vdash \forall g \\
&\bullet \text{galaxy } g \\
&\quad \Rightarrow (\forall s \\
&\bullet s \in_g g \\
&\quad \Rightarrow \mathbb{P}_g s \in_g g \\
&\quad \land \bigcup_g s \in_g g \\
&\quad \land (\forall p \bullet \text{Sep } s p \in_g g) \\
&\quad \land (\forall t \bullet t \subseteq_g s \Rightarrow t \in_g g))
\end{align*} \]

\[ \begin{align*}
\text{tc} \in_g \text{lemma} &= \quad \vdash \forall s e \bullet e \in^+_g s \Rightarrow (\forall t \bullet \text{transitive } t \land s \subseteq_g t \Rightarrow e \in_g t) \\
\text{GClose}_{-} \text{tc} \in_g \text{.thm} &= \quad \vdash \forall s g \bullet \text{galaxy } g \Rightarrow s \in^+_g g \Rightarrow s \in_g g
\end{align*} \]

\[ \begin{align*}
\text{Gx} \text{.mono.thm} &= \quad \vdash \forall s t \bullet s \subseteq_g t \Rightarrow \text{Gx } s \subseteq_g \text{Gx } t \\
\text{Gx} \text{.mono.thm2} &= \quad \vdash \forall s t \bullet s \in_g t \Rightarrow \text{Gx } s \subseteq_g \text{Gx } t
\end{align*} \]

\[ \begin{align*}
\text{Gx} \text{.trans.thm} &= \quad \vdash \forall s \bullet \text{transitive } (\text{Gx } s) \\
\text{Gx} \text{.trans.thm2} &= \quad \vdash \forall s t \bullet s \in_g t \Rightarrow s \in_g \text{Gx } t \\
\text{Gx} \text{.trans.thm3} &= \quad \vdash \forall s t u \bullet s \in_g t \land t \in_g \text{Gx } u \Rightarrow s \in_g \text{Gx } u \\
t \text{.sub_Gx} \text{.t.thm} &= \quad \vdash \forall t \bullet t \subseteq_g \text{Gx } t
\end{align*} \]

\[ \begin{align*}
\text{Gx} \text{.mono.thm3} &= \quad \vdash \forall s t \bullet s \subseteq_g t \Rightarrow s \subseteq_g \text{Gx } t \\
\text{Gx} \text{.mono.thm4} &= \quad \vdash \forall s t \bullet s \subseteq_g t \Rightarrow s \in_g \text{Gx } t
\end{align*} \]
3.8.4 The Empty Set

We can now prove that there is an empty set.

So we define \(\emptyset\) as the empty set:

\[
\emptyset : GS
\]

\[
\forall s \cdot s \not\in g \emptyset
\]

and the empty set is a member of every galaxy:

\[
G \emptyset C = \vdash \forall g \cdot \text{galaxy } g \Rightarrow \emptyset \in g
\]

\[
\emptyset g \emptyset \text{thm} = \vdash \forall s \cdot \emptyset g \subseteq g s
\]

\[
\bigcup g \emptyset g \text{thm} = \vdash \bigcup g \emptyset g = \emptyset g
\]

\[
\emptyset g \text{spec} = \vdash \forall s \cdot s \not\in g \emptyset
\]

\[
\text{mem} \not\emptyset \text{thm} = \vdash \forall m n \cdot m \in g n \Rightarrow n = \emptyset g
\]

\[
\emptyset g \emptyset \text{galaxy thm} = \vdash \forall x \cdot \text{galaxy } x \Rightarrow \emptyset g \in g x
\]

\[
\emptyset g \emptyset G x \text{thm} = \vdash \forall x \cdot \emptyset g \in g G x x
\]

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.

\(\text{Imagep } f s\) is the image of \(s\) through \(f\).

\[
\text{Imagep } : (GS \to GS) \to GS \to GS
\]

\[
\forall f s x \cdot x \in g f s \Leftrightarrow \exists e \cdot e \in g s \wedge x = f e
\]
3.9.3 Galaxy Closure

We now show that galaxies are closed under Imagep.

\[ G Imagep C = \forall g \bullet \text{galaxy } g \Rightarrow \forall s \bullet s \in_g g \Rightarrow \forall f \bullet \text{Imagep } f \subseteq_g g \Rightarrow \text{Imagep } f \subseteq_g g \]

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

\[ Pair_g : GS \rightarrow GS \rightarrow GS \]

\[ \forall s \ t \ e:GS \bullet e \in_g Pair_g s \ t \iff e = s \lor e = t \]

\[ Pair_g \subseteq \text{thm} = \forall x \ y \bullet x \in_g Pair_g x \ y \land y \in_g Pair_g x \ y \]

\[ Pair_g \text{-toe thm} = \forall s \ t \bullet s \in_g Pair_g s \ t \land t \in_g Pair_g s \ t \]

\[ Pair_g \text{-eq thm} = \forall s \ t \ u \ v \bullet Pair_g s \ t = Pair_g u \ v \iff s = u \land t = v \lor s = v \land t = u \]

\[ GC close Pair_g = \forall g \bullet \text{galaxy } g \Rightarrow \forall s \bullet s \in_g g \land t \in_g g \Rightarrow Pair_g s \ t \in_g g \]

3.10.1 Unit Sets

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

HOL Constant

\[ Sing : GS \rightarrow GS \]

\[ \forall s \bullet Sing s = Pair_g s \ s \]

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

\[ Sing \text{-thm} = \forall s \bullet e \in_g Sing s \iff e = s \]

\[ Sing \text{-thm2} = \forall x \bullet x \in_g Sing x \]

\[ Sing \text{-toe thm} = \forall x \bullet x \in_g Sing x \]

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

\[ Sing \text{-eq thm} = \forall s \bullet Sing s = Sing t \iff s = t \]
3.10.2 Galaxy Closure

\[ G_{\text{Close\ Sing}} = \vdash \forall g \bullet \text{galaxy } g \Rightarrow \forall s \bullet s \in g \Rightarrow \text{Sing } s \in g \]

3.10.3 Sing-Pair equations

The following theorems tell you when Pairs are really Sings.

\[ \text{Sing}_{\ P\ eq\ thm} = \vdash \forall s \ t \ u \bullet \text{Sing } s = \text{Pair } t \ u \iff s = t \land s = u \]

\[ \text{Pair}_{\ S\ eq\ thm} = \vdash \forall s \ t \ u \bullet \text{Pair } s \ t = \text{Sing } u \iff s = u \land t = u \]

3.11 Union and Intersection

Binary union and distributed and binary intersection are defined.

3.11.1 Binary Union

HOL Constant

\[ \cup_g : \text{GS} \to \text{GS} \to \text{GS} \]

\[ \forall s \ t \ e \bullet e \in_g (s \cup_g t) \iff e \in_g s \lor e \in_g t \]

\[ \subseteq_g \cup_g \text{thm} = \vdash \forall A \ B \bullet A \subseteq_g A \cup_g B \land B \subseteq_g A \cup_g B \]

\[ \cup_g \subseteq_g \text{thm1} = \vdash \forall A \ B \ C \bullet A \subseteq_g C \land B \subseteq_g C \Rightarrow A \cup_g B \subseteq_g C \]

\[ \cup_g \subseteq_g \text{thm2} = \vdash \forall A \ B \ C \ D \bullet A \subseteq_g C \land B \subseteq_g D \Rightarrow A \cup_g B \subseteq_g C \cup_g D \]

\[ \cup_g \emptyset \text{clauses} = \vdash \forall A \bullet A \cup_g \emptyset = A \land \emptyset \cup_g A = A \]

3.11.2 Galaxy Closure

\[ G_{\text{Close} \cup_g} = \vdash \forall g \bullet \text{galaxy } g \Rightarrow \forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow s \cup_g t \in_g g \]

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

\[ \cap_g : \text{GS} \to \text{GS} \]

\[ \forall s \bullet \cap_g s = \text{Sep} (\bigcup_g s) \ (\forall x \bullet \forall t \bullet t \in_g s \Rightarrow x \in_g t) \]

\[ \subseteq_g \cap_g \text{thm} = \vdash \forall x \ s \ e \bullet x \in_g s \Rightarrow (e \in_g \cap_g s) \iff \forall y \bullet y \in_g s \Rightarrow e \in_g y \]

\[ \cap_g \subseteq_g \text{thm} = \vdash \forall A \ B \bullet A \in_g B \Rightarrow \forall C \bullet (\forall D \bullet D \in_g B \Rightarrow C \subseteq_g D) \Rightarrow C \subseteq_g \cap_g B \]

\[ \cap_g \emptyset \text{thm} = \vdash \cap_g \emptyset = \emptyset \]

17
3.11.4 Binary Intersection

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

\[
\text{SML}\]
\begin{verbatim}
declare_infix (240, "\cap_g");
\end{verbatim}

\[
\text{HOL Constant}
\]
\[
\forall s \bullet s \cap_g t = \text{Sep} s (\lambda x \bullet x \in_g t)
\]

3.11.5 Galaxy Closure

\[
G\text{Close}_\cap_g = \neg \forall g \bullet \text{galaxy } g \Rightarrow \forall s \bullet s \in_g g \Rightarrow \cap_g s \in_g g
\]

\[
G\text{Close}_\cap_g = \neg \forall g \bullet \text{galaxy } g \Rightarrow \forall s t \bullet s \in_g g \wedge t \in_g g \Rightarrow s \cap_g t \in_g g
\]

\[
\cap_g\text{-thm} = \neg \forall s t e \bullet e \in_g s \cap_g t \Leftrightarrow e \in_g s \wedge e \in_g t
\]

\[
\cap_g\text{-thm} = \neg \forall s t e \bullet e \in_g s \cap_g t \Leftrightarrow e \in_g s \wedge e \in_g t
\]

\[
\in_g\cap_g\text{-thm}1 = \neg \forall A B C \bullet A \cap_g B \subseteq_g A \wedge A \cap_g B \subseteq_g B
\]

\[
\in_g\cap_g\text{-thm}2 = \neg \forall A B C D \bullet A \subseteq_g C \wedge B \subseteq_g C \Rightarrow (A \cap_g B) \subseteq_g (C \cap_g D)
\]

\[
\in_g\cap_g\text{-thm}3 = \neg \forall A B C \bullet C \subseteq_g A \wedge C \subseteq_g B \Rightarrow C \subseteq_g A \cap_g B
\]

3.11.6 Consequences of Well-Foundedness

\[
\text{not}_x\text{-in}_x\text{-thm} = \neg \forall (\exists x \bullet x \in_g x)
\]

3.12 Galaxy Closure Clauses

\[
G\text{Close}_{\text{fc-clauses}2} =
\]
\[
\neg \forall g
\]

\[
\bullet \text{galaxy } g
\]
\[
\Rightarrow (\forall s t \bullet s \in_g g \wedge t \in_g g \Rightarrow \text{Pair}_g s t \in_g g)
\]
\[
\wedge (\forall s \bullet s \in_g g \Rightarrow \text{Sing}_g s \in_g g)
\]
\[
\wedge (\forall s t \bullet s \in_g g \wedge t \in_g g \Rightarrow s \cup_g t \in_g g)
\]
\[
\wedge (\forall s t \bullet s \in_g g \Rightarrow \cap_g s \in_g g)
\]
\[
\wedge (\forall s t \bullet s \in_g g \wedge t \in_g g \Rightarrow s \cap_g t \in_g g)
\]

\[
tc\text{-clauses} = \neg \forall s \bullet s \in_g^+ \text{Sing}_g s
\]
\[
\wedge s \in_g^+ \text{Pair}_g s t
\]
\[
\wedge \forall t \bullet t \in_g^+ \text{Pair}_g s t
\]

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

```
vartgst_ax_thms\[\begin{array}{l}
\emptyset\mbox{spec},
gt\mbox{spec }'\mathbb{P}g',
gt\mbox{spec }'\bigcup g',
Imagep\mbox{spec},
Pair\!g\mbox{-eq thm},
gt\mbox{spec }'Pair\!g',
Sing\!g\mbox{-eq thm},
Sing\mbox{thm},
Pair\!g\mbox{-Sing eq thm},
Sing\!Pair\!g\mbox{-eq thm},
Sep\mbox{thm},
\cup g\mbox{-thm},
\cap g\mbox{-thm}
\end{array}\]
```

```
vartgst_opext_clauses =
\begin{array}{l}
\text{(all}\forall\mbox{-intro}
o \text{list}\wedge\mbox{-intro}
o (\text{map all}\forall\mbox{-elim})
gst\mbox{ax thms};
save\mbox{thm }"\text{gst_opext clauses"},\text{gst_opext clauses};
\end{array}
```

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.
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:
3.13.6 Examples

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

\[
\begin{align*}
\text{val } & \text{gst\_ax\_rule } = \\
& (\text{merge\_pcs\_rule }1 \\
& \text{["basic\_hol", "gst\_ax"]} \\
& \text{prove\_rule }) [];
\end{align*}
\]

\[
\begin{align*}
\text{val } & \text{gst\_ax\_conv } = \\
& \text{MERGE\_PCS\_C1} \\
& \text{["basic\_hol", "gst\_ax"]} \\
& \text{prove\_conv};
\end{align*}
\]

\[
\begin{align*}
\text{val } & \text{gst\_ax\_tac } = \\
& \text{conv\_tac o gst\_ax\_conv};
\end{align*}
\]

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.

\[
\begin{align*}
\text{open\_theory } & \text{"gst\_ax\"}; \\
\text{force\_new\_theory } & \text{"gst\_fun\"};
\end{align*}
\]

\[
\begin{align*}
\text{force\_new\_pc } & \text{"gst\_fun\"}; \\
\text{merge\_pcs } & \text{["savedthm\_cs\_3\_proof\"] } \text{"gst\_fun\"}; \\
\text{set\_merge\_pcs } & \text{["basic\_hol", "gst\_ax", "gst\_fun"]};
\end{align*}
\]

4.1 Ordered Pairs

\[
\begin{align*}
\text{declare\_infix } (240,"\rightarrow\_g");
\end{align*}
\]
I first attempted to define ordered pairs in a more abstract way than by explicit use of the Wiener-Kuratovski representation, but this gave 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**

$$\forall s \; t \cdot (s \mapsto_g t) = \text{Pair}_g (\text{Sing} \; s) \; (\text{Pair}_g \; s \; t)$$

4.1.1 Projections

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

**HOL Constant**

$$\text{fst} \; \text{snd} : \text{GS} \rightarrow \text{GS}$$

$$\forall s \; t \cdot \text{fst}(s \mapsto_g t) = s \wedge \text{snd}(s \mapsto_g t) = t$$

$$\forall x \; y \cdot \text{tc} \; x \; \text{y} \mapsto_g x \wedge \text{tc} \; \text{y} \mapsto_g y$$

4.1.2 MkPair and MkTriple

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

\[ \text{MkTriple}_g : GS \times GS \times GS \rightarrow GS \]

\[ \forall t \cdot \text{MkTriple}_g t = (\text{Fst} t) \mapsto_g (\text{MkPair}_g (\text{Snd} t)) \]

4.2 Relations

HOL Constant

\[ \text{rel} : GS \rightarrow BOOL \]

\[ \forall x \cdot \text{rel} x \iff \forall y \cdot y \in g x \Rightarrow \exists s \cdot s = s \mapsto_g t \]

| \text{rel}_\emptyset\text{-thm} = \vdash \text{rel} \emptyset |

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

HOL Constant

\[ \text{dom} : GS \rightarrow GS \]

\[ \forall x \cdot \text{dom} x = \text{Sep} (Gx x) (\lambda w \cdot \exists v \cdot w \mapsto_g v \in g x) \]

\[ \text{dom}_\emptyset\text{-thm} = \vdash \text{dom} \emptyset = \emptyset \]

\[ \text{dom}_{Gx}\text{-thm} = \vdash \forall r \cdot \text{dom} r \in_g Gx r \]

\[ \text{GCLOSE}_\text{dom}\text{-thm} = \vdash \forall g \cdot \text{galaxy} g \Rightarrow (\forall r \cdot r \in_g g \Rightarrow \text{dom} r \in_g g) \]

HOL Constant

\[ \text{ran} : GS \rightarrow GS \]

\[ \forall x \cdot \text{ran} x = \text{Sep} (Gx x) (\lambda w \cdot \exists v \cdot v \mapsto_g w \in g x) \]

\[ \text{ran}_\emptyset\text{-thm} = \vdash \text{ran} \emptyset = \emptyset \]

\[ \text{ran}_{\text{Gx}}\text{-thm} = \vdash \forall r \cdot y \in g r \Rightarrow (\exists x \cdot x \mapsto_g y \in g r) \]

\[ \text{GCLOSE}_{\text{ran}}\text{-thm} = \vdash \forall g \cdot \text{galaxy} g \Rightarrow (\forall r \cdot r \in_g g \Rightarrow \text{ran} r \in_g g) \]

\[ \text{tc}\text{-ran}\text{-thm} = \vdash \forall x \cdot y \in_g x \Rightarrow x \in_g y \]

HOL Constant

\[ \text{field} : GS \rightarrow GS \]

\[ \forall s \cdot e \in_g (\text{field} s) \iff e \in_g (\text{dom} s) \lor e \in_g (\text{ran} s) \]

| \text{field}_\emptyset\text{-thm} = \vdash \text{field} \emptyset = \emptyset |
4.3 Domain and Range Restrictions

SML

\[
\begin{align*}
declare\_infix (300, "\triangleleft_g");
declare\_infix (300, "\triangleright_g");
declare\_infix (300, "\triangleleft_g");
declare\_infix (300, "\triangleright_g");
\end{align*}
\]

HOL Constant

\[
\begin{align*}
\triangleleft_g : GS \rightarrow GS \rightarrow GS \\
\forall s \ r \bullet s \triangleleft_g r = Sep r (\lambda p \bullet \text{fst } p \in_g s)
\end{align*}
\]

HOL Constant

\[
\begin{align*}
\triangleright_g : GS \rightarrow GS \rightarrow GS \\
\forall s \ r \bullet r \triangleright_g s = Sep r (\lambda p \bullet \text{snd } p \in_g s)
\end{align*}
\]

HOL Constant

\[
\begin{align*}
\triangleleft_g : GS \rightarrow GS \rightarrow GS \\
\forall s \ r \bullet s \triangleleft_g r = Sep r (\lambda p \bullet \text{fst } p \in_g s)
\end{align*}
\]

HOL Constant

\[
\begin{align*}
\triangleright_g : GS \rightarrow GS \rightarrow GS \\
\forall s \ r \bullet r \triangleright_g s = Sep r (\lambda p \bullet \text{snd } p \in_g s)
\end{align*}
\]

SML

\[
\begin{align*}
declare\_alias ("\triangleleft", "\triangleleft_g");
declare\_alias ("\triangleright", "\triangleright_g");
declare\_alias ("\triangleleft", "\triangleleft_g");
declare\_alias ("\triangleright", "\triangleright_g");
\end{align*}
\]

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.
\[
\forall r \bullet \text{Rel2DepType}_g r = \text{Sep} \\
\quad (Gx r) \\
\quad (\lambda e \bullet \exists i : GS \bullet e = i \mapsto_g t) \\
\quad \land i \in_g \text{dom } r \\
\quad \land (\forall j \bullet j \in_g t \iff i \mapsto_g j \in_g r))
\]

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:

\[
\forall t \bullet \text{DepSum}_g t = \text{Sep} \\
\quad (Gx t) \\
\quad (\lambda e \bullet \exists i : t^2 \bullet e = i \mapsto_g v) \\
\quad \land v \in_g t^2 \\
\quad \land (\forall j \bullet j \in_g t^2 \in_g t)
\]

### 4.5 Dependent Sums and Cartesian Products

SML

\[
\text{declare\_binder } "\Sigma_g";
\]

HOL Constant

\[
\Sigma_g : (GS \rightarrow GS) \rightarrow GS \rightarrow GS
\]

\[
\forall f \bullet \Sigma_g f s = \bigcup_g (Imagep (\lambda x \bullet (\lambda e \bullet Imagep (\lambda e \bullet e \mapsto_g x) (f e))))
\]

SML

\[
\text{declare\_infix}(240,"\times_g");
\]
\[ g : GS \to GS \to GS \]

\[
\forall s \; t \bullet s \times_g t = \bigcup_{g} (Imagep (\lambda s e \bullet (Imagep (\lambda t e \bullet se \mapsto_g te) t))) s
\]

\[
f \mapsto_{g_{\text{thm}}} = \\
\vdash \forall s \; t \; p \bullet p \in_g s \times_g t \Rightarrow \text{fst } p \mapsto_g \text{snd } p = p
\]

\[
v \in_g x_{\text{thm}} = \\
\vdash \forall p \; s \; t \bullet p \in_g s \times_g t \Rightarrow \text{fst } p \in_g s \land \text{snd } p \in_g t
\]

\[
\mapsto_{g \in_g x_{\text{thm}}} = \\
\vdash \forall l \; r \; s \; t \bullet l \mapsto_g r \in_g s \times_g t \iff (l \in_g s \land r \in_g t)
\]

### 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**

```sml
declare infix(240,"\mapsto_g");
```

### 4.5.2 Another Pair-Projection Inverse Theorem

Couched in terms of membership of relation spaces.
4.5.3 Member of Relation Theorem

SML

```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 (REPEAT
  (asm_fc_tac[f→_g_thm]));

val f→_g_thm1 =
save_pop_thm "f→_g_thm1";
```

4.5.4 Relational Composition

SML

```sml
declare infix 250, "ο_g";
```

HOL Constant

\[ \exists g \cdot f \circ g = \]

\[ \begin{align*}
  \text{Imagep} & \cdot (\lambda p \cdot (\text{fst}(\text{fst} p) \leftrightarrow g \text{snd}(\text{snd} p))) \\
  (\text{Sep} (g \times_g f) \lambda p \cdot \exists q \ r \ s \cdot p = (q \leftrightarrow_g r) \leftrightarrow_g (r \leftrightarrow_g s))
\end{align*} \]
\[ o_g \text{-thm} = \]
\[ \vdash \forall f \; g \bullet x \in_g f \; o_g \; g \iff \]
\[ \exists q \; r \bullet q \mapsto_g r \in_g g \land x = q \mapsto_g s \]

\[ o_g \text{-thm2} = \]
\[ \vdash \forall f \; g \; x \bullet x \mapsto_g y \in_g f \; o_g \; g \]
\[ \iff (\exists z \bullet x \mapsto_g z \in_g g \land z \mapsto_g y \in_g f) \]

\[ o_g \text{-associative-thm} = \]
\[ \vdash \forall f \; g \; h \bullet (f \; o_g \; g) \; o_g \; h = f \; o_g \; g \; o_g \; h \]

\[ o_g \text{-rel-thm} = \]
\[ \vdash \forall r \; s \bullet \text{rel} \; r \land \text{rel} \; s \Rightarrow \text{rel} \; (r \; o_g \; s) \]

### 4.5.5 Relation Subset of Cartesian Product

\[ \text{rel-sub-cp-thm} = \]
\[ \vdash \forall x \bullet \text{rel} \; x \iff (\exists s \; t \bullet x \subseteq_g s \times_g t) \]

### 4.6 Functions

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

#### 4.6.1 fun

**HOL Constant**

\[ \text{fun} : \text{GS} \rightarrow \text{BOOL} \]

\[ \forall x \bullet \text{fun} \; x \iff \text{rel} \; x \land \]
\[ \forall s \; t \bullet s \mapsto_g u \in_g x \land \]
\[ \forall s \bullet s \mapsto_g t \in_g x \Rightarrow u = t \]

#### 4.6.2 lemmas

\[ \text{fun-} \emptyset_g \text{-thm} = \]
\[ \vdash \text{fun} \; \emptyset_g \]

\[ o_g \text{-fun-thm} = \]
\[ \vdash \forall f \; g \bullet \text{fun} \; f \land \text{fun} \; g \Rightarrow \text{fun} \; (f \; o_g \; g) \]

\[ \text{ran}_o \text{-} o_g \text{-thm} = \]
\[ \vdash \forall f \; g \bullet \text{ran} \; (f \; o_g \; g) \subseteq_g \text{ran} \; f \]

\[ \text{dom}_o \text{-} o_g \text{-thm} = \]
\[ \vdash \forall f \; g \bullet \text{dom} \; (f \; o_g \; g) \subseteq_g \text{dom} \; g \]

\[ \text{dom}_o \text{-} o_g \text{-thm2} = \]
\[ \vdash \forall f \; g \bullet \text{ran} \; g \subseteq_g \text{dom} \; f \Rightarrow \text{dom} \; (f \; o_g \; g) = \text{dom} \; g \]
4.6.3 Partial Function Space

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

\[
\forall s \bullet s \rightarrow g t = \text{Sep} (s \leftrightarrow g t) \text{ fun}
\]

4.6.4 Partial Function Space Non-Empty

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

\[
\begin{align*}
&\forall s \bullet \emptyset \in_g s \rightarrow_g t) ; \\
&\text{a (prove_tac[}
\quad &\text{get_spec } \forall s \rightarrow_g t) ; \\
&\quad &\text{fun}_g(); \\
&\quad &\text{val } \emptyset \in_g \rightarrow_g \text{thm} = \\
&\quad &\text{save_pop_thm } "\emptyset \in_g \rightarrow_g \text{thm}";
\end{align*}
\]

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

\[
\begin{align*}
&\forall s \bullet \exists f \bullet f \in_g s \rightarrow_g t) ; \\
&\text{a (REPEAT strip_tac}
\quad &\text{THEN } \exists\text{ _ _ _ _ _ _} ; \\
&\quad &\text{THEN} \\
&\quad &\text{rewrite_tac } [\emptyset \in_g \rightarrow_g \text{thm}]; \\
&\quad &\text{val } \exists \rightarrow_g \text{thm} = \\
&\quad &\text{save_pop_thm } "\exists \rightarrow_g \text{thm}";
\end{align*}
\]

4.6.5 Function Space

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

```sml
set_goal([], "\s \cdot \emptyset \in \emptyset \to \emptyset t");
```

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

```
∃\to \emptyset thm =
∀\t \cdot (∃ x \cdot x \in \emptyset t) \Rightarrow (∃ f \cdot f \in \emptyset \to \emptyset t)
```

HOL Constant

```
→ \emptyset_closed : GS → BOOL
```

```
∀s → \emptyset_closed s ⇔ ∀d c d \in \emptyset s ∧ c \in \emptyset s ⇒ d → \emptyset c \in \emptyset s
```

4.7 Functional Abstraction

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

4.7.1 Abstraction

Because of the closeness to lambda abstraction \(\lambda_g\) is used as the name of a new binder for set theoretic functional abstraction.

SML

```sml
declare_binder "\lambda_g";
```

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

```
$\lambda_g : (GS → GS) → GS → GS
```

```
∀f s \cdot \$\lambda_g f s = Sep (s \times \emptyset (Imagep f s)) (\lambda p \cdot 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

\[ \text{declare infix (250, "} g ");} \]

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

$g : GS \rightarrow GS \rightarrow GS$

\[ \forall f \bullet f \circ g x = \]

\[ \text{if } \exists y \bullet x \mapsto y \in g f \]

\[ \text{then } \epsilon y \bullet x \mapsto y \in g f \]

\[ \text{else } f \]

\[ \text{app_thm1} = \]

\[ \text{app_thm2} = \]

\[ \text{app_thm3} = \]

\[ o_{g \cdot g \cdot \text{thm}} = \]

\[ (f \circ g) g x = f g g g x \]
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

```ml
set_goal([],
  \forall f s t u \cdot f \in A \Rightarrow
  s \in A \Rightarrow
  t \in A \Rightarrow
  u \in A \Rightarrow
  f \in A \Rightarrow
  t \in A).

a (prove_tac[
  get_spec \$->->A,
  get_spec \$ dom]);

a (all_fc_tac [app_thm2] THEN asm_rewrite_tac[]);

a (all_fc_tac [f->gs_thm])

a (all_fc_tac [f\in A \Rightarrow g->A]);

a (POP_ASM T ante_tac THEN asm_rewrite_tac[]);

val g->A_thm = save_pop_thm "$g->A_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.)

```
\in A \Rightarrow \in A \Rightarrow g->A_thm = \\
\forall f s t u \cdot f \in g \Rightarrow s \in g \Rightarrow t \Rightarrow f \in g \Rightarrow dom f \Rightarrow g t
```

4.9 The Identity Function

4.9.1 specification

HOL Constant

```
id : GS \to GS
```

```
\forall s \cdot id s = Sep

(s \times g s)

\lambda x \cdot \text{fst } x = \text{snd } x
```
4.9.2 lemmas

\[ id\_thm1 = \]
\[ \vdash \forall s \cdot x \in_g id\ s \]
\[ \iff \exists y \cdot y \in_g s \land x = y \mapsto_g y \]

\[ id\_ap\_thm = \]
\[ \vdash \forall s \cdot x \in_g s \]
\[ \Rightarrow (id\ s)\ y \cdot x = x \]

\[ id\in_g \mapsto_g \_thm1 = \]
\[ \vdash \forall s\ t\ u\cdot s \subseteq_g t \cap_g u \]
\[ \Rightarrow id\ s \in_g t \mapsto_g u \]

\[ id\in_g \mapsto_g \_thm2 = \]
\[ \vdash \forall s\ t\ u\cdot s \subseteq_g t \]
\[ \Rightarrow id\ s \in_g t \mapsto_g t \]

\[ id\_clauses = \]
\[ \vdash \forall s\cdot \text{rel}(id\ s) \land \text{fun}(id\ s) \]
\[ \land \text{dom}(id\ s) = s \land \text{ran}(id\ s) = s \]

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**

\[ \text{declare infix } (250,"\oplus_g") ; \]

**HOL Constant**

\[ \oplus_g : GS \rightarrow GS \rightarrow GS \]

\[ \forall s\ t\cdot s \oplus_g t = \text{Sep}\ (s \cup_g t) \]
\[ \lambda x\cdot \text{if}\ \text{fst}\ x \in_g \text{dom}\ t\ \text{then}\ x \in_g t\ \text{else}\ x \in_g s \]

\[ \in_g \oplus_g \_thm = \]
\[ \vdash \forall s\ t\ x\cdot x \in_g s \oplus_g t = (\text{if}\ \text{fst}\ x \in_g \text{dom}\ t\ \text{then}\ x \in_g t\ \text{else}\ x \in_g s) \]

\[ \mapsto_g \in_g \oplus_g \_thm = \]
\[ \vdash \forall s\ t\ x\ y \]
\[ \cdot x \mapsto_g y \in_g s \oplus_g t = (x \mapsto_g y \in_g t \lor \neg x \in_g \text{dom}\ t \land x \mapsto_g y \in_g s) \]

\[ \oplus_g \_\text{rel}_\text{thm} = \]
\[ \vdash \forall s\ t\cdot \text{rel}\ s \land \text{rel}\ t \Rightarrow \text{rel}\ (s \oplus_g t) \]

33
\[ @s \text{ fun } s \land \text{ fun } t \Rightarrow \text{ fun } (s \oplus g \, t) \]

4.11 Proof Contexts

Finalisation of a proof context.

4.11.1 Proof Context

SML

```sml
add_pc_thms "gst-fun" [field_\emptyset_g-thm,
fun_\emptyset_g-thm,
\emptyset_g^g_{\rightarrow 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 correspond 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

```sml
open theory "gst-ax";
force_new_theory "gst-ord";
(* new_parent "wf-recp"; *)
force_new_pc "gst-ord";
merge_pcs ["savedthm_cs.3-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

```haskell
connected : GS → BOOL

∀s :GS • connected s ⇔
∀t u :GS • t ∈_gs s ∧ u ∈_gs s ⇒ t ∈_gs u ∨ t = u ∨ u ∈_gs t
```

HOL Constant

```haskell
ordinal : GS → BOOL

∀s :GS • ordinal s ⇔ transitive s ∧ connected s
```

We now introduce infix ordering relations over ordinals.

SML

```sml
declare_infix(240,"<_o");
declare_infix(240,">_o");
```
The following definition gives the successor function over the ordinals (this appears later in Drake).

\[
\forall x: GS \ni x = x \cup_g (Sing \ x)
\]
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.

\[
\text{set} \text{goal}([], \forall x: \text{GS} \cdot \text{transitive } x \Rightarrow \text{transitive } (\text{suc}_o x))
\]

\[
\text{set} \text{goal}([], \forall x: \text{GS} \cdot \text{connected } x \Rightarrow \text{connected } (\text{suc}_o x))
\]

These together enable us to prove:

\[
\text{set} \text{goal}([], \forall x: \text{GS} \cdot \text{ordinal } x \Rightarrow \text{ordinal } (\text{suc}_o x))
\]

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

\[
a \text{ (rewrite_tac }\text{get spec }\forall \text{ ordinal})
\]

\[
\text{THEN REPEAT strip_tac}
\]

\[
\text{THEN fc_tac }[\text{trans_suc_trans, conn_suc_conn}]
\]

\[
\text{val ord_suc_ord_thm }= \text{save_pop_thm }"\text{ord_suc_ord_thm}"
\]

5.2.3 The Ordinal Zero is not a successor

\[
\emptyset \text{. not suc}_o \text{. thm }=
\]

\[
\vdash \neg (\exists \alpha \bullet \text{suc}_o \alpha = \emptyset)
\]

\[
\text{not_in_suc}_o \text{. thm }=
\]

\[
\vdash \forall \alpha \bullet \neg \alpha = \text{suc}_o \alpha
\]

\[
\text{leo_suc} \text{. thm }=
\]

\[
\vdash \forall \alpha \bullet \text{ordinal } \alpha \Rightarrow \alpha \preceq_o \text{suc}_o \alpha
\]

\[
\text{lo_suc} \text{. thm }=
\]

\[
\vdash \forall \alpha \bullet \text{ordinal } \alpha \Rightarrow \alpha <_o \text{suc}_o \alpha
\]
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([], '\forall x:GS\bullet connected x \Rightarrow \forall y:GS\bullet y \subseteq x \Rightarrow 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', '\subseteq_g'])) THEN REPEAT_N 7 strip_tac;
```

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

SML

```
set_goal([], '\forall x:GS\bullet ordinal x \Rightarrow \forall y:GS\bullet y \in_g x \Rightarrow 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 (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([], '\forall x:GS\bullet ordinal x \Rightarrow \forall y:GS\bullet y \in_g x \Rightarrow 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.
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
\[
\text{set_goal([], } \forall x:GS \bullet ordinal \ x \Rightarrow \forall y:GS \ y \in_x x \Rightarrow transitive \ y)\];

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

SML
\[
\text{set_goal([], } \forall x:GS \bullet ordinal \ x \Rightarrow \forall y:GS \ y \in_x x \Rightarrow ordinal \ y)\];

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
\[
\text{set_goal([], } \forall x \ y:GS \bullet transitive \ x \land transitive \ y \Rightarrow transitive \ (x \cap_g y)\]);
\]

SML
\[
\text{set_goal([], } \forall x \ y:GS \bullet transitive \ x \land transitive \ y \Rightarrow transitive \ (x \cup_g y)\]);
\]

SML
\[
\text{set_goal([], } \forall x \ y:GS \bullet connected \ x \land connected \ y \Rightarrow connected \ (x \cap_g y)\]);
\]

SML
\[
\text{set_goal([], } \forall x \ y:GS \bullet ordinal \ x \land ordinal \ y \Rightarrow ordinal \ (x \cap_g y)\]);
\]

SML
\[
\text{set_goal([], } \forall x \ y:GS \bullet ordinal \ x \land ordinal \ y \land x \leq_g y \land \neg x = y \Rightarrow x \in_g y)\];
\]

\textbf{trich_for_ords.thm =}
\[
\vdash \ \forall x \ y:gs \bullet ordinal \ x \land ordinal \ y \Rightarrow x <_o y \lor x = y \lor y <_o x
\]

\textbf{sub_leo.thm =}
\[
\vdash \ \forall x \ y:gs \bullet ordinal \ x \land ordinal \ y \Rightarrow (x \leq_g y \iff x \leq_o y)
\]

\textbf{sub_leo.thm1 =}
\[
\vdash \ \forall x \ y:gs \bullet ordinal \ x \land ordinal \ y \land x \leq_g y \Rightarrow x \leq_o y
\]
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 choosing longer names.

HOL Constant

\texttt{successor} : GS \to BOOL

\forall s : GS \bullet \text{successor } s \leftrightarrow \exists t \bullet \text{ordinal } t \land s = suc_o t

HOL Constant

\texttt{limit\_ordinal} : GS \to BOOL

\forall s : GS \bullet \text{limit\_ordinal } s \leftrightarrow \text{ordinal } s \land \neg \text{successor } s \land s = \emptyset_g

5.5 Theorem 2.7

Induction theorems over ordinals.

5.5.1 Successors are Ordinals

SML

\texttt{set\_goal}([], \forall x:GS \bullet \text{successor } x \Rightarrow \text{ordinal } x \Rightarrow);\n\texttt{a (rewrite\_tac[get\_spec \text{"successor"]})}\n\texttt{THEN \hspace{1em} REPEAT strip\_tac}\n\texttt{\hspace{1em} THEN fc\_tac [ord\_suc\_ord\_thm]}\n\texttt{\hspace{1em} THEN asm\_rewrite\_tac[]}\n\texttt{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

\texttt{ordinal\_kind\_thm} =

\forall n \bullet \text{ordinal } n \Rightarrow n = \emptyset_g \lor \text{successor } n \lor \text{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.

\[
\text{SML}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

\[
\text{HOL Constant}
\]

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.

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.

\[\text{respect\_lemma} = \]
\[\vdash (\lambda f \cdot \bigcup_g (\text{Imagep} \ (\text{suc}_o \circ f) \ x)) \text{ respects } \in_g\]

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

HOL Constant
\[\text{rank} : GS \rightarrow GS\]
\[\forall x \cdot \text{rank } x = \bigcup_g (\text{Imagep} \ (\text{suc}_o \circ \text{rank}) \ x)\]

5.8 Ordinal Arithmetic

5.8.1 Addition

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

\[\text{plus\_o\_respect\_lemma} = \]
\[\vdash \forall x \cdot (\lambda x.+: y \cdot \text{if } y = \emptyset \text{ then } x \text{ else } \text{ssup} (\text{Imagep} \ x.+: y)) \text{ respects } \in_g\]

HOL Constant
\[\text{+_o} : GS \rightarrow GS \rightarrow GS\]
\[\forall \alpha \beta \cdot \alpha \text{+_o} \beta = \text{if } \beta = \emptyset \text{ then } \alpha \text{ else } \text{ssup} (\text{Imagep} \ (\text{+_o} \ \alpha) \ \beta)\]

\[\text{ord\_plus\_thm} = \]
\[\vdash \forall \alpha \beta \cdot \text{ordinal } \alpha \land \text{ordinal } \beta \Rightarrow \text{ordinal } (\alpha \text{+_o} \beta)\]

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:

\[\text{--o\_lemma} = \]
\[\forall \alpha \beta \cdot \alpha \leq_o \beta \Rightarrow \alpha \text{+_o} (\beta \text{--o} \alpha) = \beta\]

HOL Constant
\[\text{--o} : GS \rightarrow GS \rightarrow GS\]
\[T\]
5.9 Proof Context

In this section we define a proof context for ordinals.

5.9.1 Proof Context

SML

```sml
add_pc_thms "gst-ord" ([]);
set_mergepcs ["basic_hol", "gst-ax", "gst-ord"];
commit_pc "gst-ord";
```

6 Natural Numbers

SML

```sml
open theory "gst-ord";
force_new_theory "gst-nat";
force_new_pc "gst-nat";
merge_pcs ["savedthm.cs.3.proof"] "gst-nat";
set_mergepcs ["basic_hol", "gst-ax", "gst-ord", "gst-nat"];
```

HOL Constant

```text
natural_number : GS → BOOL

∀s : GS• natural_number s ⇔ s = ∅g ∨ (successor s ∧ ∀t• t ∈ g s ⇒ t = ∅g ∨ 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 <ₚn (less than over the natural numbers defined in galactic set theory).

SML

```sml
declare_infix (250,"<ₚn");
```

HOL Constant

```text
$<ₚn : GS → GS → BOOL

∀x y:GS• x <ₚn y ⇔ natural_number x ∧ natural_number y ∧ x ∈ₚ 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.
This allows us to do well-founded induction over the natural number which the way I have implemented it is \textit{“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.

I've tried using that principle and it too has disadvantages. Because \textless{} \textit{gn} \textit{is} used the induction hypothesis is more awkward to use (weaker) than it would have been if \texttt{P} \textit{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.
6.1 Theorem 2.8

The set of natural numbers.

6.1.1 Natural Numbers are Ordinals

SML

```
set_goal ([], "\forall n\cdot natural_number n \Rightarrow ordinal n");
a (rewrite_tac [get_spec "natural_number", get_spec "successor"]);
a (REPEAT strip_tac THEN_TRY asm_rewrite_tac[ordinal_\empty_set]);
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 ([], "\forall n\cdot natural_number n \land \forall m\cdot m \in n \Rightarrow 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 ([], "\forall n\cdot natural_number n \Rightarrow \neg 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 ([], "\forall n\cdot natural_number n \Rightarrow successor n \lor n = 0");
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 ([], "\forall m\cdot natural_number n \land m \in n \Rightarrow \neg limit_ordinal m");
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

```
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_fe_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_fe_tac[]);
val mem_nat_nat_thm = save_pop_thm "mem_nat_nat_thm";
```

6.1.7 Natural Numbers are in the Smallest Galaxy

```
set_goal([], ∀n ▷ natural_number n ⇒ n ∈ g G x ⊊ 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 < g n ▷
   THEN1 asm_rewrite_tac [get_spec ▷ $< g n, get_spec ▷ suc_o]);
(* *** Goal "1.1" *** *)
a (lemma_tac ▷ t ∈ g n ▷
   THEN1 asm_rewrite_tac [get_spec ▷ suc_o]);
a (all_fc_tac [mem_nat_nat_thm]);
(* *** Goal "1.2" *** *)
a (asm_tac (∀ elim ▷ ⊊ g ⊊ galaxy_G x));
a (asm_rewrite_tac[]);
a (REPEAT (all_asm_fe_tac [G_closeSuc]));
(* *** 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.

```
set_goal([], ∃ w ▷ ∀ z ▷ z ∈ g w ⇔ natural_number z ▷);
a (∃ tac ▷ Sep (G x ⊊ 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]);
```
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

\[
\text{Nat}_g : \mathbb{N} \rightarrow GS
\]

\[
\begin{align*}
\text{Nat}_g 0 &= \emptyset_g \\
\forall n \cdot \text{Nat}_g (n+1) &= \text{suc}_o (\text{Nat}_g n)
\end{align*}
\]

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

\[
\text{ord}_\text{nat}_\text{thm2} =
\]

\[
\vdash \forall n \cdot \text{ordinal} (\text{Nat}_g n)
\]

\[
\text{not}_\text{suc}_\text{nat}_\text{zero}_\text{thm} =
\]

\[
\vdash \forall n \cdot \neg \text{suc}_o (\text{Nat}_g n) = \emptyset_g
\]

\[
\text{less}_\text{sum}_\text{thm} =
\]

\[
\vdash \forall x y \cdot x \leq y \Rightarrow (\exists z \cdot x + z = y)
\]

\[
\text{nat}_\text{g}_\text{mono}_\text{thm} =
\]

\[
\vdash \forall x y \cdot \text{Nat}_g x \leq_o \text{Nat}_g (x + y)
\]

\[
\text{nat}_\text{g}_\text{one}_\text{_one}_\text{thm} =
\]

\[
\vdash \forall x y \cdot \text{Nat}_g x = \text{Nat}_g y \Rightarrow x = y
\]

\[
\text{nat}_\text{g}_\text{one}_\text{_one}_\text{thm2} =
\]

\[
\vdash \forall x y \cdot \text{Nat}_g x = \text{Nat}_g y \Leftrightarrow x = y
\]

6.3 Proof Context

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

6.3.1 Proof Context

SML

\[
\begin{align*}
\text{add}_\text{pc}_\text{thms} \ "\text{gst}_\text{nat}\" \ ([\text{nat}_\text{g}_\text{one}_\text{_one}_\text{thm2}]); \\
\text{set}_\text{merge}_\text{pcs} \ ["\text{basic}\_\text{hol}\", \"\text{gst}_\text{ax}\", \"\text{gst}_\text{ord}\", \"\text{gst}_\text{nat}\"];
\text{commit}_\text{pc} \ "\text{gst}_\text{nat}\";
\end{align*}
\]
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", "GST1"]
commit_pc "GST1";
merge_pcs ["basic_hol", "GST1"] "GS";
commit_pc "GS";


8 The Theory gst-ax

8.1 Parents

\[ \text{wf\_recp} \quad \text{wf\_relp} \quad \text{U\_orders} \quad \text{rbjmisc} \]

8.2 Children

\[ \text{gst\_ord} \quad \text{gst\_fun} \]

8.3 Constants

\begin{align*}
\forall \in_g & \quad \text{GS} \to \text{GS} \to \text{BOOL} \\
X_g & \quad \text{GS} \to \text{GS} \ P \\
\forall \in_g^+ & \quad \text{GS} \to \text{GS} \to \text{BOOL} \\
\forall \subseteq_g & \quad \text{GS} \to \text{GS} \to \text{BOOL} \\
\subseteq_g \text{-closed} & \quad \text{GS} \to \text{BOOL} \\
P_g & \quad \text{GS} \to \text{GS} \\
\bigcup_g & \quad \text{GS} \to \text{GS} \\
\text{RelIm} & \quad (\text{GS} \to \text{GS} \to \text{BOOL}) \to \text{GS} \to \text{GS} \\
\text{Sep} & \quad \text{GS} \to (\text{GS} \to \text{BOOL}) \to \text{GS} \\
\text{galaxy} & \quad \text{GS} \to \text{BOOL} \\
\text{Gx} & \quad \text{GS} \to \text{GS} \\
\text{transitive} & \quad \text{GS} \to \text{BOOL} \\
\varnothing_g & \quad \text{GS} \\
\text{Imagep} & \quad (\text{GS} \to \text{GS}) \to \text{GS} \to \text{GS} \\
\text{Pair}_g & \quad \text{GS} \to \text{GS} \to \text{GS} \\
\text{Sing} & \quad \text{GS} \to \text{GS} \\
\forall \cup_g & \quad \text{GS} \to \text{GS} \to \text{GS} \\
\cap_g & \quad \text{GS} \to \text{GS} \\
\forall \cap_g & \quad \text{GS} \to \text{GS} \to \text{GS}
\end{align*}

8.4 Types

\[ \text{GS} \]

8.5 Fixity

\text{Right Infix 230:} \\
\begin{align*}
\subseteq_g & \quad \forall_g \quad \forall_g^+ \quad \subseteq_g \\
\text{Right Infix 240:} & \\
\cap_g & \quad \cup_g
\end{align*}
8.6 Axioms

\text{gs\_ext\_axiom} \vdash \forall s \ t \bullet s = t \iff (\forall e \bullet e \in_g s \iff e \in_g t)

\text{gs\_wf\_axiom} \vdash \text{UWellFounded} \ \& \text{g}_s

\text{Ontology\_axiom}
\vdash \forall s
  \bullet \exists g
  \bullet s \in_g g
  \wedge (\forall t
    \bullet t \in_g g
    \Rightarrow t \subseteq_g g
    \wedge (\exists p
      \bullet (\forall w \bullet w \in_g p \iff w \subseteq_g t) \wedge p \in_g g)
    \wedge (\exists u
      \bullet (\forall v
        \bullet v \in_g u \iff (\exists w \bullet v \in_g w \wedge w \in_g t))
        \wedge u \in_g g)
    \wedge (\forall rel
      \bullet \text{ManyOne rel}
      \Rightarrow (\exists r
        \bullet (\forall v
          \bullet v \in_g r
          \iff (\exists w \bullet w \in_g t \wedge rel w v))
          \wedge (r \subseteq_g g \Rightarrow r \in_g g))))

8.7 Definitions

\text{X}_g \vdash \forall s \bullet X_g s = \{t | t \in_g s\}
\text{\&}_g \vdash \exists \text{g}_s = \text{tc} \ \& \text{g}_s
\subseteq_g \vdash \forall s \ t \bullet s \subseteq_g t \iff (\forall e \bullet e \in_g s \Rightarrow e \in_g t)
\text{\&}_g \vdash \forall s \ t \bullet s \subseteq_g t \iff s \subseteq_g t \wedge \sim t \subseteq_g s
\text{\&}_g\text{-closed} \vdash \forall s
  \bullet \subseteq_g\text{-closed} s \iff (\forall f \bullet e \in_g s \wedge f \subseteq_g e \Rightarrow f \in_g s)
\mathbb{P}_g \vdash \forall s \ t \bullet t \in_g \mathbb{P}_g s \iff t \subseteq_g s
\mathbb{U}_g \vdash \forall s \ t \bullet t \in_g \mathbb{U}_g s \iff (\exists e \bullet t \in_g e \wedge e \in_g s)
\text{RelIm} \vdash \forall \ rel \ s \ t
  \bullet \text{ManyOne rel}
  \Rightarrow (t \in_g \text{RelIm} \ rel \ s \iff (\exists e \bullet e \in_g s \wedge \text{rel} e t))
\text{Sep} \vdash \forall s \ p \ e \bullet e \in_g \text{Sep} s \ p \iff e \in_g s \wedge p \ e
\text{galaxy} \vdash \forall s
  \bullet \text{galaxy} s
  \iff (\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 \mathbb{U}_g t \in_g s
    \wedge (\forall rel
      \bullet \text{ManyOne rel}
      \Rightarrow \text{RelIm} \ rel \ t \subseteq_g s
      \Rightarrow \text{RelIm} \ rel \ t \in_g s))
\( Gx \quad \vdash \forall s \ t \quad \text{transitive} \)
\( Gx \quad \iff (\forall g \cdot \text{galaxy } g \land s \in_g g \Rightarrow t \in_g g) \)

\[ \emptyset_g \quad \vdash \forall s \cdot \emptyset \in_g \emptyset_g \]

\( \text{Imagep} \quad \vdash \forall f \ s \ x \cdot x \in_g f \ \text{Imagep } f \ s \iff (\exists e \cdot e \in_g s \land x = f \ e) \)

\( \text{Pair}_g \quad \vdash \forall s \ t \ e \cdot e \in_g \text{Pair}_g \ s \ t \iff e = s \lor e = t \)

\( \text{Sing} \quad \vdash \forall s \ t \ e \cdot e \in_g \text{Sing} \ s \ t = \text{Pair}_g \ s \ t \)

\[ \cap_g \quad \vdash \forall s \quad \text{founded} \]
\[ \cap_g \quad \vdash \bigcap_g s = \text{Sep} (\bigcup_g s) (\lambda x \cdot \forall t \bullet t \in_g s \Rightarrow x \in_g t) \]

\[ \cap_g \quad \vdash \forall s \ t \bullet \cap_g t = \text{Sep} s (\lambda x \bullet x \in_g t) \]

\[ 8.8 \text{ Theorems} \]

\( \text{UWellFounded}_\text{well-founded.thm} \]
\[ \vdash \forall g \bullet \text{UWellFounded } g \iff \text{well-founded } g \iff \]

\( \text{gs_wf_thm1} \quad \vdash \text{well-founded } g \in_g \)

\( \text{gs_wf_thm2} \quad \vdash \text{well-founded } (tc \in_g g) \)

\( \text{gs_wf_min_thm} \]
\[ \vdash \forall s \quad \text{founded} \]

\( \text{gs_wf_min_thm2} \]
\[ \vdash (\exists y \bullet y \in_g x) \quad \Rightarrow (\exists z \bullet z \in_g x \land \neg (\exists v \bullet v \in_g z \land v \in_g x)) \]

\( \text{gs_wf_cases_thm} \quad \vdash \forall s \ y \cdot x \in_g y \Rightarrow x \in_g y \+)

\( \text{gs_wf_trans_thm} \quad \vdash \forall s \ t \ u \cdot s \in_g t \land t \in_g u \Rightarrow s \in_g u \)

\( \text{gs_wf_ind_thm} \quad \vdash \forall p \bullet (\forall x \bullet (\forall y \bullet y \in_g x \Rightarrow p \ y) \Rightarrow p \ x) \Rightarrow (\forall x \bullet p \ x) \)

\( \text{gs_cv_ind_thm} \]
\[ \vdash \forall p \quad \text{founded} \]

\( \text{gs_cv_ind_thm2} \]
\[ \vdash (\forall x \bullet (\forall y \bullet tc \in_g y \ x = \ p \ y) \Rightarrow p \ x) \Rightarrow (\forall x \bullet p \ x) \]

\( \text{wf_l1} \quad \vdash \forall s \bullet \neg x \in_g x \)

\( \text{wf_l2} \quad \vdash \forall y \bullet \neg (x \in_g y \land y \in_g x) \)

\( \text{wf_l3} \quad \vdash \forall x \ y \ z \bullet \neg (x \in_g y \land y \in_g z \land z \in_g x) \)

\( \subseteq_g \text{eq_thm} \quad \vdash \forall A \ B \bullet A = B \iff A \subseteq_g B \land B \subseteq_g A \)

\( \subseteq_g \text{refl_thm} \quad \vdash \forall A \bullet A \subseteq_g A \)

\( \subseteq_g \text{trans_thm} \quad \vdash \forall e \ A \ B \bullet e \in_g A \land A \subseteq_g B \Rightarrow e \in_g B \)

\( \subseteq_g \text{trans_thm2} \quad \vdash \forall A \ B \ C \bullet A \subseteq_g B \land B \subseteq_g C \Rightarrow A \subseteq_g C \)

\( \text{not_psub_thm} \quad \vdash \forall s \bullet \neg x \in_g x \)

\( \text{scPs_thm} \quad \vdash \forall s \bullet s \in_g \mathbb{P}_g s \)

\( \text{scPs_thm} \quad \vdash \forall s \bullet s \in_g \mathbb{P}_g s \)

\( \text{epsUg_thm} \quad \vdash \forall s \ t \bullet t \in_g s \Rightarrow t \subseteq_g \bigcup_g s \)

\( \text{Sep_sub_thm} \quad \vdash \forall s \ p \bullet \text{Sep } s \ p \subseteq_g s \)

\( \text{Sep_sub_thm2} \quad \vdash \forall s \ p \ e \bullet e \in_g \text{Sep } s \ p \Rightarrow e \in_g s \)

\( \text{Sep} \in_\mathbb{F} \text{thm} \quad \vdash \forall s \ p \bullet \text{Sep } s \ p \in_g \mathbb{F}_g s \)

51
\textbf{Sep}_{\subseteq \text{thm}} \quad \vdash \forall s \ t \bullet t \subseteq_{s} s \Rightarrow \text{Sep}_{s} \left( \text{Comb}_{C} \ s_{E_{g}} \ t \right) = t

\textbf{galaxies}_{\exists \text{thm}} \quad \vdash \forall s \bullet \exists g \bullet s \in_{g} g \land \text{galaxy}_{g}

\textbf{t_in}_{Gx_{t}}_{\text{thm}} \quad \vdash \forall t \bullet t \in_{g} Gx_{t}

\textbf{tc\in}_{Gx_{t}}_{\text{thm}} \quad \vdash \forall t \bullet t \in_{g^{+}} Gx_{t}

\textbf{Gx}_{\subseteq_{g}}_{\text{galaxy}} \quad \vdash \forall s \bullet \text{galaxy}_{g} \land s \in_{g} g \Rightarrow Gx_{s} \subseteq_{g} g

\textbf{galaxy}_{Gx} \quad \vdash \forall s \bullet \text{galaxy}_{s} \Rightarrow \text{transitive}_{s}

\textbf{GClose}_{\text{fc}}_{\text{clauses}} \quad \vdash \forall g

\quad \bullet \text{galaxy}_{g}

\quad \Rightarrow \left( \forall s

\quad \bullet s \in_{g} g

\quad \Rightarrow \exists_{g} s \in_{g} g

\quad \land \bigcup_{g} s \in_{g} g

\quad \land \left( \forall p \bullet \text{Sep}_{s} p \in_{g} g \right)

\quad \land \left( \forall t \bullet t \subseteq_{g} s \Rightarrow t \in_{g} g \right) \right)

\textbf{GClose}_{\text{tc\subseteq}_{g}}_{\text{thm}} \quad \vdash \forall s \bullet \text{galaxy}_{g} \Rightarrow s \in_{g^{+}} g \Rightarrow s \in_{g} g

\textbf{Gx}_{\text{mono}}_{\text{thm}} \quad \vdash \forall s \bullet s \subseteq_{g} t \Rightarrow Gx_{s} \subseteq_{g} Gx_{t}

\textbf{Gx}_{\text{mono}}_{\text{thm}2} \quad \vdash \forall s \bullet s \subseteq_{g} t \Rightarrow Gx_{s} \subseteq_{g} Gx_{t}

\textbf{Gx}_{\text{trans}}_{\text{thm}} \quad \vdash \forall s \bullet \text{transitive}_{(Gx_{s})}

\textbf{t_sub}_{Gx_{t}}_{\text{thm}} \quad \vdash \forall t \bullet t \subseteq_{g} Gx_{t}

\textbf{Gx}_{\text{mono}}_{\text{thm}3} \quad \vdash \forall s \bullet s \subseteq_{g} t \Rightarrow s \subseteq_{g} Gx_{t}

\textbf{Gx}_{\text{mono}}_{\text{thm}4} \quad \vdash \forall s \bullet s \subseteq_{g} t \Rightarrow s \subseteq_{g} Gx_{t}

\textbf{Gx}_{\text{trans}}_{\text{thm}2} \quad \vdash \forall s \bullet s \subseteq_{g} t \Rightarrow s \subseteq_{g} Gx_{t}

\textbf{Gx}_{\text{trans}}_{\text{thm}3} \quad \vdash \forall s \bullet s \subseteq_{g} t \Rightarrow s \subseteq_{g} Gx_{t}

\textbf{G\emptyset_{g}C} \quad \vdash \forall g \bullet \text{galaxy}_{g} \Rightarrow \emptyset_{g} \in_{g} g

\textbf{\emptyset_{g}}_{\subseteq_{g}}_{\text{thm}} \quad \vdash \forall s \bullet \emptyset_{g} \subseteq_{g} s

\textbf{\emptyset_{g}}_{\exists_{g}}_{\text{galaxy}}_{\text{thm}} \quad \vdash \forall s \bullet \text{galaxy}_{s} \Rightarrow \emptyset_{g} \in_{g} x

\textbf{\emptyset_{g}}_{\exists_{g}}_{\text{Gx}}_{\text{thm}} \quad \vdash \forall x \bullet \text{galaxy}_{s} \Rightarrow \emptyset_{g} \in_{g} x

\textbf{GImage}_{p_{C}}_{\text{thm}} \quad \vdash \forall g

\quad \bullet \text{galaxy}_{g}

\quad \Rightarrow \left( \forall s

\quad \bullet s \in_{g} g

\quad \Rightarrow \left( \forall f \bullet \text{Image}_{p_{f}} s \subseteq_{g} g \Rightarrow \text{Image}_{p_{f}} s \in_{g} g \right) \right)

\textbf{Pair}_{g_{\subseteq}}_{\text{thm}} \quad \vdash \forall x y \bullet x \in_{g} \text{Pair}_{g_{x}} y y \Rightarrow y \in_{g} \text{Pair}_{g_{x}} y y

\textbf{Pair}_{g_{\text{tc}}}_{\text{thm}} \quad \vdash \forall s t \bullet s \in_{g^{+}} \text{Pair}_{g_{s}} t t \Rightarrow t \in_{g^{+}} \text{Pair}_{g_{s}} t t

\textbf{Pair}_{g_{\text{eq}}}_{\text{thm}}
\[ \vdash \forall s \ t \ u \ v \]
- \[ \text{Pair}_g \ s \ t = \text{Pair}_g \ u \ v \]
  \[ \iff s = u \land t = v \lor s = v \land t = u \]

**GClose Pair**
- \( \vdash \forall \ g \)
  - \[ \text{galaxy} \ g \]
  - \( \Rightarrow (\forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow \text{Pair}_g s \ t \in_g g) \)

**Sing thm 2**
- \( \vdash \forall x \bullet x \in_g \text{Sing} \ x \)

**GClose Sing**
- \( \vdash \forall \ b \bullet \text{galaxy} \ g \Rightarrow (\forall s \bullet s \in_g g \Rightarrow \text{Sing}_g s \in_g g) \)

**\( \equiv_g \cup_g \text{thm} \)**
- \( \vdash \forall A \ B \ C \bullet A \equiv_g A \cup_g B \land B \equiv_g A \cup_g B \)

**\( \cup_g \equiv_g \text{thm} 2 \)**
- \( \vdash \forall A \ B \ C \ D \bullet A \equiv_g C \land B \equiv_g D \Rightarrow A \cup_g B \equiv_g C \cup_g D \)

**\( \cup_g \varnothing_g \text{ clauses} \)**
- \( \vdash \forall A \bullet A \cup_g \varnothing_g = A \land \varnothing_g \cup_g A = A \)

**GClose \( \cup_g \)**
- \( \vdash \forall g \)
  - \[ \text{galaxy} \ g \]
  - \( \Rightarrow (\forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow s \cup_g t \in_g g) \)

**\( \cap_g \text{ thm} \)**
- \( \vdash \forall x \ s \ e \)
  - \[ x \in_g s \Rightarrow (e \in_g \cap_g s \iff (\forall y \bullet y \in_g s \Rightarrow e \in_g y)) \]

**\( \cap_g \equiv_g \text{ thm} \)**
- \( \vdash \forall s \ t \bullet s \in_g t \Rightarrow \cap_g t \equiv_g s \)

**\( \equiv_g \cap_g \text{ thm} \)**
- \( \vdash \forall \ A \ B \)
  - \[ \bullet A \in_g B \]
  - \( \Rightarrow (\forall C \bullet (\forall D \bullet D \in_g B \Rightarrow C \equiv_g D) \Rightarrow C \equiv_g \cap_g B) \)

**GClose \( \cap_g \)**
- \( \vdash \forall g \bullet \text{galaxy} \ g \Rightarrow (\forall s \bullet s \in_g g \Rightarrow \cap_g s \in_g g) \)

**GClose \( \cap_g \)**
- \( \vdash \forall \ g \)
  - \[ \text{galaxy} \ g \]
  - \( \Rightarrow (\forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow s \cap_g t \in_g g) \)

**\( \equiv_g \cap_g \text{ thm} \)**
- \( \vdash \forall A \ B \ C \bullet A \equiv_g C \land B \equiv_g C \Rightarrow A \cap_g B \equiv_g C \cap_g D \)

**\( \cap_g \equiv_g \text{ thm} 3 \)**
- \( \vdash \forall A \ B \ C \ D \bullet A \equiv_g C \land B \equiv_g D \Rightarrow A \cap_g B \equiv_g C \cap_g D \)

**not_\_x_\_in_\_x_\_thm**
- \( \vdash \neg (\exists x \bullet x \in_g x) \)

**GClose_\_fc_\_clauses**
- \( \vdash \forall \ g \)
  - \[ \text{galaxy} \ g \]
  - \( \Rightarrow (\forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow \text{Pair}_g s \ t \in_g g) \)
  - \( \land (\forall s \bullet s \in_g g \Rightarrow \text{Sing}_g s \in_g g) \)
  - \( \land (\forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow s \cup_g t \in_g g) \)
  - \( \land (\forall s \bullet s \in_g g \Rightarrow \cap_g s \in_g g) \)
  - \( \land (\forall s \ t \bullet s \in_g g \land t \in_g g \Rightarrow s \cap_g t \in_g g) \)

**tce_\_clauses**
- \( \vdash \forall s \)
  - \[ s \in_g \+)
  - \[ \text{Sing}_g s \]
  - \[ \land s \in_g \+)
  - \[ \text{Pair}_g s \ t \land s \in_g \+)
  - \[ \text{Pair}_g s \ t \land s \in_g \+)

**gst_opext_\_clauses**
- \( \vdash \forall s \ t \ x \ f \ u \ v \ e \ p \)
  - \[ \neg s \in_g \varnothing_g \]
  - \[ \land (t \in_g \text{Pair}_g s \ t \land s \in_g \+)

53
\( \land (t \in_g \bigcup_g s \iff (\exists e \bullet t \in_g e \land e \in_g s)) \)
\( \land (x \in_g \text{Image} f s \iff (\exists e \bullet e \in_g s \land x = f e)) \)
\( \land (\text{Pair}_g s t = \text{Pair}_g u v \iff s = u \land t = v \lor s = v \land t = u) \)
\( \land (e \in_g \text{Pair}_g s t \iff e = s \lor e = t) \)
\( \land (\text{Sing} s = \text{Sing} t \iff s = t) \)
\( \land (e \in_g \text{Sing} s \iff e = s) \)
\( \land (\text{Pair}_g s t = \text{Sing} u \iff s = u \land t = u) \)
\( \land (\text{Sing} s = \text{Pair}_g t u \iff s = t \land s = u) \)
\( \land (e \in_g \text{Sep} s p \iff e \in_g s \land p e) \)
\( \land (e \in_g s \cup_g t \iff e \in_g s \lor e \in_g t) \)
\( \land (e \in_g s \cap_g t \iff e \in_g s \land e \in_g t) \)

**gst_relext_clauses**

\[ \vdash \forall s \ t \]
\[ \bullet (s = t \iff (\forall e \bullet e \in_g s \iff e \in_g t)) \]
\[ \land (s \subsetneq_g t \iff (\forall e \bullet e \in_g s \Rightarrow e \in_g t)) \]
9 The Theory gst-fun

9.1 Parents

\[ gst - ax \]

9.2 Children

\[ GS \]

9.3 Constants

\[
\begin{align*}
\langle \rangle_g & : GS \to GS \\
\text{snd} & : GS \to GS \\
\text{fst} & : GS \to GS \\
\text{MkPair}_g & : GS \times GS \to GS \\
\text{MkTriple}_g & : GS \times GS \times GS \to GS \\
\text{rel} & : GS \to BOOL \\
\text{dom} & : GS \to GS \\
\text{ran} & : GS \to GS \\
\text{field} & : GS \to GS \\
\triangleleft_g & : GS \to GS \to GS \\
\triangleright_g & : GS \to GS \to GS \\
\triangleleft_{g,closed} & : GS \to BOOL \\
\lambda_g & : (GS \to GS) \to GS \to GS \\
\triangleright_{g, closed} & : GS \to GS \to GS \\
\text{id} & : GS \to GS \\
\preceq_g & : GS \to GS \to GS \\
\succeq_{g} & : GS \to GS \to GS
\end{align*}
\]

9.4 Aliases

\[
\begin{align*}
\triangleleft & \quad \triangleright \\
\triangleleft_{g} & \quad \triangleright_{g} \\
\preceq_{g} & \quad \succeq_{g}
\end{align*}
\]
9.5 Fixity

Binder:
\[\Sigma_g \land_g\]

Right Infix 240:
\[\mapsto_g \rightarrow_g \times_g \mapsto_g \rightarrow_g\]

Right Infix 250:
\[\phi_g \gamma_g \bigoplus_g\]

Right Infix 300:
\[\triangleright_g \triangleright_g \triangleleft_g \triangleleft_g\]

9.6 Definitions

\[\mapsto_g\]
\[\vdash \forall \ s \ t \bullet \ s \mapsto_g t = \text{Pair}_g (\text{Sing} \ s) (\text{Pair}_g \ s \ t)\]

\[\text{fst}\]
\[\vdash \forall \ s \ t \bullet \ \text{fst} (s \mapsto_g t) = s \land \ \text{snd} (s \mapsto_g t) = t\]

\[\text{MkPair}_g\]
\[\vdash \exists \ \text{lr} \bullet \text{MkPair}_g \text{lr} = \text{Fst} \ \text{lr} \mapsto_g \text{Snd} \ \text{lr}\]

\[\text{MkTriple}_g\]
\[\vdash \exists \ x \bullet \text{MkTriple}_g \text{t} = \text{Fst} \ t \mapsto_g \text{MkPair}_g (\text{Snd} \ t)\]

\[\text{rel}\]
\[\vdash \forall \ x \bullet \text{rel} \ x \iff (\forall \ y \bullet \ y \in_g x \Rightarrow (\exists \ s \bullet y = s \mapsto_g t))\]

\[\text{dom}\]
\[\vdash \forall \ x \bullet \text{dom} \ x = \text{Sep} (Gx \ x) (\lambda \ w \bullet \exists \ v \bullet v \mapsto_g w \in_g x)\]

\[\text{ran}\]
\[\vdash \forall \ x \bullet \text{ran} \ x = \text{Sep} (Gx \ x) (\lambda \ w \bullet \exists \ v \bullet v \mapsto_g w \in_g x)\]

\[\text{field}\]
\[\vdash \forall \ s \bullet \ e \in_g \text{field} \ s \iff e \in_g \text{dom} \ s \lor e \in_g \text{ran} \ s\]

\[\triangleleft_g\]
\[\vdash \forall \ s \ r \bullet \ s < r = \text{Sep} \ r (\lambda \ p \bullet \text{fst} \ p \in_g s)\]

\[\triangleright_g\]
\[\vdash \forall \ s \ r \bullet \ r \triangleright s = \text{Sep} \ r (\lambda \ p \bullet \text{snd} \ p \in_g s)\]

\[\triangleleft_g\]
\[\vdash \forall \ s \ r \bullet \ s \triangleleft r = \text{Sep} \ r (\lambda \ p \bullet \text{fst} \ p \in_g s)\]

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

\[\rel2\text{DepType}_g\]
\[\vdash \forall \ r\]
\[\bullet \text{Rel2DepType}_g \ r = \text{Sep} (Gx \ r) (\lambda \ e\]
\[\bullet \exists \ i \ t\]
\[\bullet \ e = i \mapsto_g t \land i \in_g \text{dom} \ r \land (\forall \ j \bullet j \in_g t \iff i \mapsto_g j \in_g r)\]

\[\text{DepSum}_g\]
\[\vdash \forall \ t\]
\[\bullet \text{DepSum}_g \ t = \text{Sep} (Gx \ t) (\lambda \ e\]
\[\bullet \exists \ i \ t2 \ v\]
\[\bullet \ e = i \mapsto_g v \land v \in_g t2 \land i \mapsto_g t2 \in_g t)\]

\[\Sigma_g\]
\[\vdash \forall \ f \ s\]
\[\bullet \Sigma_g f \ s = \bigcup_g (\text{Imagep} (\lambda \ e \bullet \text{Imagep} (\lambda \ x \bullet e \mapsto_g x) (f \ e)) \ s)\]

\[\times_g\]
\[\vdash \forall \ s \ t\]
\[\bullet \ s \times_g t = \bigcup_g (\text{Imagep} (\lambda \ se \bullet \text{Imagep} (\lambda \ te \bullet se \mapsto_g te) \ t) \ s)\]

\[\leftrightarrow_g\]
\[\vdash \forall \ s \ t \bullet \ s \leftrightarrow_g t = \mathbb{P}_g (s \times_g t)\]
\[ o_g \]

\[ \vdash \forall f \ g \]

\[ f = \text{Imagep} \]

\[ (\lambda \ p \ast \ \text{fst} (\text{fst} \ p) \mapsto_g \ \text{snd} \ (\text{snd} \ p)) \]

\[ (\text{Sep}) \]

\[ (g \times_g f) \]

\[ (\lambda \ p \ast \ p \mapsto (q \mapsto_g r) \mapsto_g r \mapsto_g s)) \]

\[ \text{fun} \]

\[ \vdash \forall x \]

\[ \text{fun} \ x \]

\[ \Leftrightarrow \ \text{rel} \ x \]

\[ \wedge (\forall \ s \ t \ u \ s \mapsto_g u \in_g x \wedge s \mapsto_g t \in_g x \Rightarrow u = t) \]

\[ \mapsto_g \]

\[ \vdash \forall \ s \ t \ s \mapsto_g t = \text{Sep} (s \mapsto_g t) \]

\[ \rightarrow_g \]

\[ \vdash \forall \ s \ t \ s \mapsto_g t = \text{Sep} (s \mapsto_g t) (\lambda r \mapsto \text{dom} \ r = s) \]

\[ \rightarrow_g{-}\text{closed} \]

\[ \vdash \forall \ s \]

\[ \rightarrow_g{-}\text{closed} \ s \]

\[ \Leftrightarrow (\forall \ d \ c \cdot d \in_g s \wedge c \in_g s \Rightarrow d \mapsto_g c \in_g s) \]

\[ \lambda_g \]

\[ \vdash \forall f \ s \]

\[ \$\lambda_g \ f \ s \]

\[ = \text{Sep} (s \times_g \text{Imagep} \ f \ s) (\lambda \ p \ast \ \text{snd} \ p = f (\text{fst} \ p)) \]

\[ g \]

\[ \vdash \forall f \ x \]

\[ f \ g \ x \]

\[ = (\text{if} \ \exists \ y \ast x \mapsto_g y \in_g f \]

\[ \text{then} \ x \mapsto_g y \in_g f \]

\[ \text{else} \ f) \]

\[ \text{id} \]

\[ \vdash \forall \ s \ast \ \text{id} \ s = \text{Sep} (s \times_g s) (\lambda x \mapsto \text{fst} \ x = \text{snd} \ x) \]

\[ \oplus_g \]

\[ \vdash \forall \ s \ t \]

\[ s \oplus_g t \]

\[ = \text{Sep} \]

\[ (s \cup_g t) \]

\[ (\lambda x \]

\[ \ast \text{if} \ \text{fst} \ x \in_g \text{dom} \ t \]

\[ \text{then} \ x \in_g t \]

\[ \text{else} \ x \in_g s) \]

9.7 Theorems

\[ \mapsto_g{-}\text{eq.thm} \]

\[ \vdash \forall \ s \ t \ u \ast s \mapsto_g t = u \mapsto_g v \Leftrightarrow s = u \wedge t = v \]

\[ \text{Pair}_{g \in \mapsto_g{-}\text{thm}} \]

\[ \vdash \forall \ s \ t \ast \text{Pair}_g \ s \ t \in_g s \mapsto_g t \]

\[ \text{Pair}_{g \in \mapsto_g{-}\text{Gx} \ast \mapsto_g{-}\text{thm}} \]

\[ \vdash \forall \ s \ t \ast \text{Pair}_g \ s \ t \in_g \text{Gx} (s \mapsto_g t) \]

\[ \mapsto_g{-}\text{Gx} \ast \text{Pair}_{g \mapsto_g{-}\text{thm}} \]

\[ \vdash \forall \ s \ t \ast s \mapsto_g t \in_g \text{Gx} (\text{Pair}_g \ s \ t) \]

\[ \mapsto_g\partial_g{-}\text{thm} \]

\[ \vdash \forall \ x \ y \ast \not\mapsto_g y = \partial_g \]

\[ \not\partial_g \ast \mapsto_g{-}\text{thm} \]

\[ \vdash \forall \ x \ y \ast \not\partial_g = x \mapsto_g y \]

\[ \text{GClose}_g{-}\text{thm} \]

\[ \vdash \forall \ g \]

\[ \ast \text{galaxy} \ g \]

\[ \Rightarrow (\forall \ s \ t \ast s \in_g g \wedge t \in_g g \Rightarrow s \mapsto_g t \in_g g) \]

\[ \text{tc} \ast \mapsto_g{-}\text{left.thm} \]

57
\( t \subseteq \_ \rightarrow \text{right_thm} \)
\[
\vdash \forall \ s \ t \bullet s \in g^+ \ \Rightarrow \ t \in g \ t
\]
\( \_ \rightarrow \text{tc_thm} \)
\[
\vdash \forall \ x \ y \cdot \text{tc} \ \$ g \ x \ (x \rightarrow g \ y) \wedge \text{tc} \ \$ g \ y \ (x \rightarrow g \ y)
\]
\( \text{rel}_{\mathcal{O}_g}\text{-thm} \)
\[
\vdash \text{rel} \ \mathcal{O}_g
\]
\( \text{dom}_{\mathcal{O}_g}\text{-thm} \)
\[
\vdash \text{dom} \ \mathcal{O}_g = \mathcal{O}_g
\]
\( \text{dom_thm} \)
\[
\vdash \forall \ r \ y \cdot \ y \in g \ \text{dom} \ r \ \Leftrightarrow (\exists \ x \cdot x \rightarrow g \ y \in g \ r)
\]
\( \text{GClose}_{\mathcal{O}_g}\text{-thm} \)
\[
\vdash \forall \ g \cdot \text{galaxy} \ g \Rightarrow (\forall \ r \cdot r \in g \ g \Rightarrow \text{dom} \ r \in g \ g)
\]
\( \text{ran}_{\mathcal{O}_g}\text{-thm} \)
\[
\vdash \text{ran} \ \mathcal{O}_g = \mathcal{O}_g
\]
\( \text{ran_thm} \)
\[
\vdash \forall \ r \ y \cdot \ y \in g \ \text{ran} \ r \ \Leftrightarrow (\exists \ x \cdot x \rightarrow g \ y \in g \ r)
\]
\( \text{GClose}_{\text{ran}}\text{-thm} \)
\[
\vdash \forall \ g \cdot \text{galaxy} \ g \Rightarrow (\forall \ r \cdot r \in g \ g \Rightarrow \text{ran} \ r \in g \ g)
\]
\( \text{to}\text{-ran_thm} \)
\[
\vdash \forall \ x \ y \cdot x \in g^+ \ \Rightarrow \ y \in g^+ \ y
\]
\( \times_{\_g}\text{-spec} \)
\[
\vdash \forall \ s \ t \ e \ \bullet \ e \in g \ s \times g \ t
\]
\[
\Leftrightarrow (\exists \ l \ r \cdot l \in g \ s \wedge r \in g \ t \wedge e = l \rightarrow g \ r)
\]
\( f^{\rightarrow gs}\text{-thm} \)
\[
\vdash \forall \ s \ t \ p \cdot p \in g \ s \times g \ t \Rightarrow \text{fst} \ p \rightarrow g \ \text{snd} \ p = p
\]
\( \in^{\rightarrow gs}\text{-thm} \)
\[
\vdash \forall \ p \ s \ t \ \bullet \ p \in g \ s \times g \ t \Rightarrow \text{fst} \ p \in g \ s \wedge \text{snd} \ p \in g \ t
\]
\( \rightarrow g^{\rightarrow gs}\times g\text{-thm} \)
\[
\vdash \forall \ l \ r \ s \ t \cdot l \rightarrow g \ r \in g \ s \times g \ t \Leftrightarrow l \in g \ s \wedge r \in g \ t
\]
\( \mathcal{O}_g^{\rightarrow gs}\text{-thm} \)
\[
\vdash \forall \ s \ t \ \bullet \ \mathcal{O}_g \in g \ s \times g \ t
\]
\( f^{\rightarrow gs}\text{-thm1} \)
\[
\vdash \forall \ p \ r \ s \ t \ \bullet \ p \in g \ r \wedge r \in g \ s \rightarrow g \ t \Rightarrow \text{fst} \ p \rightarrow g \ \text{snd} \ p = p
\]
\( \in^{\rightarrow gs}\text{-thm} \)
\[
\vdash \forall \ p \ r \ s \ t \ \bullet \ p \in g \ r \wedge r \in g \ s \rightarrow g \ t \Rightarrow \text{fst} \ p \in g \ s \wedge \text{snd} \ p \in g \ t
\]
\( o_g\text{-thm} \)
\[
\vdash \forall \ f \ g \ x
\]
\[
\bullet \ x \in g \ f \ o_g \ g
\]
\[
\Leftrightarrow (\exists \ q \ r \ s
\]
\[
\bullet \ q \rightarrow g \ r \in g \ g \wedge r \rightarrow g \ s \in g \ f \wedge x = q \rightarrow g \ s)
\]
\( o_g\text{-thm2} \)
\[
\vdash \forall \ f \ g \ x \ y
\]
\[
\bullet \ x \rightarrow g \ y \in g \ f \ o_g \ g
\]
\[
\Leftrightarrow (\exists \ z \cdot x \rightarrow g \ z \in g \ g \wedge z \rightarrow g \ y \in g \ f)
\]
\( o_g\text{-rel_thm} \)
\[
\vdash \forall \ r \ s \cdot \text{rel} \ r \wedge \text{rel} \ s \Rightarrow \text{rel} \ (r \ o_g \ s)
\]
\( o_g\text{-associative_thm} \)
\[
\vdash \forall \ f \ g \ h \cdot (f \ o_g \ g) \ o_g \ h = f \ o_g \ g \ o_g \ h
\]
\( \text{rel}_{\text{sub}}\text{-cp_thm} \)
\[
\vdash \forall \ x \cdot \text{rel} \ x \Leftrightarrow (\exists \ s \ t \cdot x \in g \ s \times g \ t)
\]
\( \text{fun}_{\mathcal{O}_g}\text{-thm} \)
\[
\vdash \text{fun} \ \mathcal{O}_g
\]
\( o_g\text{-fun_thm} \)
\[
\vdash \forall \ f \ g \cdot \text{fun} \ f \wedge \text{fun} \ g \Rightarrow \text{fun} \ (f \ o_g \ g)
\]
\( \text{ran}_{o_g}\text{-thm} \)
\[
\vdash \forall \ f \ g \cdot \text{ran} \ (f \ o_g \ g) \subseteq g \ \text{ran} \ f
\]
\( \text{dom}_{o_g}\text{-thm} \)
\[
\vdash \forall \ f \ g \cdot \text{dom} \ (f \ o_g \ g) \subseteq g \ \text{dom} \ g
\]
\( \text{dom}_{o_g}\text{-thm2} \)
\[
\vdash \forall \ f \ g \cdot \text{ran} \ g \subseteq g \ \text{dom} \ f \Rightarrow \text{dom} \ (f \ o_g \ g) = \text{dom} \ g
\]
\( \mathcal{O}_g^{\rightarrow g}\text{-thm} \)
\[
\vdash \forall \ s \ t \ \bullet \ \mathcal{O}_g \in g \ s \rightarrow g \ t
\]
\( \exists \mapsto_g \text{thm} \quad \vdash \forall s \cdot \exists f \cdot f \in_g s \mapsto_g t \)

\( \emptyset \in_g \emptyset \mapsto_g \text{thm} \quad \vdash \forall s \cdot \emptyset \in_g \emptyset \mapsto_g t \)

\( \exists \mapsto_g \text{thm} \quad \vdash \forall s \cdot (\exists x \in_g t) \Rightarrow (\exists f \cdot f \in_g s \mapsto_g t) \)

**app.thm1**  
\( \vdash \forall f \cdot x \in_g f \Rightarrow x \mapsto_g f \)

**app.thm2**  
\( \vdash \forall f \cdot f \mapsto_g f \)

**app.thm3**  
\( \vdash \forall f \cdot \text{fun } f \cdot x \in_g f \Rightarrow x \mapsto_g f \)

\( o_{g \mapsto_g} \text{thm} \quad \vdash \forall f \cdot x \mapsto_g f \cdot x \mapsto_g f \)

\( g \in_g \text{thm} \quad \vdash \forall f \cdot s \cdot t \cdot u \cdot f \in_g s \mapsto_g t \wedge u \in_g f \Rightarrow f \in_g u \in_g t \)

\( g \in_g \text{thm} \quad \vdash \forall f \cdot s \cdot t \cdot u \cdot f \in_g s \mapsto_g t \wedge u \in_g f \Rightarrow f \in_g u \in_g t \)

\( g \mapsto_g \mapsto_g \text{thm} \quad \vdash \forall f \cdot s \cdot t \cdot u \cdot f \in_g s \mapsto_g t \Rightarrow f \in_g dom f \mapsto_g t \)

\( \text{id.thm1} \quad \vdash \forall x \cdot x \in_g id s \Leftrightarrow (\exists y \cdot y \in_g x \wedge x = y \mapsto_g y) \)

\( \text{id.ap.thm} \quad \vdash \forall s \cdot x \cdot x \in_g s \Rightarrow id s \mapsto_g x \)

\( \text{id.g \mapsto_g \text{thm}} \quad \vdash \forall s \cdot t \cdot u \cdot s \subseteq_g t \mapsto_g u \Rightarrow id s \in_g t \mapsto_g u \)

\( \text{id \mapsto_g \text{thm}} \quad \vdash \forall s \cdot t \cdot u \cdot s \subseteq_g t \Rightarrow id s \in_g t \mapsto_g t \)

\( \text{id \mapsto \text{thm}} \quad \vdash \forall s \cdot t \cdot x \cdot \text{rel } (id s) \wedge \text{dom } (id s) = s \wedge \text{ran } (id s) = s \)

\( g \in_g \text{thm} \quad \vdash \forall s \cdot t \cdot x \cdot x \in_g s \mapsto_g t \wedge x \in_g t \mapsto_g x \in_g s \)

\( g \mapsto_g \text{thm} \quad \vdash \forall s \cdot t \cdot x \cdot y \cdot x \mapsto_g y \in_g s \mapsto_g t \wedge x \in_g y \in_g s \)

\( (\mapsto_g \text{rel.thm}} \quad \vdash \forall s \cdot \text{rel } s \wedge \text{rel } t \Rightarrow \text{rel } (s \mapsto_g t) \)

\( (\mapsto_g \text{fun.thm}} \quad \vdash \forall s \cdot \text{fun } s \wedge \text{fun } t \Rightarrow \text{fun } (s \mapsto_g t) \)
10 The Theory gst-ord

10.1 Parents

\[ \text{gst}_\text{ax} \]

10.2 Children

\[ \text{GST} \rightarrow \text{nat} \]

10.3 Constants

- \textit{connected} \quad \text{GST} \rightarrow \text{BOOL}
- \textit{ordinal} \quad \text{GST} \rightarrow \text{BOOL}
- \textit{\$<}_{\text{o}} \quad \text{GST} \rightarrow \text{GST} \rightarrow \text{BOOL}
- \textit{\$\leq}_{\text{o}} \quad \text{GST} \rightarrow \text{GST} \rightarrow \text{BOOL}
- \textit{suc}_{\text{o}} \quad \text{GST} \rightarrow \text{GST}
- \textit{successor} \quad \text{GST} \rightarrow \text{BOOL}
- \textit{limit}_{\text{ordinal}} \quad \text{GST} \rightarrow \text{BOOL}
- \textit{\$sub} \quad \text{GST} \rightarrow \text{GST} \rightarrow \text{BOOL}
- \textit{sup} \quad \text{GST} \rightarrow \text{GST}
- \textit{\$sub} \quad \text{GST} \rightarrow \text{GST} \rightarrow \text{BOOL}
- \textit{ssup} \quad \text{GST} \rightarrow \text{GST}
- \textit{rank} \quad \text{GST} \rightarrow \text{GST}
- \textit{\$+}_{\text{o}} \quad \text{GST} \rightarrow \text{GST} \rightarrow \text{GST}
- \textit{\$--}_{\text{o}} \quad \text{GST} \rightarrow \text{GST} \rightarrow \text{GST}

10.4 Fixity

\textit{Right Infix 200}: \quad \textit{sub} \quad \textit{ub}

\textit{Right Infix 240}: \quad \textit{<}_{\text{o}} \quad \textit{\leq}_{\text{o}}

\textit{Right Infix 300}: \quad \textit{+}_{\text{o}} \quad \textit{--}_{\text{o}}

10.5 Definitions

**connected** \quad \vdash \forall s

\quad \bullet \textit{connected} s

\quad \Leftrightarrow (\forall t \ u

\quad \bullet \ t \in_{\text{g}} s \land u \in_{\text{g}} s \Rightarrow t \in_{\text{g}} u \lor t = u \lor u \in_{\text{g}} t)

**ordinal** \quad \vdash \forall s \bullet \textit{ordinal} s \Leftrightarrow \text{transitive} \ s \land \text{connected} \ s

\textit{\$<}_{\text{o}} \quad \vdash \forall x \ y \bullet x \leq_{\text{o}} y \Leftrightarrow \text{ordinal} \ x \land \text{ordinal} \ y \land (x \in_{\text{g}} y \lor x = y)

\textit{\$\leq}_{\text{o}} \quad \vdash \forall x \ y

\quad \bullet x \leq_{\text{o}} y \Leftrightarrow \text{ordinal} \ x \land \text{ordinal} \ y \land (x \in_{\text{g}} y \lor x = y)

\textit{suc}_{\text{o}} \quad \vdash \forall x \bullet \textit{suc}_{\text{o}} \ x = x \cup_{\text{g}} \text{Sing} \ x

\textit{successor} \quad \vdash \forall s \bullet \textit{successor} \ s \Leftrightarrow (\exists t \bullet \text{ordinal} \ t \land s = \textit{suc}_{\text{o}} \ t)

\textit{limit}_{\text{ordinal}}
\[ \vdash \forall s \\
\bullet \text{limit}_\text{ordinal} \ s \\
\quad \Leftrightarrow \text{ordinal} \ s \land \neg \text{successor} \ s \land \neg s = \emptyset_g \]

\[ \text{ub} \vdash \forall \alpha \beta \bullet \alpha \text{ub} \beta \Leftrightarrow (\forall \gamma \bullet \gamma \in_g \alpha \Rightarrow \gamma <_o \beta) \]

\[ \text{sup} \vdash \forall \alpha \bullet \sup \alpha = \bigcup_g \alpha \]

\[ \text{sub} \vdash \forall \alpha \beta \bullet \alpha \text{sub} \beta \Leftrightarrow (\forall \gamma \bullet \gamma \in_g \alpha \Rightarrow \gamma <_o \beta) \]

\[ \text{ssup} \vdash \forall \alpha \bullet \ssup \alpha = \bigcup_g (\text{Imagep suc}_o \alpha) \]

\[ \text{rank} \vdash \forall \alpha \bullet \text{rank} \ x = \bigcup_g (\text{Imagep (suc}_o \alpha \text{rank}) \ x) \]

\[ \vdash \forall \alpha \beta \\
\bullet \alpha +_o \beta \\
\quad = (\text{if } \beta = \emptyset_g \text{ then } \alpha \text{ else } \text{ssup (Imagep ($+_o \alpha$) } \beta)) \]

\[ \vdash T \]

### 10.6 Theorems

\[ \text{mem_less_thm} \vdash \forall \alpha \beta \bullet \text{ordinal} \alpha \land \text{ordinal} \beta \land \alpha \in_g \beta \Rightarrow \alpha <_o \beta \]

\[ \text{less_mem_thm} \vdash \forall \alpha \beta \bullet \alpha <_o \beta \Rightarrow \text{ordinal} \alpha \land \text{ordinal} \beta \land \alpha \in_g \beta \]

\[ \text{ord_mem_psub_thm} \]

\[ \vdash \forall \alpha \bullet \text{ordinal} \alpha \Rightarrow (\forall \beta \bullet \beta \in_g \alpha \Rightarrow \beta \in_g \alpha) \]

\[ \text{lto_psub_thm} \]

\[ \vdash \forall \alpha \beta \bullet \alpha <_o \beta \Rightarrow \alpha \in_g \beta \]

\[ \text{lo_trans_thm} \]

\[ \vdash \forall \alpha \beta \gamma \bullet \alpha <_o \beta \land \beta <_o \gamma \Rightarrow \alpha <_o \gamma \]

\[ \text{leo_lo_thm} \]

\[ \vdash \forall x y \\
\bullet x <_o y \Leftrightarrow \text{ordinal} x \land \text{ordinal} y \land (x <_o y \lor x = y) \]

\[ \text{leo_sub_thm} \]

\[ \vdash \forall \alpha \beta \bullet \alpha \leq_o \beta \Rightarrow \alpha \leq_g \beta \]

\[ \text{leo_trans_thm} \]

\[ \vdash \forall \alpha \beta \gamma \bullet \alpha \leq_o \beta \land \beta \leq_o \gamma \Rightarrow \alpha \leq_o \gamma \]

\[ \text{lo_leo_trans_thm} \]

\[ \vdash \forall \alpha \beta \gamma \bullet \alpha \leq_o \beta \land \beta <_o \gamma \Rightarrow \alpha <_o \gamma \]

\[ \text{ordinal}_\emptyset_g \]

\[ \vdash \text{ordinal} \emptyset_g \]

\[ \text{trans_suc_trans} \]

\[ \vdash \forall x \bullet \text{transitive} \ x \Rightarrow \text{transitive} \ (\text{suc}_o \ x) \]

\[ \text{ord_suc_conn} \]

\[ \vdash \forall x \bullet \text{connected} \ x \Rightarrow \text{connected} \ (\text{suc}_o \ x) \]

\[ \text{ord_suc Ord_thm} \]

\[ \vdash \forall x \bullet \text{ordinal} \ x \Rightarrow \text{ordinal} \ (\text{suc}_o \ x) \]

\[ \emptyset_g \text{-not suc}_o \text{thm} \]

\[ \vdash \neg (\exists \alpha \bullet \text{suc}_o \alpha = \emptyset_g) \]

\[ \text{not in suc}_o \text{thm} \]

\[ \vdash \forall \alpha \bullet \neg \alpha = \text{suc}_o \alpha \]

\[ \text{leo suc_thm} \]

\[ \vdash \forall \alpha \bullet \text{ordinal} \alpha \Rightarrow \alpha \leq_o \text{suc}_o \alpha \]

\[ \text{lo suc_thm} \]

\[ \vdash \forall \alpha \bullet \text{ordinal} \alpha \Rightarrow \alpha <_o \text{suc}_o \alpha \]

\[ \text{conn sub thm} \]

\[ \vdash \forall x \bullet \text{connected} \ x \Rightarrow (\forall y \bullet y \leq_g x \Rightarrow \text{connected} \ y) \]

\[ \text{conn mem ord} \]

\[ \vdash \forall x \bullet \text{ordinal} \ x \Rightarrow (\forall y \bullet y \in_g x \Rightarrow \text{connected} \ y) \]

\[ \text{trans mem ord} \]

\[ \vdash \forall x \bullet \text{ordinal} \ x \Rightarrow (\forall y \bullet y \in_g x \Rightarrow \text{transitive} \ y) \]

\[ \text{ord mem ord} \]

\[ \vdash \forall x \bullet \text{ordinal} \ x \Rightarrow (\forall y \bullet y \in_g x \Rightarrow \text{ordinal} \ y) \]

\[ \text{GCloseSuc} \]

\[ \vdash \forall g \bullet \text{galaxy} \ g \Rightarrow (\forall x \bullet x \in_g g \Rightarrow \text{suc}_o \ x \in_g g) \]
\(\text{tran}_\cap \text{thm}\) \(\vdash \forall x y\)
\(\rightarrow\) transitive \(x \wedge\) transitive \(y \rightarrow\) transitive \((x \cap y)\)

\(\text{tran}_\cup \text{thm}\) \(\vdash \forall x y\)
\(\rightarrow\) transitive \(x \wedge\) transitive \(y \rightarrow\) transitive \((x \cup y)\)

\(\text{conn}_\cap \text{thm}\) \(\vdash \forall x y\)
\(\rightarrow\) connected \(x \wedge\) connected \(y \rightarrow\) connected \((x \cap y)\)

\(\text{ord}_\cap \text{thm}\) \(\vdash \forall x y\)
\(\rightarrow\) ordinal \(x \wedge\) ordinal \(y \rightarrow\) ordinal \((x \cap y)\)

\(\text{trich}_\text{for}_\text{ords}_\text{thm}\)
\(\vdash \forall x y\)
\(\rightarrow\) ordinal \(x \wedge\) ordinal \(y \rightarrow\) ordinal \(x \leq y \vee y \leq x\)

\(\text{sub}_\text{leo}_\text{thm}\)
\(\vdash \forall x y\)
\(\rightarrow\) ordinal \(x \wedge\) ordinal \(y \rightarrow (x \leq y \equiv x \leq y)\)

\(\text{sub}_\text{leo}_\text{thm1}\)
\(\vdash \forall x y\)
\(\rightarrow\) ordinal \(x \wedge\) ordinal \(y \wedge x \leq y \Rightarrow x \leq y\)

\(\text{successor}_\text{ord}_\text{thm}\)
\(\vdash \forall x\)
\(\rightarrow\) successor \(x \Rightarrow\) ordinal \(x\)

\(\text{wf}_\text{ordinals}_\text{thm}\)
\(\vdash\) well-founded \(\lhd\)

\(\text{ordinal}_\text{kind}_\text{thm}\)
\(\vdash \forall n\)
\(\rightarrow\) ordinal \(n \Rightarrow n = \emptyset \vee\) successor \(n \vee\) limit ordinal \(n\)

\(\text{ordinal}_\text{limit}_\text{thm}\)
\(\vdash \forall n\)
\(\rightarrow\) ordinal \(n \Rightarrow n = \emptyset \vee\) successor \(n \vee\) limit ordinal \(n\)

\(\text{sup}_\text{lub}_\text{thm}\)
\(\vdash \forall (\forall \beta \rightarrow\) ordinal \(\beta \Rightarrow\) ordinal \((\cup_g \alpha)\)

\(\text{sup}_\text{lub}_\text{thm}\)
\(\vdash \forall (\forall \beta \rightarrow\) ordinal \(\beta \Rightarrow\) ordinal \((\cup_g \alpha)\)

\(\text{sup}_\text{lub}_\text{thm}\)
\(\vdash \forall (\forall \beta \rightarrow\) ordinal \(\beta \Rightarrow\) ordinal \((\cup_g \alpha)\)

\(\text{ord}_\text{ssup}_\text{thm}\)
\(\vdash \forall (\forall \beta \rightarrow\) ordinal \(\beta \Rightarrow\) ordinal \((\cup_g \alpha)\)

\(\text{ord}_\text{plus}_\text{thm}\)
\(\vdash \forall (\forall \beta \rightarrow\) ordinal \(\alpha \wedge\) ordinal \(\beta \Rightarrow\) ordinal \((\alpha + \beta)\)
11 The Theory gst-nat

11.1 Parents

\(gst\)-\(ord\)

11.2 Children

\(GS\)

11.3 Constants

**natural\_number**

\(GS \rightarrow BOOL\)

\(\$<_{\text{gn}}\) \(GS \rightarrow GS \rightarrow BOOL\)

\(Nat_{g}\) \(\mathbb{N} \rightarrow GS\)

11.4 Fixity

*Right Infix 240:*

\(<_{\text{gn}}\)

11.5 Definitions

**natural\_number**

\(\vdash \forall s\)

- \(natural\_number\ s\)
  \(\iff s = \emptyset_{g}\)
  \(\lor\ successor\ s\)
  \(\land (\forall t \cdot t \in_{g} s \Rightarrow t = \emptyset_{g} \lor successor\ t)\)

\(<_{\text{gn}}\)

\(\vdash \forall x\ y\)

- \(x <_{\text{gn}} y\)
  \(\iff natural\_number\ x \land natural\_number\ y \land x \in_{g} y\)

\(Nat_{g}\)

\(\vdash Nat_{g}\ 0 = \emptyset_{g}\)
\(\land (\forall n \cdot Nat_{g}\ (n + 1) = suc_{o}\ (Nat_{g}\ n))\)

11.6 Theorems

**wf\_nat\_thm** \(\vdash well\_founded\ \$<_{\text{gn}}\)

**nat\_induct\_thm**

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

**nat\_induct\_thm2**

\(\vdash \forall p\)

- \(\forall x\)
  \(\bullet natural\_number\ x \land (\forall y\ y <_{\text{gn}} x \Rightarrow p\ y)\)
  \(\Rightarrow p\ x)\)
  \(\Rightarrow (\forall x\ natural\_number\ x \Rightarrow p\ x)\)

**ord\_nat\_thm** \(\vdash \forall n\bullet natural\_number\ n \Rightarrow ordinal\ n\)

**mem\_nat\_ord\_thm**

\(\vdash \forall n\bullet natural\_number\ n \Rightarrow (\forall m\bullet m \in_{g} n \Rightarrow ordinal\ m)\)
\text{nat_not_lim_thm} \\
\quad \vdash \forall n \bullet \text{natural} \_\text{number} \ n \Rightarrow - \text{limit} \_\text{ordinal} \ n \\
\text{nat_zero_or_suc_thm} \\
\quad \vdash \forall n \bullet \text{natural} \_\text{number} \ n \Rightarrow \text{successor} \ n \lor n = \emptyset \\
\text{mem_nat_not_lim_thm} \\
\quad \vdash \forall m \ n \bullet \text{natural} \_\text{number} \ n \land m \in_g n \Rightarrow - \text{limit} \_\text{ordinal} \ m \\
\text{mem_nat_nat_thm} \\
\quad \vdash \forall n \\
\quad \quad \bullet \text{natural} \_\text{number} \ n \\
\quad \quad \quad \Rightarrow (\forall m \bullet m \in_g n \Rightarrow \text{natural} \_\text{number} \ m) \\
\text{nat_in_G}\emptyset_g \text{thm} \\
\quad \vdash \forall n \bullet \text{natural} \_\text{number} \ n \Rightarrow n \in_g G \times \emptyset_g \\
\text{w_exists_thm} \\
\quad \exists w \bullet \forall z \bullet z \in_g w \Leftrightarrow \text{natural} \_\text{number} z \\
\text{ord_nat_thm2} \\
\quad \vdash \forall n \bullet \text{ordinal} \ (\text{Nat}_g n) \\
\text{not_suc_nat_zero_thm} \\
\quad \vdash \forall n \bullet \neg \text{suc}_o (\text{Nat}_g n) = \emptyset_g \\
\text{less_sum_thm} \\
\quad \vdash \forall x \ y \bullet x \leq y \Rightarrow (\exists z \bullet x + z = y) \\
\text{natg_mono_thm} \\
\quad \vdash \forall x \ y \bullet \text{Nat}_g x \leq_o \text{Nat}_g (x + y) \\
\text{natg_one_one_thm} \\
\quad \vdash \forall x \ y \bullet \text{Nat}_g x = \text{Nat}_g y \Rightarrow x = y \\
\text{natg_one_one_thm2} \\
\quad \vdash \forall x \ y \bullet \text{Nat}_g x = \text{Nat}_g y \Leftrightarrow x = y
12 The Theory GS

12.1 Parents

\[
gst\text{-nat} \quad gst\text{-ord} \quad gst\text{-fun}
\]

12.2 Children

\[
misc2 \quad ICsyn
\]
<table>
<thead>
<tr>
<th>Theorem</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>galaxies_3_thm</td>
<td>13, 52</td>
</tr>
<tr>
<td>GalaxiesTransitive_thm</td>
<td>14, 52</td>
</tr>
<tr>
<td>galaxy</td>
<td>13, 49, 50</td>
</tr>
<tr>
<td>galaxy_Gx</td>
<td>13, 52</td>
</tr>
<tr>
<td>GS</td>
<td>18, 53</td>
</tr>
<tr>
<td>GClose ∩_thm</td>
<td>14</td>
</tr>
<tr>
<td>GClose ∪_thm</td>
<td>18, 53</td>
</tr>
<tr>
<td>GClose ∈_thm</td>
<td>17, 53</td>
</tr>
<tr>
<td>GClose_⊆_thm</td>
<td>14</td>
</tr>
<tr>
<td>GClose_dom_thm</td>
<td>23, 58</td>
</tr>
<tr>
<td>GClose_fc_clauses</td>
<td>14, 52</td>
</tr>
<tr>
<td>GClose_fc_clauses2</td>
<td>18, 53</td>
</tr>
<tr>
<td>GClose_ran_thm</td>
<td>23, 58</td>
</tr>
<tr>
<td>GClose_tc ∈_thm</td>
<td>14, 52</td>
</tr>
<tr>
<td>GClose ⇒ g_thm</td>
<td>22, 57</td>
</tr>
<tr>
<td>GClosePair_g</td>
<td>16, 53</td>
</tr>
<tr>
<td>GCloseSep_thm</td>
<td>14</td>
</tr>
<tr>
<td>GCloseSing</td>
<td>17, 53</td>
</tr>
<tr>
<td>GImagePC</td>
<td>18, 52</td>
</tr>
<tr>
<td>GS</td>
<td>8, 49</td>
</tr>
<tr>
<td>gs_eq_ind_thm</td>
<td>10, 51</td>
</tr>
<tr>
<td>gs_eq_ind_thm2</td>
<td>10, 51</td>
</tr>
<tr>
<td>gs_ext_axiom</td>
<td>9, 50</td>
</tr>
<tr>
<td>GSINDUCTION_T</td>
<td>10</td>
</tr>
<tr>
<td>GSINDUCTION_T2</td>
<td>10</td>
</tr>
<tr>
<td>gs_induction_tac</td>
<td>10</td>
</tr>
<tr>
<td>gs_induction_tac2</td>
<td>10</td>
</tr>
<tr>
<td>gs_wf_axiom</td>
<td>9, 50</td>
</tr>
<tr>
<td>gs_wf_ind_thm</td>
<td>10, 51</td>
</tr>
<tr>
<td>gs_wf_min_thm</td>
<td>9, 51</td>
</tr>
<tr>
<td>gs_wf_thm1</td>
<td>9, 51</td>
</tr>
<tr>
<td>gs_wf_tc_thm</td>
<td>9, 51</td>
</tr>
<tr>
<td>gs_wf_tc_thm2</td>
<td>10, 51</td>
</tr>
<tr>
<td>gset</td>
<td>8</td>
</tr>
<tr>
<td>gset_opext_clauses</td>
<td>53</td>
</tr>
<tr>
<td>gset_reext_clauses</td>
<td>54</td>
</tr>
<tr>
<td>Gx</td>
<td>13, 49, 51</td>
</tr>
<tr>
<td>Gx ⊆ g-galaxy</td>
<td>13, 52</td>
</tr>
<tr>
<td>Gx_mono_thm</td>
<td>14, 52</td>
</tr>
<tr>
<td>Gx_mono_thm2</td>
<td>14, 52</td>
</tr>
<tr>
<td>Gx_mono_thm3</td>
<td>14, 52</td>
</tr>
<tr>
<td>Gx_mono_thm4</td>
<td>14, 52</td>
</tr>
<tr>
<td>Gx_trans_thm</td>
<td>14, 52</td>
</tr>
<tr>
<td>Gx_trans_thm2</td>
<td>14, 52</td>
</tr>
<tr>
<td>Gx_trans_thm3</td>
<td>14, 52</td>
</tr>
<tr>
<td>id</td>
<td>55, 57</td>
</tr>
<tr>
<td>id ∈ g-g_thm1</td>
<td>59</td>
</tr>
<tr>
<td>id ∈ g-g_thm2</td>
<td>59</td>
</tr>
<tr>
<td>id_ap_thm</td>
<td>59</td>
</tr>
<tr>
<td>id_clauses</td>
<td>59</td>
</tr>
<tr>
<td>id_thm1</td>
<td>59</td>
</tr>
<tr>
<td>ImageP</td>
<td>15, 49, 51</td>
</tr>
<tr>
<td>leo_lo_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>leo_lo_trans_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>leo_sub_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>less_mem_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>less_sum_thm</td>
<td>47, 64</td>
</tr>
<tr>
<td>limit_ordinal</td>
<td>40, 60</td>
</tr>
<tr>
<td>lo_leo_trans_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>lo_trans_thm</td>
<td>37, 61</td>
</tr>
<tr>
<td>lto_psub_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>mem_less_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>mem_nat_thm</td>
<td>64</td>
</tr>
<tr>
<td>mem_nat_not_thm</td>
<td>64</td>
</tr>
<tr>
<td>mem_nat_not_lim_thm</td>
<td>63</td>
</tr>
<tr>
<td>mem_nat_or_thm</td>
<td>63</td>
</tr>
<tr>
<td>mem_not_empty_thm</td>
<td>15, 52</td>
</tr>
<tr>
<td>MkPair_g</td>
<td>22, 55, 56</td>
</tr>
<tr>
<td>MkTriple_g</td>
<td>23, 55, 56</td>
</tr>
<tr>
<td>nat_eq_G∅_g_thm</td>
<td>64</td>
</tr>
<tr>
<td>nat_in_G∅_g_thm</td>
<td>63</td>
</tr>
<tr>
<td>nat_induct_thm</td>
<td>63</td>
</tr>
<tr>
<td>nat_induct_thm2</td>
<td>63</td>
</tr>
<tr>
<td>nat_not_lim_thm</td>
<td>64</td>
</tr>
<tr>
<td>nat_zero_or_suc_thm</td>
<td>64</td>
</tr>
<tr>
<td>natg_monoid_thm</td>
<td>47, 64</td>
</tr>
<tr>
<td>natg_one_thm</td>
<td>47, 64</td>
</tr>
<tr>
<td>natg_one_thm2</td>
<td>47, 64</td>
</tr>
<tr>
<td>natg_one_thm3</td>
<td>47, 64</td>
</tr>
<tr>
<td>natural_number</td>
<td>43, 63</td>
</tr>
<tr>
<td>Nat_g</td>
<td>47, 63</td>
</tr>
<tr>
<td>not_in_G∅_g_thm</td>
<td>37, 61</td>
</tr>
<tr>
<td>not_in_suc_thm</td>
<td>11, 51</td>
</tr>
<tr>
<td>not_suc_natural_thm</td>
<td>47, 64</td>
</tr>
<tr>
<td>not_x_in_x_thm</td>
<td>18, 53</td>
</tr>
<tr>
<td>Ontology_axiom</td>
<td>11, 50</td>
</tr>
<tr>
<td>ord_∩_thm</td>
<td>62</td>
</tr>
<tr>
<td>ord_mem_∩_thm</td>
<td>61</td>
</tr>
<tr>
<td>ord_mem_∩_thm</td>
<td>36, 61</td>
</tr>
<tr>
<td>ord_mem_psub_thm</td>
<td>63</td>
</tr>
<tr>
<td>ord_mem_psub_thm</td>
<td>63</td>
</tr>
<tr>
<td>ord_nat_thm</td>
<td>47, 64</td>
</tr>
<tr>
<td>ord_plus_thm</td>
<td>62</td>
</tr>
<tr>
<td>ordplus_thm</td>
<td>61</td>
</tr>
<tr>
<td>ord_suc_ordinal_thm</td>
<td>35, 60</td>
</tr>
<tr>
<td>ordinal</td>
<td>40, 62</td>
</tr>
<tr>
<td>ordinal_G∅_g_thm</td>
<td>41, 62</td>
</tr>
<tr>
<td>ordinal_kind_thm</td>
<td>41, 62</td>
</tr>
<tr>
<td>ordinal_limit_thm</td>
<td>41, 62</td>
</tr>
<tr>
<td>ordinal התו_ thm</td>
<td>41, 62</td>
</tr>
<tr>
<td>o_g</td>
<td>55, 57</td>
</tr>
<tr>
<td>og</td>
<td>58</td>
</tr>
<tr>
<td>og_ensemble_thm</td>
<td>58</td>
</tr>
<tr>
<td>o_g_fun_thm</td>
<td>58</td>
</tr>
<tr>
<td>o_g_rel_thm</td>
<td>58</td>
</tr>
<tr>
<td>o_g_thm</td>
<td>58</td>
</tr>
<tr>
<td>o_g_thm2</td>
<td>58</td>
</tr>
<tr>
<td>o_g_thm3</td>
<td>59</td>
</tr>
<tr>
<td>o_g_thm4</td>
<td>59</td>
</tr>
<tr>
<td>Pair_g</td>
<td>16, 49, 51</td>
</tr>
<tr>
<td>Pair_g ∈ _→ _g_thm</td>
<td>22, 57</td>
</tr>
<tr>
<td>Pair_g ∈ _→ _thm</td>
<td>16, 52</td>
</tr>
<tr>
<td>Pair_g ∈ _→ _thm</td>
<td>22, 57</td>
</tr>
<tr>
<td>Pair_g ∈ Gx → _g_thm</td>
<td>22, 57</td>
</tr>
<tr>
<td>Pair_g_eq_thm</td>
<td>16, 52</td>
</tr>
</tbody>
</table>