Mathematical Case Studies:

—

Some Group Theory*

R.D. Arthan
Lemma 1 Ltd.
rda@lemma-one.com

27 September 2017

Abstract

This ProofPower-HOL script contains definitions and proofs concerning the elements of group theory. What is currently covered is what is covered in the first chapter of any good text on the subject: in preparation for the introduction of quotient groups, we begin with a purely set-theoretical study of equivalence relations and the quotient of a set with respect to an equivalence relation. This is followed by the definitions of the concepts of group, homomorphism between groups, subgroup, normal subgroup, kernel of a homomorphism, congruence modulo a subgroup, coset of a subgroup, and quotient group. Theorems proved included the three isomorphism theorems, the Cayley representation theorem and Lagrange's theorem. Several examples of groups are exhibited and used to show that the various abstract notions lead to the expected theorems: e.g., that the exponential function is an isomorphism between the additive group of real numbers and the multiplicative group of positive real numbers.

Reference: LEMMA1/HOL/WRK068; Current git revision: 5991a98

*First posted 2 May 2004; for full changes history see: `https://github.com/RobArthan/pp-contrib`.

Contents

To Do

This version now uses labelled product types rather than type abbreviations for the signature of a group.

- Implement the proposed extension to the ProofPower-HOL concrete syntax to allow infix operators with a parameter and see how it works out.
- Prove a lot more results!
- Extend the examples.
- Compare with other approaches (e.g., Elsa Gunter's).

References

[1] P.M. Cohn. *Algebra*, volume 1. John Wiley & Sons, Inc., 1974.

[2] John Harrison. Theorem Proving with the Real Numbers. Technical report, University of Cambridge Computer Laboratory, 1996.

[3] Peter V. Homeier. Quotient types. In *TPHOLs 2001: Supplemental Proceedings. Informatics Report EDI-INF-RR-0046, See* `http://www.informatics.ed.ac.uk/publications/report/`. Division of Informatics, University of Edinburgh, 2001.

[4] L. Paulson. Defining Functions on Equivalence Classes. *Preprint: available at* `http://www.cl.cam.ac.uk/users/lcp/papers/Reports/equivclasses.pdf`, 2004.

[5] W.V. Quine. *Quiddities*. Harvard University Press, 1987.

[6] LEMMA1/HOL/DTD115. *Detailed Design: Theory of Orderings*. R.D. Arthan, Lemma 1 Ltd., `rda@lemma-one.com`.

[7] LEMMA1/HOL/WRK066. *Mathematical Case Studies: Basic Analysis*. R.D. Arthan, Lemma 1 Ltd., `rda@lemma-one.com`.

[8] LEMMA1/HOL/WRK067. *Mathematical Case Studies: Some Topology*. R.D. Arthan, Lemma 1 Ltd., `rda@lemma-one.com`.

1 INTRODUCTION

This document gives specifications and proofs relating to group theory. It includes a theory of equivalence relations to support the construction of quotient groups. This is part of a series of case studies in formalising some basic pure mathematics in ProofPower-HOL. Other parts of the case study deal with real analysis [7] and with topology [8].

While the mathematical content of this document is very elementary, it does raise some interesting points about how to formalise abstract algebraic theory in polymorphic simple type theory. We want the abstract theory to be readable, general and easy to develop, we also want it to be easy to apply to specific examples.

Substructures and quotient structures in algebra are very important, so it is vital to deal smoothly with subgroups and quotient groups. Taken verbatim, the traditional explication of these concepts in set theory leads to significant notational and semantic difficulties. The problem is this: in doing the general theory, an expression like $x.y$ denoting the product of two elements of a group G actually contains three variables: the group elements 'x', 'y', and the multiplication operator '.'. Syntactic tricks allow one to preserve something like the traditional infix notation for such expressions. But there is a semantic problem when we need to deal with subgroups: according to the traditional account, the '.', in $x.y$ will denote a different set-theoretic function in H from what it does in G. Coercing operations from subgroup to containing group or from one subgroup to another becomes an excessive burden.

Our solution to this problem is to formulate all definitions relative to some carrier set of interest in such a way that the behaviour of operators or properties outside the carrier set is irrelevant. We advocate this approach in general for dealing with algebraic structures. The apparent extra complication actually achieves an economy, because when you are working with substructures, the operators and properties can all be those of the containing structure: you have no need to restrict them to the substructures or to worry about coercing the operations of one substructure into the operations of another. *Pace* Quine [5, article on "Mathematosis"], it is actually counter-productive to define the concept of a group so that the carrier set can be recovered from the set that represents the multiplication.

As an example, we define the operations on a group G to be total functions on the universe of the type of its elements whose behaviour outside the carrier set of G is immaterial. We require the operations on a subgroup H of G to be represented by the same total functions. This involves no loss of generality and removes a great deal of complexity in both specifications and proofs. It may be objected that this approach results in the wrong notion of equality between groups (since the same group can be represented using two different ways of totalising the operations). However, in normal algebraic practice, one almost never needs to assert equality between two groups that are not known to be subgroups of some other group, and in that case equality has the usual meaning.

This document is a ProofPower literate script. It contains all the metalanguage (ML) commands required to create three theories, populate them with the formal definitions and prove and record all the theorems. The three theories, "equiv_rel", "groups" and "group_egs" are described in sections 2, 3 and 4 respectively. The descriptions include all the formal definitions in the Z-like concrete syntax for specification in ProofPower-HOL. and a discussion of the theorems that have been proved about the objects specified. To keep our use of the ProofPower document preparation system simple, in this discussion we identify the theorems by name and refer the reader to the theory listings in sections A, B and C for the formal statements of the theorems. There is an index to the formal definitions and the theory listings in section 4.2.

2 EQUIVALENCE RELATIONS

The construction of quotient groups is very important in group theory and so introductory texts on the subject often begin with a review of the set-theoretic notions that support this construction, *viz.* the notion of the quotient of a set by an equivalence relation. This section contains our formal development of this material.

John Harrison [2] and Peter Homeier [3] have both produced powerful metalanguage tools for automating such constructions when a new HOL type is introduced as a quotient set. However, these tools do not fit the case in question: we will only wish to construct new types for specific quotient constructions: when we are doing general theory everything in view is a variable including the set of sets representing a quotient group.

Larry Paulson [4] has pointed out that there are advantages in providing a lemma library to support quotient constructions rather than metalanguage tools. In particular, Paulson notes that if we use a lemma library, "we are not restricted to top-level properties, but can reason about equivalence classes in a larger proof". This is precisely what happens in our development of elementary group theory. This section presents the definitions and results that make up the lemma library. The mathematics is entirely trivial: the point in formulating the theorems is not for their intrinsic interest, but to provide templates for carrying out quotient constructions in larger proofs.

2.1 Technical Prelude

First of all, we must give the the ML commands to introduce the new theory "equiv_rel" as a child of the theory "orders" of ordered sets (whence comes our definition of transitivity, see [6]).

SML

```
force_delete_theory"equiv_rel" handle Fail _ => ();
open_theory"orders";
set_merge_pcs["basic_hol1", "'sets_alg"];
new_theory"equiv_rel";
```

2.2 The Definitions

We need to define the notion of an equivalence relation, i.e., a binary relation that is transitive, reflexive and symmetric. Transitivity is defined elsewhere, but we now need the notions of reflexivity and symmetry. We follow the theory of orders in making these notions properties of set-relation pairs. We will often use the infix symbol $\widehat{=}$ as a variable ranging over binary relations, sometimes with a subscript if there are several relations involved. This symbol appears preceded by a '$' where infix notation is not being used. (If we are not going to use infix notation, we use R and S as variables ranging over relations).

SML

```
declare_infix(210, "≙");
declare_infix(210, "≙₁");
declare_infix(210, "≙₂");
```

HOL Constant

$$\boldsymbol{Refl} : ('a\ SET \times ('a \to 'a \to BOOL)) \to BOOL$$

$$\forall\ X\ \$\widehat{=}\bullet\ Refl(X, \$\widehat{=}) \Leftrightarrow \forall x \bullet x \in X \Rightarrow x\ \widehat{=}\ x$$

HOL Constant

$$\boldsymbol{Sym} : ('a\ SET \times ('a \to 'a \to BOOL)) \to BOOL$$

$$\forall\ X\ \$\widehat{=}\bullet\ Sym(X, \$\widehat{=}) \Leftrightarrow \forall x\ y\bullet x \in X \wedge y \in X \wedge x \mathrel{\widehat{=}} y \Rightarrow y \mathrel{\widehat{=}} x$$

An equivalence relation on a set X is then one which is reflexive, symmetric and transitive on X. In the traditional explication of mathematics as set theory, one requires an equivalence relation on a set X to be restricted to X. Instead, we prefer to ignore the behaviour of the relation outside X.

This means that an equivalence relation on a set X is, as it stands, an equivalence relation on any subset of X. This works well in the formal treatment and works well informally too. For example, the relation that holds between two numbers x and y when $x - y$ is an integer, is an equivalence relation on any subset of the real numbers. There seems to be no conceptual or practical gain in treating such a relation as having a different set-theoretic representation for different subsets.

HOL Constant

$$\boldsymbol{Equiv} : ('a\ SET \times ('a \to 'a \to BOOL)) \to BOOL$$

$$\forall\ X\ \$\widehat{=}\bullet\ Equiv(X, \$\widehat{=}) \Leftrightarrow Refl(X, \$\widehat{=}) \wedge Sym(X, \$\widehat{=}) \wedge Trans(X, \$\widehat{=})$$

Now we can define the notions of equivalence class and the quotient set (which is the set of all equivalence classes). There is no need to stipulate that the relations be equivalence relations in these definitions — that is done in the statements of the theorems about them. We arrange things so that if $\widehat{=}$ is an equivalence relation on X, the function $EquivClass(X,\ \$\widehat{=})$ is the projection of X onto the quotient set.

HOL Constant

$$\boldsymbol{EquivClass} : ('a\ SET \times ('a \to 'a \to BOOL)) \to 'a \to 'a\ SET$$

$$\forall\ X\ \$\widehat{=}\ x\bullet\ EquivClass(X, \$\widehat{=})\ x = \{y \mid y \in X \wedge x \mathrel{\widehat{=}} y\}$$

HOL Constant

$$\boldsymbol{QuotientSet} : 'a\ SET \to ('a \to 'a \to BOOL) \to 'a\ SET\ SET$$

$$\forall\ X\ \$\widehat{=}\bullet\ QuotientSet\ X\ (\$\widehat{=}) = \{A \mid \exists x\bullet\ x \in X \wedge A = EquivClass(X, \$\widehat{=})\ x\}$$

We introduce an alias to let us write $X\ /\ \widehat{=}$ for the quotient set. This overloads arithmetic division (and will be further overloaded later for the quotient group construction).

SML

```
declare_infix(300, "/");
declare_alias("/", ⌜QuotientSet⌝);
```

We say a function f respects an equivalence relation $\widehat{=}$ on X, iff. the function does not distinguish between related values:

SML

```
declare_infix(200, "Respects");
```

HOL Constant

$$\$\boldsymbol{Respects} : ('a \to 'b) \to ('a \to 'a \to BOOL) \to 'a\ SET \to BOOL$$

$$\forall\ f\ \$\widehat{=}\ X\bullet\ (f\ Respects\ \$\widehat{=})\ X \Leftrightarrow \forall x\ y\bullet x \in X \wedge y \in X \wedge x\ \widehat{=}\ y \Rightarrow f\ x = f\ y$$

We say that a relation, R, refines another, S iff. R-equivalence implies S-equivalence. In other words, iff. each R-equivalence class is contained in an S-equivalence class.

SML

```
declare_infix(200, "Refines");
```

HOL Constant

$$\$\boldsymbol{Refines} :\ ('a \to 'a \to BOOL) \to ('a \to 'a \to BOOL) \to 'a\ SET \to BOOL$$

$$\forall\ X\ \$\widehat{=}_1\ \$\widehat{=}_2 \bullet\ (\$\widehat{=}_1\ Refines\ \$\widehat{=}_2)\ X \Leftrightarrow \forall x\ y\bullet x \in X \wedge y \in X \wedge x\ \widehat{=}_1\ \ y \Rightarrow x\ \widehat{=}_2\ y$$

Following Paulson, we define a function *Contents* whose value on a singleton set $\{x\}$ is x (and whose value on any other kind of set is unspecified). The consistency of this definition is not proved automatically, and the development of the theorems begins with the easy proof that it is consistent.

HOL Constant

$$\boldsymbol{Contents} : 'a\ SET \to 'a$$

$$\forall\ x\bullet\ Contents\ \{x\} = x$$

Given any function $f : X \to Y$, the relation, R_f, say, defined so that $x\,R_f\,y$ holds iff. $f(x) = f(y)$ is an equivalence relation. f respects an equivalence relation R iff. R refines R_f. Given such an R, f induces a function, $\overline{f}$, from the quotient of X/R to Y. The following function comprises the union over all R of the corresponding $\overline{f}$ extended to a total function in an unspecified way. We write it using the postfix notation $f\ ^-$.

SML

```
declare_postfix(330, "⁻");
```

HOL Constant

$$\$^- : ('a \to 'b) \to 'a\ \ SET \to 'b$$

$$\forall\ f\ \ A\bullet\ (f\ ^-)\ A = Contents\{y \mid \exists x\bullet\ \ x \in A \wedge y = f\ x\}$$

Our lemma library begins with the consistency of the contents function and then a handful of simple facts about equivalence relations, equivalence classes and the contents function. This comprises the following theorems.

Contents_consistent
contents_def
equiv_class_eq_thm
equiv_class_∈_thm
constant_img_thm
respects_img_thm
respects_img_contents_thm
quotient_map_onto_thm
quotient_∈_thm
quotient_rep_∃_thm
equiv_mono_thm

The following theorem says that if f respects S and R refines S, then f respects R. This theorem is quite trivial, but provides a useful pattern for proving that a function f respects a relation R: find some coarser relation S that f is known to respect and apply this theorem.

respects_refines_thm

The next block of theorems begins with two theorems which show, in effect, that given any function, $f : X \to Y$, and any relation $\widehat{=}$ that f respects on X, f factors through the projection of X onto $X \;/\; \widehat{=}$, the induced function from $X \;/\; \widehat{=}$ to Y being given by f^{-}. The first version states this in terms of equivalence classes and the second in terms of members of a quotient set. The third theorem in this block is our alternative to the treatment of dyadic functions suggested by Paulson. It shows that the operation $\lambda f \bullet f^{-}$ can be iterated to produced a curried version of the induced function theorem for functions of two arguments.

induced_fun_equiv_class_thm *induced_fun_thm* *induced_fun_induced_fun_thm*

Finally we list the theorems that act as the main "external interface" to the lemma library. These give the characterising properties of the quotient set construction as pure existence theorems for the one-argument and two-argument cases.

The characterising properties are that the induced functions exist under the appropriate hypotheses and are unique. The uniqueness is trivial and is only stated formally for completeness. In applications of the lemma library, one will typically just prove instances of the hypotheses for a particular function and a particular equivalence relation (or relations) and forward chain with the existence theorems to give the induced function. In this sense, these theorems act as templates for constructing induced functions.

induced_fun_∃_thm
induced_fun_∃_unique_thm
dyadic_induced_fun_∃_thm
dyadic_induced_fun_∃_unique_thm

In fact, while the two-argument version was used in an earlier treatment of quotient groups, it has since turned out to be better to use the group-theoretic product of two sets for this construction. In general in developing a typical algebraic theory (e.g., rings, modules over a ring, vector spaces over a field), the induced function existence theorem will typically be used to prove the version of the first isomorphism theorem that is appropriate to that theory. Thereafter, the first isomorphism theorem will generally replace uses of the induced function existence theorem, because it gives functions that are morphisms of the theory, not just set-theoretic functions.

3 GROUPS

3.1 Technical Prelude

First of all, we must give the the ML commands to introduce the new theory "groups" as a child of the theory "equiv_rel" of equivalence relations.

SML

```
force_delete_theory"groups" handle Fail _ => ();
open_theory"equiv_rel";
new_theory"groups";
set_merge_pcs["basic_hol1", "'sets_alg"];
new_parent "fincomb";
```

3.2 The Signature of a Group

We will represent a group as a quadruple comprising a carrier set, a two-argument multiplication function, a unit element and a one-argument inverse function. This signature is captured in the following labelled product type definition, parametrised by the type variable $'a$ giving the type of the elements of the group.

HOL Labelled Product

GROUP

$\mathbf{Car_G}$	$: 'a\ SET$;
$\mathbf{Times_G}$	$: 'a \to 'a \to 'a$;
$\mathbf{Unit_G}$	$: 'a$;
$\mathbf{Inverse_G}$	$: 'a \to 'a$

If G is a structure with the above signature (i.e., a member of an instance of the above type), we write $Car\ G$ for the carrier set, $(x.y)G$ for the product of two elements, x and y, $Unit\ G$ for the unit element and $(x\ ^\sim)G$ for the inverse of an element, x. This is achieved by the following fixity declarations and definition of access functions for the signature.

SML

```
declare_infix(310, ".");
declare_postfix(330, "~");
```

HOL Constant

$$
\begin{array}{l}
\mathbf{Car} : 'a\ GROUP \to 'a\ SET; \\
\mathbf{\$.} : 'a \to 'a \to 'a\ GROUP \to 'a; \\
\mathbf{\$\sim} : 'a \to 'a\ GROUP \to 'a; \\
\mathbf{Unit} : 'a\ GROUP \to 'a \\
\hline
\forall\ set\ times\ one\ inverse \bullet \\
\quad Car\ (MkGROUP\ set\ times\ one\ inverse) = set \\
\wedge \quad (\forall x\ y \bullet (x\ .\ y)\ (MkGROUP\ set\ times\ one\ inverse) = times\ x\ y) \\
\wedge \quad Unit\ (MkGROUP\ set\ times\ one\ inverse) = one \\
\wedge \quad (\forall x \bullet (x\ ^\sim)\ (MkGROUP\ set\ times\ one\ inverse) = inverse\ x)
\end{array}
$$

[**Aside**: For the future: one may want the above with *Car* and *Unit* as aliases, but with the defining property as above proved as a theorem to give the general purpose rewrite rule to use in reasoning about these aliases and the defined constants for multiplication and inverse.]

We prove one theorem about these which is just a convenience for proving that two structures for this signature are equal.

group_eq_group_thm

3.3 The Group Laws and Equational Reasoning in a Group

We can now specify the group laws. The polymorphic set *Group* comprises all structures with the signature of a group that satisfy the group laws. The statement is entirely standard following our convention of relativising everything to the carrier set of the group. The first two conditions on G say that the carrier set is closed under multiplication and that multiplication is associative, the next two conditions say that the unit is a member of G and is indeed a two-sided unit for multiplication. The remaining conditions say that G is closed under inverse and that the inverse does indeed give a two-sided inverse for the multiplication.

HOL Constant

$$\boldsymbol{Group} : 'a\ GROUP\ SET$$

$$\begin{array}{ll}
\forall\ G\bullet & \\
\quad & G \in Group \\
\Leftrightarrow & (\forall x\ y\bullet\ x \in Car\ G \wedge y \in Car\ G \Rightarrow (x\ .\ y)\ G \in Car\ G) \\
\wedge & (\forall x\ y\ z\bullet\ x \in Car\ G \wedge y \in Car\ G \wedge z \in Car\ G \Rightarrow ((x\ .\ y)G\ .\ z)G = (x\ .\ (y\ .\ z)G)G) \\
\wedge & Unit\ G \in Car\ G \\
\wedge & (\forall x\bullet\ x \in Car\ G \Rightarrow (x\ .\ Unit\ G)\ G = x \wedge (Unit\ G\ .\ x)G = x) \\
\wedge & (\forall x\bullet\ x \in Car\ G \Rightarrow (x\ ^{\sim})\ G \in Car\ G) \\
\wedge & (\forall x\bullet\ x \in Car\ G \Rightarrow (x\ .\ (x\ ^{\sim})G)G = Unit\ G \wedge ((x\ ^{\sim})G\ .\ x)G = Unit\ G)
\end{array}$$

The above definition shows that our approach to the syntax of the group operations is not unworkable. The syntax is certainly readable if one pretends not to notice all the "G"s. When working with a specific group, the definitions can easily be expanded to give the familiar notations for the group (see examples in section 4).

However, the syntax is not particular convenient to write for complex expressions, mainly because it forces the writer to write all the brackets in an expression explicitly. An extension to the concrete syntax of ProofPower-HOL is being considered which would remedy this. The extension would allow a form of ternary infix operator. What we are currently writing as $(x.y)G$ would become $x\,.G\,y$. One would be allowed to write $x\,.G\,y.G\,z$ with no brackets. Brackets would only be required when they are significant.

On the basis of this definition, we can prove the usual elementary consequences of these laws. These are presented as a set of four portmanteau theorems (*group_clauses1* ... *group_clauses4*) together with some particular results such as a cancellation law that are needed to bootstrap the theory and prove the portmanteau theorems.

group_clauses1
group_clauses2
group_eq_thm
group_eq_thm1
left_cancel_thm
inverse_inverse_thm
group_clauses3
times_inverse_thm
inverse_unique_thm
group_clauses4
unit_unique_thm

Taken together, the portmanteau theorems provide the closure statements one needs to show that arbitrary combinations of the group operations applied to members of the group give members of the group and give the "free group normal form" for expressions over the signature of the group. This normal form is obtained by repeatedly rewriting with the following rules:

$$
\begin{array}{lrcl}
(1) & (x.y)^{-1} & = & y^{-1}.x^{-1} \\
(2) & (x^{-1})^{-1} & = & x \\
(3) & x.x^{-1} & = & 1 \\
(4) & x^{-1}.x & = & 1 \\
(5) & x.(x^{-1}.z) & = & z \\
(6) & x^{-1}.(x.z) & = & z \\
(7) & (x.y).z & = & x.(y.z) \\
(8) & x.1 & = & x \\
(9) & 1.x & = & x \\
(10) & 1^{-1} & = & 1
\end{array}
$$

Now for us, these are conditional rewrite rules: they only hold if the operands of the expressions on the left are members of the carrier set of the group. At this point in the proof script, we use the portmanteau theorems to implement automated proof procedures for normalising expressions in the operations of a group and for membership of the carrier set of a group. The proof procedures for normalisation apply the above rewrite rules setting the necessary membership conditions as lemmas. The proof procedures for membership apply the closure conditions for the operations to simplify membership conditions on complex expressions into conditions on atomic subexpressions. These procedures deal automatically with all of the equational reasoning that will be needed later except for providing existential witnesses and identifying points at which the argument for a membership condition is non-trivial (i.e., does not follow just from membership conditions on atomic subexpressions).

3.4 Homomorphisms

The definition of homomorphism between two groups, G and H, is completely standard For us, a homomorphism is given by a total function whose behaviour outside the carrier set of G is irrelevant.

HOL Constant

$$
\begin{array}{l}
\boldsymbol{Homomorphism} : 'a\ GROUP \times 'b\ GROUP \to ('a \to 'b)\ SET \\
\hline
\forall\ G\ H\ f \bullet \\
\quad f \in Homomorphism(G,\ H) \\
\Leftrightarrow \quad (\forall x \bullet x \in Car\ G \Rightarrow f\ x \in Car\ H) \\
\wedge \quad (\forall x\ y \bullet x \in Car\ G \wedge y \in Car\ G \Rightarrow f((x\ .\ y)\ G) = (f\ x\ .\ f\ y) H)
\end{array}
$$

One only needs to specify that a homomorphism preserves the multiplication, since a homomorphism in that sense will automatically preserve the unit element and inverses as shown by the first two theorems in the following block. The third and fourth present these facts and the properties in the definition in a convenient form.

homomorphism_unit_thm *homomorphism_clauses*
homomorphism_inverse_thm *homomorphism_∈_car_thm*

3.5 Subgroups

Our definition of a subgroup is standard except that we require the multiplication and the inverse function of the subgroup to be identical with those in the containing group. This is invaluable in simplifying later definitions and in stating and proving theorems. It guarantees that one can just use the operations of the containing group wherever appropriate. The reader who does not like this is cordially invited to replace the last two equations in the following by equations conditional on membership of the carrier set of H and then to prove the theorems that follow.

HOL Constant

$$\boldsymbol{Subgroup} : 'a\ GROUP \rightarrow 'a\ GROUP\ SET$$

$$\begin{array}{ll} \forall\ G\ H\bullet & \\ \quad H \in Subgroup\ G & \\ \Leftrightarrow \quad Car\ H \subseteq Car\ G & \\ \wedge \quad H \in Group & \\ \wedge \quad (\forall x\ y\bullet\ (x.y)H = (x.y)G) & \\ \wedge \quad (\forall x\bullet\ (x\ ^{\sim})H = (x\ ^{\sim})G) & \end{array}$$

The following theorem extends the equations in the above definition to add the statement that the unit element of a subgroup is the same as that of the containing group.

subgroup_clauses

The unit subgroup of a group is the subgroup whose carrier set comprises only the unit element.

HOL Constant

$$\boldsymbol{UnitSubgroup} : 'a\ GROUP \rightarrow 'a\ GROUP$$

$$\forall\ G\bullet$$
$$\quad UnitSubgroup\ G = MkGROUP\ \{Unit\ G\}\ (\lambda x\ y\bullet(x.y)G)\ (Unit\ G)\ (\lambda x\bullet(x\ ^{\sim})G)$$

If G is a group then G itself and its unit subgroup are both subgroups of G. The relation of being a subgroup is transitive. If K and H are subgroups of G and the carrier set of K is contained in that of H, then K is subgroup of H. Two subgroups are equal iff. their carrier sets are equal.

trivial_subgroups_thm *subgroup_⊆_subgroup_thm*
subgroup_trans_thm *subgroup_eq_thm*

The inclusion of a subgroup, H, of G, is a homomorphism from H to G In particular, the identity function on G is a homomorphism from G to G. The function which map every element of some group to the unit of some other group is a homomorphism. The composite of two homomorphisms is a homomorphism.

subgroup_homomorphism_thm *unit_homomorphism_thm*
id_homomorphism_thm *comp_homomorphism_thm*

If A is a subset of the carrier set of a group G, we write $A \cap G$ for the structure with the same operations as G and with carrier set the intersection of A and the carrier set of G. In the narrative and in the names of theorems we refer to this as the restriction of G to A.

HOL Constant

$$\boldsymbol{Restriction} : 'a\ SET \to 'a\ GROUP \to 'a\ GROUP$$

$$\forall\ A\ G\bullet\ Restriction\ A\ G = MkGROUP\ (A \cap Car\ G)\ (\lambda x\ y\bullet\ (x.y)G)\ (Unit\ G)\ (\lambda x\bullet(x\ ^\sim)G)$$

SML

```
declare_alias("∩", ⌜Restriction⌝);
```

$A \cap G$ is a group iff. A contains the unit element and is closed under multiplication and inverse.

restriction_subgroup_thm

3.6 Normal Subgroups and Kernels of Homomorphisms

A normal subgroup is one that is closed under conjugation. (The conjugate of x by y is $y^{-1}.x.y$).

HOL Constant

$$\boldsymbol{NormalSubgroup} : 'a\ GROUP \to 'a\ GROUP\ SET$$

$$\begin{array}{ll}\forall\ G\ H\bullet & \\ & H \in NormalSubgroup\ G \\ \Leftrightarrow & H \in Subgroup\ G \\ \wedge & (\forall x\ y\bullet\ x \in Car\ H \wedge y \in Car\ G \Rightarrow ((y\ ^\sim)G.(x.y)G)G \in Car\ H)\end{array}$$

The kernel of a homomorphism from G to H has as its carrier set the pre-image of the unit element of H and inherits the group operations from G:

HOL Constant

$$\boldsymbol{Ker} : ('a \to 'b) \to 'a\ GROUP \times 'b\ GROUP \to 'a\ GROUP$$

$$\begin{array}{ll}\forall\ f\ G\ H\bullet & \\ & Ker\ f\ (G,\ H) \\ = & MkGROUP\ \{x \mid x \in Car\ G \wedge f\ x = Unit\ H\}\ (\lambda x\ y\bullet\ (x\ .\ y)G)\ (Unit\ G)\ (\lambda x\bullet(x\ ^\sim)\ G)\end{array}$$

Kernels of homomorphisms are normal subgroups as stated in the following theorem. If a normal subgroup, K, of G is a subgroup of a subgroup H, then it is normal in H.

ker_normal_subgroup_thm *subgroup_normal_subgroup_thm*

3.7 Congruence modulo a Subgroup

Elements x and y are (right) congruent modulo a subgroup H iff. $x^{-1}y$ is a member of H. Right congruence is equivalence modulo right translation by elements of H.

HOL Constant

$$\boldsymbol{RightCongruent} :'a\ GROUP \to 'a\ GROUP \to 'a \to 'a \to BOOL$$

$$\forall\ G\ H\ x\ y\bullet\ RightCongruent\ H\ G\ x\ y \Leftrightarrow ((x\ ^{\sim})\ G\ .\ y)\ G \in Car\ H$$

The following theorems states that right congruence is an equivalence relation:

right_congruent_equiv_thm

3.8 Operations on Sets

If A and B are any subsets of the (universe of the) carrier set of a group, we will write $(A.B)G$ for the set of all $(a.b)G$ as a ranges over A and b ranges over B. We will also write $(x.B)G$ and $(A.y)G$ for $(\{x\}.B)G$ and $(A.\{y\})G$ respectively. Similarly we will write $(A\ ^{\sim})G$ for the set of all inverse of elements of A. This overloads '.', and '~', so we define these operations using other names and then introduce aliases:

HOL Constant

$$\boldsymbol{SetTimesSet} :'a\ SET \to 'a\ SET \to 'a\ GROUP \to 'a\ SET;$$
$$\boldsymbol{ElemTimesSet} :'a \to 'a\ SET \to 'a\ GROUP \to 'a\ SET;$$
$$\boldsymbol{SetTimesElem} :'a\ SET \to 'a \to 'a\ GROUP \to 'a\ SET;$$
$$\boldsymbol{SetInverse} :'a\ SET \to 'a\ GROUP \to 'a\ SET$$

$$\begin{array}{ll} & (\forall G\ A\ B\bullet\ SetTimesSet\ A\ B\ G = \{z \mid \exists a\ b\bullet a \in A \wedge b \in B \wedge z = (a.b)G\}) \\ \wedge & (\forall G\ x\ B\bullet\ ElemTimesSet\ x\ B\ G = \{z \mid \exists b\bullet b \in B \wedge z = (x.b)G\}) \\ \wedge & (\forall G\ A\ y\bullet\ SetTimesElem\ A\ y\ G = \{z \mid \exists a\bullet a \in A \wedge z = (a.y)G\}) \\ \wedge & (\forall G\ A\bullet\ SetInverse\ A\ G = \{z \mid \exists a\bullet a \in A \wedge z = (a\ ^{\sim})G\}) \end{array}$$

SML

```
declare_alias(".", ⌜SetTimesSet⌝);
declare_alias(".", ⌜ElemTimesSet⌝);
declare_alias(".", ⌜SetTimesElem⌝);
declare_alias("~", ⌜SetInverse⌝);
```

If H and K are groups with elements of the same type, we will write $H.K$ for the structure with the set product of the carrier sets as its carrier set under the operations of H (which will agree with those of K under our conventions if they are both subgroups of some other group).

HOL Constant

$$\boldsymbol{GroupTimesGroup} :'a\ GROUP \to 'a\ GROUP \to 'a\ GROUP$$

$$\begin{array}{l} \forall H\ K\bullet \\ \quad GroupTimesGroup\ H\ K = \\ \quad MkGROUP\ ((Car\ H.Car\ K)H)\ (\lambda x\ y\bullet(x.y)H)\ (Unit\ H)\ (\lambda x\bullet(x\ ^{\sim})H) \end{array}$$

SML

```
declare_alias(".", ⌜GroupTimesGroup⌝);
```

The product of two subgroups can be expressed as a restriction. The product is itself a subgroup if either of the two subgroups is normal. We just prove this for the case when the second subgroup is normal.

group_product_restriction_thm *group_product_subgroup_thm*

3.9 Cosets

If H is a subgroup of G and x is an element of G, the right coset generated by x is the set $x.H$ and the left coset generated by x is $H.x$. I.e., perhaps confusingly, a right coset is a left translate of H. Some texts call these left cosets (and then, perhaps confusingly, left congruence is equivalence modulo right translation by elements of H). Our terminology follows [1].

The following theorems state that the right cosets of a subgroup are the equivalence classes of its right congruence relation and that the right cosets generated by elements x and y are equal iff. x and y are right congruent.

right_coset_equiv_class_thm
right_coset_eq_thm

3.10 The Quotient Group Construction

If G is a group and H is a normal subgroup, then the multiplication and inverse operations on G induce a multiplication and inverse operation on the set of right cosets of H which make it into a group in such a way that the projection onto the set of right cosets is a homomorphism. Following our approach of embedding the operations of an algebraic structure in the most useful general extension to a total function, we make the following definition, which embeds the quotient group in the monoid of all subsets of G under the multiplication of sets induced by the multiplication of G. This monoid contains all quotient groups of G as submonoids (or, more precisely, it embeds it in the structure whose carrier set comprises all subsets of the universe of G under the induced multiplication; this structure is not in general a monoid, but becomes one if its carrier set is restricted to subsets of G).

HOL Constant

$$\textbf{\textit{QuotientGroup}} : 'a\ GROUP \to 'a\ GROUP \to 'a\ SET\ GROUP$$

$$\forall\ G\ H \bullet\ QuotientGroup\ G\ H = MkGROUP$$
$$\{A \mid \exists x \bullet x \in Car\ G \wedge A = (x.Car\ H)G\}$$
$$(\lambda A\ B \bullet (A.B)G)$$
$$((Unit\ G.Car\ H)G)$$
$$(\lambda A \bullet\ (A\ ^{\sim})G)$$

We write G/H for the quotient of G by H by dint of the following alias declaration (overloading the alias for the quotient set operator).

SML

```
declare_alias("/", ⌜QuotientGroup⌝);
```

We show that if H is a normal subgroup of G, then the product and inverse operations of the quotient group are represented by the product and inverse operations of the group, i.e., that $(x.H).(y.H) = (xy.H)$ and $(x.H)^{-1} = (x^{-1}.H)$. From this it follows easily that the projection to the quotient of G by a normal subgroup H is surjective, that the quotient group is indeed a group under the induced operations and that the projection is a homomorphism with kernel H.

quotient_group_times_thm
quotient_group_group_thm
quotient_group_inverse_thm
quotient_group_homomorphism_thm
quotient_group_rep_∃_thm
ker_right_coset_thm

3.11 Images of Homomorphisms

The following image group construction is needed in the statement of the first isomorphism theorem.

HOL Constant

```
Img : ('a → 'b) → 'a SET → 'b GROUP  → 'b GROUP
──────────────────────────────────────────────
∀ f X G•
    Img f X G =
    MkGROUP {y | ∃x•x ∈ X ∧ y = f x} (λx y• (x.y)G) (Unit G) (λx• (x ~)G)
```

The next theorem states that the image of a group homomorphism is a subgroup of the range of the homomorphism:

img_subgroup_thm

3.12 Isomorphisms

An isomorphism is a one-to-one onto homomorphism

HOL Constant

```
Isomorphism : 'a GROUP × 'b GROUP → ('a → 'b) SET
──────────────────────────────────────────────
∀ G H f•
    f ∈ Isomorphism(G, H)
⇔   f ∈ Homomorphism(G, H)
∧   (∀x y• x ∈ Car G ∧ y ∈ Car G ∧ f x = f y ⇒ x = y)
∧   (∀z• z ∈ Car H ⇒ ∃x•x ∈ Car G ∧ f x = z)
```

The following theorems lead up to the proof of the first isomorphism theorem (which says that if $f : G \to H$ is a homomorphism with kernel K, then f factors through a homomorphism $H/K \to G$ which gives an isomorphism between H/K and the image of f). The proofs of earlier results have

made some use of the equivalence class lemma library, but this is where the result on induced functions is first used in anger.

mage_subgroup_thm
equiv_right_congruent_ker_thm
homomorphism_respects_ker_thm
car_quotient_group_thm
subgroup_refines_thm
subgroup_ker_induced_thm
isomorphism_ker_img_thm
first_isomorphism_thm

The second isomorphism theorem says that if H is a subgroup of G and K is a normal subgroup of G, then $H \cap K$ is a normal subgroup of H and the quotient $H/H \cap K$ is isomorphic to the quotient $H.K/K$. We prove this in a series of lemmas.

second_isomorphism_lemma1
second_isomorphism_lemma2
second_isomorphism_lemma3
second_isomorphism_lemma4
second_isomorphism_thm

We also prove the third isomorphism theorem, which says that if H and K are normal subgroups of a group G, and K is a subgroup of H, then there is an isomorphism between G/H and $(G/K)/(H/K)$. As with the second isomorphism theorem, we sneak up on this in a series of lemmas.

third_isomorphism_lemma1
third_isomorphism_lemma2
third_isomorphism_lemma3
third_isomorphism_lemma4
third_isomorphism_lemma5
third_isomorphism_thm

3.13 The Symmetric Group

The symmetric group on a set X is the group of all one-to-one onto functions from X to X. In a typed context, it is convenient to turn this round and say that it is the group of all functions from the universe of the type of the elements of X to itself that are one-to-one and onto and that fix anything not in X. This has several technical advantages and involves no loss of generality. Before giving the definition, we need to define the notion of an inverse function. (This definition probably belongs elsewhere in the theory hierarchy.)

HOL Constant

$$\boldsymbol{Inverse} : ('a \to 'b) \to ('b \to 'a)$$

$$\forall\, f \bullet\ OneOne\ f \wedge Onto\ f \Rightarrow (\forall x \bullet Inverse\ f\ (f\ x) = x) \wedge (\forall y \bullet f(Inverse\ f\ y) = y)$$

HOL Constant

$$\boldsymbol{SymGroup} : 'a\ SET \to ('a \to 'a)\ GROUP$$

$$\begin{aligned} \forall\, X \bullet \quad & SymGroup\ X = MkGROUP \\ & \{f \mid OneOne\ f \wedge Onto\ f \wedge \forall y \bullet \neg y \in X \Rightarrow f\ y = y\} \\ & (\lambda f\ g \bullet \lambda x \bullet f(g\ x)) \\ & (\lambda x \bullet\ x) \end{aligned}$$

Inverse

The symmetric group on any set is indeed a group. Moreover, if G is any group, G is isomorphic to a subgroup of the symmetric group on its carrier set, which is the Cayley representation theorem.

sym_group_group_thm *cayley_thm*

3.14 Finite Groups

If G is a group (or any structure with the same signature as a group), we will write $\#(G)$ for the number of elements of the carrier set of G. To allow this, we make $\#$ an alias of the following function:

HOL Constant

$$\boldsymbol{Size_G} : 'a\ GROUP \to \mathbb{N}$$

$$\forall\ G\bullet\ Size_G\ G = \#(Car\ G)$$

SML

```
declare_alias("#", ⌜Size_G⌝);
```

We prove the theorem of Lagrange that if G is a finite group, and H is a subgroup of G, then H and the set of cosets G/H are both finite and $\#(G) = \#(H) \times \#(G/H)$.

finite_subgroup_thm *finite_cosets_thm* *lagrange_cosets_thm*

3.15 Cartesian Product

The Cartesian product of two groups is the standard pointwise construction:

SML

```
declare_infix(290, "×_G");
```

HOL Constant

$$\$\times_{\boldsymbol{G}} : 'a\ GROUP \to 'b\ GROUP \to ('a \times 'b)\ GROUP$$

$$\begin{array}{l}\forall G\ H\bullet\quad (G \times_G H) = \\ \quad MkGROUP \\ \qquad (Car\ G \times Car\ H) \\ \qquad (\lambda(a,\ b)(c,\ d)\bullet\ ((a.c)G,\ (b.d)H)) \\ \qquad (Unit\ G,\ Unit\ H) \\ \qquad (\lambda(a,\ b)\bullet((a\ ^\sim)G,\ (b\ ^\sim)H))\end{array}$$

The Cartesian product of two groups is again a group; the projections onto the factors of the products are homomorphisms as is the lift of a pair of homomorphism into the product.

product_group_thm *fst_homomorphism_thm* *snd_homomorphism_thm*
product_homomorphism_thm

4 Examples of Groups

In this section we give one or two examples of specific groups. We specialise some of the theorems from section 3 to show how they look.

4.1 Technical Prelude

First of all, we must give the the ML commands to introduce the new theory "group_egs" as a child of the theory 'groups" of groups and the theory "analysis" of real analysis.

SML

```
force_delete_theory"group_egs" handle Fail _ => ();
open_theory"groups";
new_theory"group_egs";
new_parent"analysis";
set_merge_pcs["basic_hol1", "'sets_alg", "'ℤ", "'ℝ"];
```

4.2 The Examples

We define the group of integers under addition and the group of unit integers under multiplication:

HOL Constant

$$\begin{array}{l} \mathbb{Z}\_\boldsymbol{plus} : \mathbb{Z}\ GROUP; \\ \mathbb{Z}\_\boldsymbol{units} : \mathbb{Z}\ GROUP \\ \hline \quad \mathbb{Z}\_plus = MkGROUP\ Universe\ \$+\ (\mathbb{NZ}\ 0) \sim \\ \wedge \quad \mathbb{Z}\_units = MkGROUP\ \{\sim(\mathbb{NZ}\ 1);\ \mathbb{NZ}\ 1\}\ \$*\ (\mathbb{NZ}\ 1)\ (\lambda x \bullet x) \end{array}$$

We prove that both of these are indeed groups. The notational devices of our treatment of abstract group theory convert easily into the familiar notation for specific groups. As an example of this, we show how the definition of a homomorphism specialises to the notion of a homomorphism from $\mathbb{Z}\_plus$ to $\mathbb{Z}\_units$.

$\mathbb{Z}\_plus\_group\_thm$
$\mathbb{Z}\_units\_group\_thm$
$\mathbb{Z}\_plus\_ops\_thm$
$\mathbb{Z}\_units\_ops\_thm$
$\mathbb{Z}\_plus\_\mathbb{Z}\_units\_homomorphism\_def$
$\mathbb{Z}\_plus\_\mathbb{Z}\_units\_homomorphism\_unit\_thm$
$\mathbb{Z}\_plus\_\mathbb{Z}\_units\_homomorphism\_inverse\_thm$

We now define the group of reals under addition and the group of positive reals under multiplication:

HOL Constant

$$\begin{array}{l} \mathbb{R}_+ : \mathbb{R}\ GROUP; \\ \mathbb{R}\_\boldsymbol{pos} : \mathbb{R}\ GROUP \\ \hline \quad \mathbb{R}_+ = MkGROUP\ Universe\ \$+\ (\mathbb{NR}\ 0) \sim \\ \wedge \quad \mathbb{R}\_pos = MkGROUP\ \{x \mid \mathbb{NR}\ 0 < x\}\ \$*\ (\mathbb{NR}\ 1)\ \$^{-1} \end{array}$$

As with the groups of integers, we show that these are groups and instantiate various definitions to them. We can then show that the function exp provides an isomorphism between the two groups. We also show that a linear mapping is an additive homomorphism from $\mathbb{R}$ to itself and that addition is an additive homomorphism from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$.

$\mathbb{R}\_additive\_group\_thm$
$\mathbb{R}\_pos\_group\_thm$
$\mathbb{R}\_additive\_ops\_thm$
$\mathbb{R}\_pos\_ops\_thm$
$\mathbb{R}\_additive\_\mathbb{R}\_pos\_homomorphism\_def$
$plus\_\mathbb{R}\_additive\_homomorphism\_thm$
$\mathbb{R}\_additive\_\mathbb{R}\_pos\_homomorphism\_unit\_thm$
$\mathbb{R}\_additive\_\mathbb{R}\_pos\_homomorphism\_inverse\_thm$
$\mathbb{R}\_additive\_\mathbb{R}\_pos\_isomorphism\_def$
$exp\_isomorphism\_thm$
$linear\_homomorphism\_thm$

A THE THEORY equiv_rel

A.1 Parents

orders

A.2 Children

groups

A.3 Constants

Refl	$'a\ \mathbb{P} \times ('a \to 'a \to BOOL) \to BOOL$
Sym	$'a\ \mathbb{P} \times ('a \to 'a \to BOOL) \to BOOL$
Equiv	$'a\ \mathbb{P} \times ('a \to 'a \to BOOL) \to BOOL$
EquivClass	$'a\ \mathbb{P} \times ('a \to 'a \to BOOL) \to 'a \to 'a\ \mathbb{P}$
QuotientSet	$'a\ \mathbb{P} \to ('a \to 'a \to BOOL) \to 'a\ \mathbb{P}\ \mathbb{P}$
$\$$***Respects***	$('a \to 'b) \to ('a \to 'a \to BOOL) \to 'a\ \mathbb{P} \to BOOL$
$\$$***Refines***	$('a \to 'a \to BOOL) \to ('a \to 'a \to BOOL) \to 'a\ \mathbb{P} \to BOOL$
Contents	$'a\ \mathbb{P} \to 'a$
$\$^{-}$	$('a \to 'b) \to 'a\ \mathbb{P} \to 'b$

A.4 Aliases

/	$QuotientSet : 'a\ \mathbb{P} \to ('a \to 'a \to BOOL) \to 'a\ \mathbb{P}\ \mathbb{P}$

A.5 Fixity

Right Infix 200:
Refines ***Respects***

Right Infix 210:
$\widehat{=}$ $\widehat{=}_1$ $\widehat{=}_2$

Right Infix 300:
/

Postfix 330:
$^{-}$

A.6 Definitions

Refl $\vdash \forall\ X\ \$\widehat{=}\bullet\ \mathit{Refl}\ (X, \$\widehat{=}) \Leftrightarrow (\forall\ x\bullet\ x \in X \Rightarrow x\ \widehat{=}\ x)$

Sym $\vdash \forall\ X\ \$\widehat{=}$
$\bullet\ \mathit{Sym}\ (X, \$\widehat{=})$
$\Leftrightarrow (\forall\ x\ y\bullet\ x \in X \land y \in X \land x\ \widehat{=}\ y \Rightarrow y\ \widehat{=}\ x)$

Equiv $\vdash \forall\ X\ \$\widehat{=}$
$\bullet\ \mathit{Equiv}\ (X, \$\widehat{=})$
$\Leftrightarrow \mathit{Refl}\ (X, \$\widehat{=}) \land \mathit{Sym}\ (X, \$\widehat{=}) \land \mathit{Trans}\ (X, \$\widehat{=})$

EquivClass $\vdash \forall\ X\ \$\widehat{=}\ x\bullet\ \mathit{EquivClass}\ (X, \$\widehat{=})\ x = \{y | y \in X \land x\ \widehat{=}\ y\}$

QuotientSet
$\vdash \forall\ X\ \$\widehat{=}$
$\bullet\ X\ /\ \$\widehat{=} = \{A | \exists\ x\bullet\ x \in X \land A = \mathit{EquivClass}\ (X, \$\widehat{=})\ x\}$

Respects $\vdash \forall\ f\ \$\widehat{=}\ X$
$\bullet\ (f\ \mathit{Respects}\ \$\widehat{=})\ X$
$\Leftrightarrow (\forall\ x\ y\bullet\ x \in X \land y \in X \land x\ \widehat{=}\ y \Rightarrow f\ x = f\ y)$

Refines $\vdash \forall\ X\ \$\widehat{=}_1\ \$\widehat{=}_2$
$\bullet\ (\$\widehat{=}_1\ \mathit{Refines}\ \$\widehat{=}_2)\ X$
$\Leftrightarrow (\forall\ x\ y\bullet\ x \in X \land y \in X \land x\ \widehat{=}_1\ y \Rightarrow x\ \widehat{=}_2\ y)$

Contents $\vdash\ \mathit{ConstSpec}$
$(\lambda\ \mathit{Contents}'\bullet \forall\ x\bullet\ \mathit{Contents}'\ \{x\} = x)$
$\mathit{Contents}$

$^-$ $\vdash \forall\ f\ A\bullet\ (f\ ^-)\ A = \mathit{Contents}\ \{y | \exists\ x\bullet\ x \in A \land y = f\ x\}$

A.7 Theorems

Contents_consistent
$\vdash\ \mathit{Consistent}\ (\lambda\ \mathit{Contents}'\bullet \forall\ x\bullet\ \mathit{Contents}'\ \{x\} = x)$

contents_def $\vdash \forall\ x\bullet\ \mathit{Contents}\ \{x\} = x$

equiv_class_eq_thm
$\vdash \forall\ X\ \$\widehat{=}\ x\ y$
$\bullet\ \mathit{Equiv}\ (X, \$\widehat{=}) \land x \in X \land y \in X$
$\Rightarrow (\mathit{EquivClass}\ (X, \$\widehat{=})\ x = \mathit{EquivClass}\ (X, \$\widehat{=})\ y$
$\Leftrightarrow x\ \widehat{=}\ y)$

equiv_class_∈_thm
$\vdash \forall\ X\ \$\widehat{=}\ x\ y$
$\bullet\ \mathit{Equiv}\ (X, \$\widehat{=}) \land x \in X \Rightarrow x \in \mathit{EquivClass}\ (X, \$\widehat{=})\ x$

constant_img_thm
$\vdash \forall\ A\ a\ c$
$\bullet\ a \in A \land (\forall\ x\bullet\ x \in A \Rightarrow f\ x = c)$
$\Rightarrow \{y | \exists\ x\bullet\ x \in A \land y = f\ x\} = \{c\}$

respects_img_thm
$\vdash \forall\ X\ \$\widehat{=}\ f\ x$
$\bullet\ \mathit{Equiv}\ (X, \$\widehat{=}) \land (f\ \mathit{Respects}\ \$\widehat{=})\ X \land x \in X$
$\Rightarrow \{y | \exists\ z\bullet\ z \in \mathit{EquivClass}\ (X, \$\widehat{=})\ x \land y = f\ z\}$
$= \{f\ x\}$

respects_img_contents_thm
$\vdash \forall\ X\ \$\widehat{=}\ f\ x$
$\bullet\ \mathit{Equiv}\ (X, \$\widehat{=}) \land (f\ \mathit{Respects}\ \$\widehat{=})\ X \land x \in X$
$\Rightarrow \mathit{Contents}$
$\{y | \exists\ z\bullet\ z \in \mathit{EquivClass}\ (X, \$\widehat{=})\ x \land y = f\ z\}$

$= f\ x$

quotient_map_onto_thm

$\vdash \forall\ X\ \$\widehat{=}$
• $Equiv\ (X,\ \$\widehat{=})$
$\Rightarrow (\forall\ A$
• $A \in X\ /\ \$\widehat{=}$
$\Rightarrow (\exists\ x\bullet\ x \in X \wedge A = EquivClass\ (X,\ \$\widehat{=})\ x))$

quotient_∈_thm

$\vdash \forall\ A\ X\ \$\widehat{=}\ x\bullet\ A \in X\ /\ \$\widehat{=} \wedge x \in A \Rightarrow x \in X$

quotient_rep_∃_thm

$\vdash \forall\ X\ \$\widehat{=}$
• $Equiv\ (X,\ \$\widehat{=}) \wedge A \in X\ /\ \$\widehat{=} \Rightarrow (\exists\ x\bullet\ x \in X \wedge x \in A)$

respects_refines_thm

$\vdash \forall\ X\ f\ R\ S$
• $(f\ Respects\ S)\ X \wedge (R\ Refines\ S)\ X$
$\Rightarrow (f\ Respects\ R)\ X$

equiv_mono_thm

$\vdash \forall\ X\ \$\widehat{=}\ Y\bullet\ Equiv\ (X,\ \$\widehat{=}) \wedge Y \subseteq X \Rightarrow Equiv\ (Y,\ \$\widehat{=})$

induced_fun_equiv_class_thm

$\vdash \forall\ X\ \$\widehat{=}\ f\ x$
• $Equiv\ (X,\ \$\widehat{=}) \wedge (f\ Respects\ \$\widehat{=})\ X \wedge x \in X$
$\Rightarrow (f\ ^{-})\ (EquivClass\ (X,\ \$\widehat{=})\ x) = f\ x$

induced_fun_thm

$\vdash \forall\ X\ \$\widehat{=}\ f$
• $Equiv\ (X,\ \$\widehat{=}) \wedge (f\ Respects\ \$\widehat{=})\ X$
$\Rightarrow (\forall\ A\ x\bullet\ A \in X\ /\ \$\widehat{=} \wedge x \in A \Rightarrow (f\ ^{-})\ A = f\ x)$

induced_fun_induced_fun_thm

$\vdash \forall\ X\ R\ Y\ S\ f$
• $Equiv\ (X,\ R)$
$\wedge\ Equiv\ (Y,\ S)$
$\wedge\ (\forall\ y\bullet\ y \in Y \Rightarrow ((\lambda\ x\bullet\ f\ x\ y)\ Respects\ R)\ X)$
$\wedge\ (\forall\ x\bullet\ x \in X \Rightarrow ((\lambda\ y\bullet\ f\ x\ y)\ Respects\ S)\ Y)$
$\Rightarrow (\forall\ A\ x\ B\ y$
• $A \in X\ /\ R \wedge x \in A \wedge B \in Y\ /\ S \wedge y \in B$
$\Rightarrow ((\lambda\ x\bullet\ (f\ x\ ^{-})\ B)\ ^{-})\ A = f\ x\ y)$

induced_fun_∃_thm

$\vdash \forall\ X\ \$\widehat{=}\ f$
• $Equiv\ (X,\ \$\widehat{=}) \wedge (f\ Respects\ \$\widehat{=})\ X$
$\Rightarrow (\exists\ g\bullet\ \forall\ A\ x\bullet\ A \in X\ /\ \$\widehat{=} \wedge x \in A \Rightarrow g\ A = f\ x)$

induced_fun_∃_unique_thm

$\vdash \forall\ X\ \$\widehat{=}\ f\ g\ h$
• $Equiv\ (X,\ \$\widehat{=})$
$\wedge\ (f\ Respects\ \$\widehat{=})\ X$
$\wedge\ (\forall\ A\ x\bullet\ A \in X\ /\ \$\widehat{=} \wedge x \in A \Rightarrow g\ A = f\ x)$
$\wedge\ (\forall\ A\ x\bullet\ A \in X\ /\ \$\widehat{=} \wedge x \in A \Rightarrow h\ A = f\ x)$
$\Rightarrow (\forall\ A\bullet\ A \in X\ /\ \$\widehat{=} \Rightarrow h\ A = g\ A)$

dyadic_induced_fun_∃_thm

$\vdash \forall\ X\ R\ Y\ S\ f$
• $Equiv\ (X,\ R)$
$\wedge\ Equiv\ (Y,\ S)$
$\wedge\ (\forall\ y\bullet\ y \in Y \Rightarrow ((\lambda\ x\bullet\ f\ x\ y)\ Respects\ R)\ X)$

$\wedge\ (\forall\ x\bullet\ x \in X \Rightarrow ((\lambda\ y\bullet\ f\ x\ y)\ \mathit{Respects}\ S)\ Y)$
$\Rightarrow (\exists\ g$
$\bullet\ \forall\ A\ x\ B\ y$
$\bullet\ A \in X\ /\ R \wedge x \in A \wedge B \in Y\ /\ S \wedge y \in B$
$\Rightarrow\ g\ A\ B = f\ x\ y)$

dyadic_induced_fun_∃_unique_thm

$\vdash \forall\ X\ R\ Y\ S\ f\ g\ h$
$\bullet\ \mathit{Equiv}\ (X,\ R)$
$\wedge\ \mathit{Equiv}\ (Y,\ S)$
$\wedge\ (\forall\ y\bullet\ y \in Y \Rightarrow ((\lambda\ x\bullet\ f\ x\ y)\ \mathit{Respects}\ R)\ X)$
$\wedge\ (\forall\ x\bullet\ x \in X \Rightarrow ((\lambda\ y\bullet\ f\ x\ y)\ \mathit{Respects}\ S)\ Y)$
$\wedge\ (\forall\ A\ x\ B\ y$
$\bullet\ A \in X\ /\ R \wedge x \in A \wedge B \in Y\ /\ S \wedge y \in B$
$\Rightarrow\ g\ A\ B = f\ x\ y)$
$\wedge\ (\forall\ A\ x\ B\ y$
$\bullet\ A \in X\ /\ R \wedge x \in A \wedge B \in Y\ /\ S \wedge y \in B$
$\Rightarrow\ h\ A\ B = f\ x\ y)$
$\Rightarrow (\forall\ A\ B\ y$
$\bullet\ A \in X\ /\ R \wedge B \in Y\ /\ S \Rightarrow h\ A\ B = g\ A\ B)$

B THE THEORY groups

B.1 Parents

cache'maths_egs *fincomb* *equiv_rel*

B.2 Children

group_egs

B.3 Constants

$\boldsymbol{Inverse_G}$	$'a\ GROUP \to 'a \to 'a$
$\boldsymbol{Unit_G}$	$'a\ GROUP \to 'a$
$\boldsymbol{Times_G}$	$'a\ GROUP \to 'a \to 'a \to 'a$
$\boldsymbol{Car_G}$	$'a\ GROUP \to 'a\ \mathbb{P}$
$\boldsymbol{MkGROUP}$	$'a\ \mathbb{P} \to ('a \to 'a \to 'a) \to 'a \to ('a \to 'a) \to 'a\ GROUP$
$\boldsymbol{Unit}$	$'a\ GROUP \to 'a$
$\$\sim$	$'a \to 'a\ GROUP \to 'a$
$\$.$	$'a \to 'a \to 'a\ GROUP \to 'a$
$\boldsymbol{Car}$	$'a\ GROUP \to 'a\ \mathbb{P}$
$\boldsymbol{Group}$	$'a\ GROUP\ \mathbb{P}$
$\boldsymbol{Homomorphism}$	$'a\ GROUP \times 'b\ GROUP \to ('a \to 'b)\ \mathbb{P}$
$\boldsymbol{Subgroup}$	$'a\ GROUP \to 'a\ GROUP\ \mathbb{P}$
$\boldsymbol{UnitSubgroup}$	$'a\ GROUP \to 'a\ GROUP$
$\boldsymbol{Restriction}$	$'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ GROUP$
$\boldsymbol{NormalSubgroup}$	$'a\ GROUP \to 'a\ GROUP\ \mathbb{P}$
$\boldsymbol{Ker}$	$('a \to 'b) \to 'a\ GROUP \times 'b\ GROUP \to 'a\ GROUP$
$\boldsymbol{RightCongruent}$	$'a\ GROUP \to 'a\ GROUP \to 'a \to 'a \to BOOL$
$\boldsymbol{SetInverse}$	$'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ \mathbb{P}$
$\boldsymbol{SetTimesElem}$	$'a\ \mathbb{P} \to 'a \to 'a\ GROUP \to 'a\ \mathbb{P}$
$\boldsymbol{ElemTimesSet}$	$'a \to 'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ \mathbb{P}$
$\boldsymbol{SetTimesSet}$	$'a\ \mathbb{P} \to 'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ \mathbb{P}$
$\boldsymbol{GroupTimesGroup}$	$'a\ GROUP \to 'a\ GROUP \to 'a\ GROUP$
$\boldsymbol{QuotientGroup}$	$'a\ GROUP \to 'a\ GROUP \to 'a\ \mathbb{P}\ GROUP$
$\boldsymbol{Img}$	$('a \to 'b) \to 'a\ \mathbb{P} \to 'b\ GROUP \to 'b\ GROUP$
$\boldsymbol{Isomorphism}$	$'a\ GROUP \times 'b\ GROUP \to ('a \to 'b)\ \mathbb{P}$
$\boldsymbol{Inverse}$	$('a \to 'b) \to 'b \to 'a$
$\boldsymbol{SymGroup}$	$'a\ \mathbb{P} \to ('a \to 'a)\ GROUP$
$\boldsymbol{Size_G}$	$'a\ GROUP \to \mathbb{N}$
$\$\times_{\boldsymbol{G}}$	$'a\ GROUP \to 'b\ GROUP \to ('a \times 'b)\ GROUP$

B.4 Aliases

$\cap$ $\quad$ $Restriction : 'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ GROUP$
. $\quad$ $SetTimesSet : 'a\ \mathbb{P} \to 'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ \mathbb{P}$
. $\quad$ $ElemTimesSet : 'a \to 'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ \mathbb{P}$
. $\quad$ $SetTimesElem : 'a\ \mathbb{P} \to 'a \to 'a\ GROUP \to 'a\ \mathbb{P}$
$\sim$ $\quad$ $SetInverse : 'a\ \mathbb{P} \to 'a\ GROUP \to 'a\ \mathbb{P}$
. $\quad$ $GroupTimesGroup : 'a\ GROUP \to 'a\ GROUP \to 'a\ GROUP$
/ $\quad$ $QuotientGroup : 'a\ GROUP \to 'a\ GROUP \to 'a\ \mathbb{P}\ GROUP$
$\quad$ $Size_G : 'a\ GROUP \to \mathbb{N}$

B.5 Types

$'1\ \boldsymbol{GROUP}$

B.6 Fixity

Right Infix 290:
$\quad \times_{\boldsymbol{G}}$
Right Infix 310:
$\quad .$
Postfix 330: $\quad \sim$

B.7 Definitions

$\boldsymbol{GROUP}$ $\quad \vdash \exists\ f \bullet\ TypeDefn\ (\lambda\ x \bullet\ T)\ f$
$\boldsymbol{MkGROUP}$
$\boldsymbol{Car_G}$
$\boldsymbol{Times_G}$
$\boldsymbol{Unit_G}$
$\boldsymbol{Inverse_G}$ $\quad \vdash \forall\ t\ x1\ x2\ x3\ x4$
$\bullet\ Car_G\ (MkGROUP\ x1\ x2\ x3\ x4) = x1$
$\wedge\ Times_G\ (MkGROUP\ x1\ x2\ x3\ x4) = x2$
$\wedge\ Unit_G\ (MkGROUP\ x1\ x2\ x3\ x4) = x3$
$\wedge\ Inverse_G\ (MkGROUP\ x1\ x2\ x3\ x4) = x4$
$\wedge\ MkGROUP$
$(Car_G\ t)$
$(Times_G\ t)$
$(Unit_G\ t)$
$(Inverse_G\ t)$
$= t$
$\boldsymbol{Car}$
.
$\sim$
$\boldsymbol{Unit}$ $\quad \vdash ConstSpec$
$(\lambda\ (Car', \$".'", \$"\sim'", Unit')$
$\bullet\ \forall\ set\ times\ one\ inverse$
$\bullet\ Car'\ (MkGROUP\ set\ times\ one\ inverse) = set$
$\wedge\ (\forall\ x\ y$
$\bullet\ \$".'"\ x\ y\ (MkGROUP\ set\ times\ one\ inverse)$

$= times\ x\ y)$
$\wedge\ Unit'\ (MkGROUP\ set\ times\ one\ inverse)$
$= one$
$\wedge\ (\forall\ x$
$\bullet\ \$"\sim'"\ x\ (MkGROUP\ set\ times\ one\ inverse)$
$= inverse\ x))$
$(Car,\ \$.,\ \$\sim,\ Unit)$

Group $\vdash \forall\ G$
$\bullet\ G \in Group$
$\Leftrightarrow (\forall\ x\ y$
$\bullet\ x \in Car\ G \wedge y \in Car\ G \Rightarrow (x\ .\ y)\ G \in Car\ G)$
$\wedge\ (\forall\ x\ y\ z$
$\bullet\ x \in Car\ G \wedge y \in Car\ G \wedge z \in Car\ G$
$\Rightarrow ((x\ .\ y)\ G\ .\ z)\ G = (x\ .\ (y\ .\ z)\ G)\ G)$
$\wedge\ Unit\ G \in Car\ G$
$\wedge\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow (x\ .\ Unit\ G)\ G = x \wedge (Unit\ G\ .\ x)\ G = x)$
$\wedge\ (\forall\ x\bullet\ x \in Car\ G \Rightarrow (x\ \sim)\ G \in Car\ G)$
$\wedge\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow (x\ .\ (x\ \sim)\ G)\ G = Unit\ G$
$\wedge\ ((x\ \sim)\ G\ .\ x)\ G = Unit\ G)$

Homomorphism $\vdash \forall\ G\ H\ f$
$\bullet\ f \in Homomorphism\ (G,\ H)$
$\Leftrightarrow (\forall\ x\bullet\ x \in Car\ G \Rightarrow f\ x \in Car\ H)$
$\wedge\ (\forall\ x\ y$
$\bullet\ x \in Car\ G \wedge y \in Car\ G$
$\Rightarrow f\ ((x\ .\ y)\ G) = (f\ x\ .\ f\ y)\ H)$

Subgroup $\vdash \forall\ G\ H$
$\bullet\ H \in Subgroup\ G$
$\Leftrightarrow Car\ H \subseteq Car\ G$
$\wedge\ H \in Group$
$\wedge\ (\forall\ x\ y\bullet\ (x\ .\ y)\ H = (x\ .\ y)\ G)$
$\wedge\ (\forall\ x\bullet\ (x\ \sim)\ H = (x\ \sim)\ G)$

UnitSubgroup $\vdash \forall\ G$
$\bullet\ UnitSubgroup\ G$
$= MkGROUP$
$\{Unit\ G\}$
$(\lambda\ x\ y\bullet\ (x\ .\ y)\ G)$
$(Unit\ G)$
$(\lambda\ x\bullet\ (x\ \sim)\ G)$

Restriction $\vdash \forall\ A\ G$
$\bullet\ A \cap G$
$= MkGROUP$
$(A \cap Car\ G)$
$(\lambda\ x\ y\bullet\ (x\ .\ y)\ G)$
$(Unit\ G)$
$(\lambda\ x\bullet\ (x\ \sim)\ G)$

NormalSubgroup
$\vdash \forall\ G\ H$

$\bullet\ H \in NormalSubgroup\ G$
$\Leftrightarrow H \in Subgroup\ G$
$\wedge\ (\forall\ x\ y$
$\bullet\ x \in Car\ H \wedge y \in Car\ G$
$\Rightarrow ((y\ ^{\sim})\ G\ .\ (x\ .\ y)\ G)\ G \in Car\ H)$

Ker $\vdash \forall\ f\ G\ H$
$\bullet\ Ker\ f\ (G,\ H)$
$= MkGROUP$
$\{x|x \in Car\ G \wedge f\ x = Unit\ H\}$
$(\lambda\ x\ y\bullet\ (x\ .\ y)\ G)$
$(Unit\ G)$
$(\lambda\ x\bullet\ (x\ ^{\sim})\ G)$

RightCongruent
$\vdash \forall\ G\ H\ x\ y$
$\bullet\ RightCongruent\ H\ G\ x\ y \Leftrightarrow ((x\ ^{\sim})\ G\ .\ y)\ G \in Car\ H$

SetTimesSet

ElemTimesSet

SetTimesElem

SetInverse $\vdash (\forall\ G\ A\ B$
$\bullet\ (A\ .\ B)\ G$
$= \{z|\exists\ a\ b\bullet\ a \in A \wedge b \in B \wedge z = (a\ .\ b)\ G\})$
$\wedge\ (\forall\ G\ x\ B$
$\bullet\ (x\ .\ B)\ G = \{z|\exists\ b\bullet\ b \in B \wedge z = (x\ .\ b)\ G\})$
$\wedge\ (\forall\ G\ A\ y$
$\bullet\ (A\ .\ y)\ G = \{z|\exists\ a\bullet\ a \in A \wedge z = (a\ .\ y)\ G\})$
$\wedge\ (\forall\ G\ A\bullet\ (A\ ^{\sim})\ G = \{z|\exists\ a\bullet\ a \in A \wedge z = (a\ ^{\sim})\ G\})$

GroupTimesGroup
$\vdash \forall\ H\ K$
$\bullet\ H\ .\ K$
$= MkGROUP$
$((Car\ H\ .\ Car\ K)\ H)$
$(\lambda\ x\ y\bullet\ (x\ .\ y)\ H)$
$(Unit\ H)$
$(\lambda\ x\bullet\ (x\ ^{\sim})\ H)$

QuotientGroup
$\vdash \forall\ G\ H$
$\bullet\ G\ /\ H$
$= MkGROUP$
$\{A|\exists\ x\bullet\ x \in Car\ G \wedge A = (x\ .\ Car\ H)\ G\}$
$(\lambda\ A\ B\bullet\ (A\ .\ B)\ G)$
$((Unit\ G\ .\ Car\ H)\ G)$
$(\lambda\ A\bullet\ (A\ ^{\sim})\ G)$

Img $\vdash \forall\ f\ X\ G$
$\bullet\ Img\ f\ X\ G$
$= MkGROUP$
$\{y|\exists\ x\bullet\ x \in X \wedge y = f\ x\}$
$(\lambda\ x\ y\bullet\ (x\ .\ y)\ G)$
$(Unit\ G)$
$(\lambda\ x\bullet\ (x\ ^{\sim})\ G)$

Isomorphism $\vdash \forall\ G\ H\ f$
$\bullet\ f \in Isomorphism\ (G,\ H)$

$\Leftrightarrow f \in Homomorphism\ (G,\ H)$
$\wedge\ (\forall\ x\ y$
$\bullet\ x \in Car\ G \wedge y \in Car\ G \wedge f\ x = f\ y \Rightarrow x = y)$
$\wedge\ (\forall\ z \bullet\ z \in Car\ H \Rightarrow (\exists\ x \bullet\ x \in Car\ G \wedge f\ x = z))$

Inverse $\vdash$ *ConstSpec*
$(\lambda\ Inverse'$
$\bullet\ \forall\ f$
$\bullet\ OneOne\ f \wedge Onto\ f$
$\Rightarrow (\forall\ x \bullet\ Inverse'\ f\ (f\ x) = x)$
$\wedge\ (\forall\ y \bullet\ f\ (Inverse'\ f\ y) = y))$
Inverse

SymGroup $\vdash \forall\ X$
$\bullet\ SymGroup\ X$
$= MkGROUP$
$\{f$
$|OneOne\ f$
$\wedge\ Onto\ f$
$\wedge\ (\forall\ y \bullet\ \neg\ y \in X \Rightarrow f\ y = y)\}$
$(\lambda\ f\ g\ x \bullet\ f\ (g\ x))$
$(\lambda\ x \bullet\ x)$
Inverse

Size$_G$ $\vdash \forall\ G \bullet\ \#\ G = \#\ (Car\ G)$

$\times_G$ $\vdash \forall\ G\ H$
$\bullet\ G \times_G H$
$= MkGROUP$
$(Car\ G \times Car\ H)$
$(\lambda\ (a,\ b)\ (c,\ d) \bullet\ ((a\ .\ c)\ G,\ (b\ .\ d)\ H))$
$(Unit\ G,\ Unit\ H)$
$(\lambda\ (a,\ b) \bullet\ ((a\ {}^{\sim})\ G,\ (b\ {}^{\sim})\ H))$

B.8 Theorems

Car_consistent
._consistent
$^{\sim}$_consistent
Unit_consistent
$\vdash$ *Consistent*
$(\lambda\ (Car',\ \$".'",\ \$"\sim'",\ Unit')$
$\bullet\ \forall\ set\ times\ one\ inverse$
$\bullet\ Car'\ (MkGROUP\ set\ times\ one\ inverse) = set$
$\wedge\ (\forall\ x\ y$
$\bullet\ \$".'"\ x\ y\ (MkGROUP\ set\ times\ one\ inverse)$
$= times\ x\ y)$
$\wedge\ Unit'\ (MkGROUP\ set\ times\ one\ inverse)$
$= one$
$\wedge\ (\forall\ x$
$\bullet\ \$"\sim'"\ x\ (MkGROUP\ set\ times\ one\ inverse)$
$= inverse\ x))$

group_ops_def
$\vdash \forall\ set\ times\ one\ inverse$
$\bullet\ Car\ (MkGROUP\ set\ times\ one\ inverse) = set$

$$
\begin{array}{l}
\quad \wedge\ (\forall\ x\ y \\
\quad \bullet\ (x\ .\ y)\ (MkGROUP\ set\ times\ one\ inverse) \\
\qquad = times\ x\ y) \\
\quad \wedge\ Unit\ (MkGROUP\ set\ times\ one\ inverse) = one \\
\quad \wedge\ (\forall\ x \\
\quad \bullet\ (x\ {}^{\sim})\ (MkGROUP\ set\ times\ one\ inverse) \\
\qquad = inverse\ x)
\end{array}
$$

Inverse_consistent

$$
\begin{array}{l}
\vdash\ Consistent \\
\quad (\lambda\ Inverse' \\
\quad \bullet\ \forall\ f \\
\quad \bullet\ OneOne\ f \wedge Onto\ f \\
\qquad \Rightarrow (\forall\ x\bullet\ Inverse'\ f\ (f\ x) = x) \\
\qquad\ \wedge (\forall\ y\bullet\ f\ (Inverse'\ f\ y) = y))
\end{array}
$$

inverse_def

$$
\begin{array}{l}
\vdash\ \forall\ f \\
\bullet\ OneOne\ f \wedge Onto\ f \\
\quad \Rightarrow (\forall\ x\bullet\ Inverse\ f\ (f\ x) = x) \\
\quad\ \wedge (\forall\ y\bullet\ f\ (Inverse\ f\ y) = y)
\end{array}
$$

group_eq_group_thm

$$
\begin{array}{l}
\vdash\ \forall\ G\ H \\
\bullet\ G = H \\
\quad \Leftrightarrow Car\ G = Car\ H \\
\quad\ \wedge (\forall\ x\ y\bullet\ (x\ .\ y)\ G = (x\ .\ y)\ H) \\
\quad\ \wedge\ Unit\ G = Unit\ H \\
\quad\ \wedge (\forall\ x\bullet\ (x\ {}^{\sim})\ G = (x\ {}^{\sim})\ H)
\end{array}
$$

group_clauses1

$$
\begin{array}{l}
\vdash\ \forall\ G\ x \\
\bullet\ G \in Group \wedge x \in Car\ G \\
\quad \Rightarrow (x\ {}^{\sim})\ G \in Car\ G \\
\quad\ \wedge (x\ .\ Unit\ G)\ G = x \\
\quad\ \wedge (Unit\ G\ .\ x)\ G = x \\
\quad\ \wedge ((x\ {}^{\sim})\ G\ .\ x)\ G = Unit\ G \\
\quad\ \wedge (x\ .\ (x\ {}^{\sim})\ G)\ G = Unit\ G
\end{array}
$$

group_clauses2

$$
\begin{array}{l}
\vdash\ \forall\ G\ x\ y\ z \\
\bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G \\
\quad \Rightarrow (x\ .\ y)\ G \in Car\ G \\
\quad\ \wedge (\forall\ z \\
\quad\ \bullet\ z \in Car\ G \\
\qquad \Rightarrow ((x\ .\ y)\ G\ .\ z)\ G = (x\ .\ (y\ .\ z)\ G)\ G)
\end{array}
$$

group_eq_thm

$$
\begin{array}{l}
\vdash\ \forall\ G\ x\ y \\
\bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G \\
\quad \Rightarrow (x = y \Leftrightarrow (x\ .\ (y\ {}^{\sim})\ G)\ G = Unit\ G)
\end{array}
$$

group_eq_thm1

$$
\begin{array}{l}
\vdash\ \forall\ G\ x\ y \\
\bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G \\
\quad \Rightarrow (x = y \Leftrightarrow ((x\ {}^{\sim})\ G\ .\ y)\ G = Unit\ G)
\end{array}
$$

left_cancel_thm

$$
\begin{array}{l}
\vdash\ \forall\ G\ x\ y\ z \\
\bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G \wedge z \in Car\ G \\
\quad \Rightarrow ((x\ .\ y)\ G = (x\ .\ z)\ G \Leftrightarrow y = z)
\end{array}
$$

inverse_inverse_thm

$$\vdash \forall\ G\ x\bullet\ G \in Group \wedge x \in Car\ G \Rightarrow ((x\ ^{\sim})\ G\ ^{\sim})\ G = x$$

group_clauses3

$$\begin{array}{l}\vdash \forall\ G\ x\ y\\ \quad \bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G\\ \qquad \Rightarrow (x\ .\ ((x\ ^{\sim})\ G\ .\ y)\ G)\ G = y\\ \qquad\quad \wedge ((x\ ^{\sim})\ G\ .\ (x\ .\ y)\ G)\ G = y\end{array}$$

times_inverse_thm

$$\begin{array}{l}\vdash \forall\ G\ x\ y\\ \quad \bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G\\ \qquad \Rightarrow ((x\ .\ y)\ G\ ^{\sim})\ G = ((y\ ^{\sim})\ G\ .\ (x\ ^{\sim})\ G)\ G\end{array}$$

inverse_unique_thm

$$\begin{array}{l}\vdash \forall\ G\ x\ y\\ \quad \bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G\\ \qquad \Rightarrow ((x\ .\ y)\ G = Unit\ G \Leftrightarrow y = (x\ ^{\sim})\ G)\end{array}$$

group_clauses4

$$\begin{array}{l}\vdash \forall\ G\\ \quad \bullet\ G \in Group \Rightarrow Unit\ G \in Car\ G \wedge (Unit\ G\ ^{\sim})\ G = Unit\ G\end{array}$$

unit_unique_thm

$$\begin{array}{l}\vdash \forall\ G\ x\ y\\ \quad \bullet\ G \in Group \wedge x \in Car\ G \wedge y \in Car\ G\\ \qquad \Rightarrow ((x\ .\ y)\ G = x \Leftrightarrow y = Unit\ G)\end{array}$$

homomorphism_unit_thm

$$\begin{array}{l}\vdash \forall\ G\ H\ f\\ \quad \bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)\\ \qquad \Rightarrow f\ (Unit\ G) = Unit\ H\end{array}$$

homomorphism_inverse_thm

$$\begin{array}{l}\vdash \forall\ G\ H\ f\ x\\ \quad \bullet\ G \in Group\\ \qquad \wedge\ H \in Group\\ \qquad \wedge\ f \in Homomorphism\ (G,\ H)\\ \qquad \wedge\ x \in Car\ G\\ \qquad \Rightarrow f\ ((x\ ^{\sim})\ G) = (f\ x\ ^{\sim})\ H\end{array}$$

homomorphism_clauses

$$\begin{array}{l}\vdash \forall\ G\ H\ f\\ \quad \bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)\\ \qquad \Rightarrow f\ (Unit\ G) = Unit\ H\\ \qquad\quad \wedge\ (\forall\ x\bullet\ x \in Car\ G \Rightarrow f\ ((x\ ^{\sim})\ G) = (f\ x\ ^{\sim})\ H)\\ \qquad\quad \wedge\ (\forall\ x\ y\\ \qquad\quad \bullet\ x \in Car\ G \wedge y \in Car\ G\\ \qquad\qquad \Rightarrow f\ ((x\ .\ y)\ G) = (f\ x\ .\ f\ y)\ H)\end{array}$$

homomorphism_∈_car_thm

$$\begin{array}{l}\vdash \forall\ G\ H\ f\ x\\ \quad \bullet\ G \in Group\\ \qquad \wedge\ H \in Group\\ \qquad \wedge\ f \in Homomorphism\ (G,\ H)\\ \qquad \wedge\ x \in Car\ G\\ \qquad \Rightarrow f\ x \in Car\ H\end{array}$$

subgroup_clauses

$$\begin{array}{l}\vdash \forall\ G\ H\ x\ y\\ \quad \bullet\ G \in Group \wedge H \in Subgroup\ G\end{array}$$

$$\Rightarrow (x\ .\ y)\ H = (x\ .\ y)\ G$$
$$\wedge\ Unit\ H = Unit\ G$$
$$\wedge\ (x\ ^{\sim})\ H = (x\ ^{\sim})\ G$$

trivial_subgroups_thm

$$\vdash \forall\ G$$
$$\bullet\ G \in Group$$
$$\Rightarrow G \in Subgroup\ G \wedge UnitSubgroup\ G \in Subgroup\ G$$

subgroup_trans_thm

$$\vdash \forall\ G\ H\ K$$
$$\bullet\ G \in Group \wedge K \in Subgroup\ H \wedge H \in Subgroup\ G$$
$$\Rightarrow K \in Subgroup\ G$$

subgroup_⊆_subgroup_thm

$$\vdash \forall\ G\ H\ K$$
$$\bullet\ G \in Group$$
$$\wedge\ H \in Subgroup\ G$$
$$\wedge\ K \in Subgroup\ G$$
$$\wedge\ Car\ K \subseteq Car\ H$$
$$\Rightarrow K \in Subgroup\ H$$

subgroup_eq_thm

$$\vdash \forall\ G\ H\ K$$
$$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in Subgroup\ G$$
$$\Rightarrow (K = H \Leftrightarrow Car\ K = Car\ H)$$

subgroup_homomorphism_thm

$$\vdash \forall\ G\ H$$
$$\bullet\ G \in Group \wedge H \in Subgroup\ G$$
$$\Rightarrow (\lambda\ x\bullet\ x) \in Homomorphism\ (H,\ G)$$

id_homomorphism_thm

$$\vdash \forall\ G\bullet\ G \in Group \Rightarrow (\lambda\ x\bullet\ x) \in Homomorphism\ (G,\ G)$$

unit_homomorphism_thm

$$\vdash \forall\ G\ H$$
$$\bullet\ G \in Group \wedge H \in Group$$
$$\Rightarrow (\lambda\ x\bullet\ Unit\ H) \in Homomorphism\ (G,\ H)$$

comp_homomorphism_thm

$$\vdash \forall\ G\ H\ K\ f\ g$$
$$\bullet\ G \in Group$$
$$\wedge\ H \in Group$$
$$\wedge\ K \in Group$$
$$\wedge\ f \in Homomorphism\ (G,\ H)$$
$$\wedge\ g \in Homomorphism\ (H,\ K)$$
$$\Rightarrow (\lambda\ x\bullet\ g\ (f\ x)) \in Homomorphism\ (G,\ K)$$

restriction_subgroup_thm

$$\vdash \forall\ A\ G$$
$$\bullet\ G \in Group$$
$$\Rightarrow (A \cap G \in Subgroup\ G$$
$$\Leftrightarrow (\forall\ x\ y$$
$$\bullet\ x \in A \wedge x \in Car\ G \wedge y \in A \wedge y \in Car\ G$$
$$\Rightarrow (x\ .\ y)\ G \in A)$$
$$\wedge\ (\forall\ x\bullet\ x \in A \wedge x \in Car\ G \Rightarrow (x\ ^{\sim})\ G \in A)$$
$$\wedge\ Unit\ G \in A)$$

ker_normal_subgroup_thm

$$\vdash \forall\ G\ H\ f$$

$\bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)$
$\Rightarrow Ker\ f\ (G,\ H) \in NormalSubgroup\ G$

subgroup_normal_subgroup_thm

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in Subgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow K \in NormalSubgroup\ H$

right_congruent_equiv_thm

$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in Subgroup\ G$
$\Rightarrow Equiv\ (Car\ G,\ RightCongruent\ H\ G)$

group_product_restriction_thm

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in Subgroup\ G$
$\Rightarrow H\ .\ K = (Car\ H\ .\ Car\ K)\ G \cap G$

group_product_subgroup_thm

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in NormalSubgroup\ G$
$\Rightarrow H\ .\ K \in Subgroup\ G$

right_coset_equiv_class_thm

$\vdash \forall\ G\ H\ x$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge x \in Car\ G$
$\Rightarrow (x\ .\ Car\ H)\ G$
$= EquivClass\ (Car\ G,\ RightCongruent\ H\ G)\ x$

right_coset_eq_thm

$\vdash \forall\ G\ H\ x$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge x \in Car\ G \wedge y \in Car\ G$
$\Rightarrow ((x\ .\ Car\ H)\ G = (y\ .\ Car\ H)\ G$
$\Leftrightarrow RightCongruent\ H\ G\ x\ y)$

quotient_group_times_thm

$\vdash \forall\ G\ H\ x\ y$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ x \in Car\ G$
$\wedge\ y \in Car\ G$
$\Rightarrow ((x\ .\ Car\ H)\ G\ .\ (y\ .\ Car\ H)\ G)\ G$
$= ((x\ .\ y)\ G\ .\ Car\ H)\ G$

quotient_group_inverse_thm

$\vdash \forall\ G\ H\ x\ y$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ x \in Car\ G$
$\wedge\ y \in Car\ G$
$\Rightarrow ((x\ .\ Car\ H)\ G\ {}^{\sim})\ G = ((x\ {}^{\sim})\ G\ .\ Car\ H)\ G$

quotient_group_rep_∃_thm

$\vdash \forall\ G\ H\ C$
$\bullet\ G \in Group \wedge C \in Car\ (G\ /\ H)$
$\Rightarrow (\exists\ x\bullet\ x \in Car\ G \wedge C = (x\ .\ Car\ H)\ G)$

quotient_group_group_thm

$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in NormalSubgroup\ G \Rightarrow G\ /\ H \in Group$

quotient_group_homomorphism_thm

$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in NormalSubgroup\ G$
$\Rightarrow (\lambda\ x\bullet\ (x\ .\ Car\ H)\ G) \in Homomorphism\ (G,\ G\ /\ H)$

ker_right_coset_thm

$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in NormalSubgroup\ G$
$\Rightarrow Ker\ (\lambda\ x\bullet\ (x\ .\ Car\ H)\ G)\ (G,\ G\ /\ H) = H$

img_subgroup_thm

$\vdash \forall\ G\ H\ f$
$\bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)$
$\Rightarrow Img\ f\ (Car\ G)\ H \in Subgroup\ H$

equiv_right_congruent_ker_thm

$\vdash \forall\ G\ H\ f$
$\bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)$
$\Rightarrow Equiv\ (Car\ G,\ RightCongruent\ (Ker\ f\ (G,\ H))\ G)$

homomorphism_respects_ker_thm

$\vdash \forall\ G\ H\ f$
$\bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)$
$\Rightarrow (f\ Respects\ RightCongruent\ (Ker\ f\ (G,\ H))\ G)$
$(Car\ G)$

car_quotient_group_thm

$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in NormalSubgroup\ G$
$\Rightarrow Car\ (G\ /\ H) = Car\ G\ /\ RightCongruent\ H\ G$

subgroup_refines_thm

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in Subgroup\ H$
$\Rightarrow (RightCongruent\ K\ G\ Refines\ RightCongruent\ H\ G)$
$(Car\ G)$

subgroup_ker_induced_thm

$\vdash \forall\ G\ K\ H\ f$
$\bullet\ G \in Group$
$\wedge\ H \in Group$
$\wedge\ f \in Homomorphism\ (G,\ H)$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ (Ker\ f\ (G,\ H))$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x\bullet\ x \in Car\ G \Rightarrow g\ ((x\ .\ Car\ K)\ G) = f\ x)$
$\wedge\ g \in Homomorphism\ (G\ /\ K,\ Img\ f\ (Car\ G)\ H))$

isomorphism_ker_img_thm

$\vdash \forall\ G\ H\ f$
$\bullet\ G \in Group \wedge H \in Group$
$\Rightarrow (f \in Isomorphism\ (G,\ H)$
$\Leftrightarrow f \in Homomorphism\ (G,\ H)$
$\wedge\ Ker\ f\ (G,\ H) = UnitSubgroup\ G$
$\wedge\ Img\ f\ (Car\ G)\ H = H)$

first_isomorphism_thm

$\vdash \forall\ G\ H\ f$

$\bullet\ G \in Group \wedge H \in Group \wedge f \in Homomorphism\ (G,\ H)$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow g\ ((x\ .\ Car\ (Ker\ f\ (G,\ H)))\ G) = f\ x)$
$\wedge\ g$
$\in Isomorphism$
$(G\ /\ Ker\ f\ (G,\ H),\ Img\ f\ (Car\ G)\ H))$

second_isomorphism_lemma1

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in NormalSubgroup\ G$
$\Rightarrow K \in NormalSubgroup\ (H\ .\ K)$

second_isomorphism_lemma2

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in NormalSubgroup\ G$
$\Rightarrow H \in Subgroup\ (H\ .\ K)$

second_isomorphism_lemma3

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in NormalSubgroup\ G$
$\Rightarrow Ker\ (\lambda\ x\bullet\ (x\ .\ Car\ K)\ G)\ (H,\ (H\ .\ K)\ /\ K)$
$= Car\ K \cap H$

second_isomorphism_lemma4

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in NormalSubgroup\ G$
$\Rightarrow (\lambda\ x\bullet\ (x\ .\ Car\ K)\ G)$
$\in Homomorphism\ (H,\ (H\ .\ K)\ /\ K)$

second_isomorphism_thm

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group \wedge H \in Subgroup\ G \wedge K \in NormalSubgroup\ G$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x$
$\bullet\ x \in Car\ H$
$\Rightarrow g\ ((x\ .\ (Car\ K \cap Car\ H))\ G)$
$= (x\ .\ Car\ K)\ G)$
$\wedge\ g$
$\in Isomorphism$
$(H\ /\ (Car\ K \cap H),\ (H\ .\ K)\ /\ K))$

third_isomorphism_lemma1

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow g\ ((x\ .\ Car\ K)\ G) = (x\ .\ Car\ H)\ G)$
$\wedge\ g \in Homomorphism\ (G\ /\ K,\ G\ /\ H))$

third_isomorphism_lemma2

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$

$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow g\ ((x\ .\ Car\ K)\ G) = (x\ .\ Car\ H)\ G)$
$\wedge\ g \in Homomorphism\ (G\ /\ K,\ G\ /\ H)$
$\wedge\ Car\ (Ker\ g\ (G\ /\ K,\ G\ /\ H)) = Car\ (H\ /\ K))$

third_isomorphism_lemma3

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow K \in NormalSubgroup\ H$

third_isomorphism_lemma4

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow H\ /\ K \in Subgroup\ (G\ /\ K)$

third_isomorphism_lemma5

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow g\ ((x\ .\ Car\ K)\ G) = (x\ .\ Car\ H)\ G)$
$\wedge\ g \in Homomorphism\ (G\ /\ K,\ G\ /\ H)$
$\wedge\ Ker\ g\ (G\ /\ K,\ G\ /\ H) = H\ /\ K$
$\wedge\ Img\ g\ (Car\ (G\ /\ K))\ (G\ /\ H) = G\ /\ H)$

third_isomorphism_thm

$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in NormalSubgroup\ G$
$\wedge\ K \in NormalSubgroup\ G$
$\wedge\ K \in Subgroup\ H$
$\Rightarrow (\exists\ g$
$\bullet\ (\forall\ x$
$\bullet\ x \in Car\ G$
$\Rightarrow g$
$(((x\ .\ Car\ K)\ G\ .\ Car\ (H\ /\ K))$
$(G\ /\ K))$
$= (x\ .\ Car\ H)\ G)$
$\wedge\ g \in Isomorphism\ (G\ /\ K\ /\ (H\ /\ K),\ G\ /\ H))$

sym_group_group_thm

$\vdash \forall\ X \bullet\ SymGroup\ X \in Group$

cayley_thm $\vdash \forall\ G$
$\bullet\ G \in Group$
$\Rightarrow (\exists\ f$
$\bullet (\forall\ x\ y$
$\bullet\ f\ x\ y$
$= (if\ y \in Car\ G\ then\ (x\ .\ y)\ G\ else\ y))$
$\wedge\ f$
$\in Isomorphism$
$(G,\ Img\ f\ (Car\ G)\ (SymGroup\ (Car\ G))))$

finite_subgroup_thm
$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge Car\ G \in Finite \wedge H \in Subgroup\ G$
$\Rightarrow Car\ H \in Finite$

finite_cosets_thm
$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge Car\ G \in Finite \wedge H \in Subgroup\ G$
$\Rightarrow Car\ (G\ /\ H) \in Finite$

lagrange_cosets_thm
$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge Car\ G \in Finite \wedge H \in Subgroup\ G$
$\Rightarrow Car\ H \in Finite$
$\wedge\ Car\ (G\ /\ H) \in Finite$
$\wedge\ \#\ G = \#\ H * \#\ (G\ /\ H)$

product_group_thm
$\vdash \forall\ G\ H \bullet\ G \in Group \wedge H \in Group \Rightarrow G \times_G H \in Group$

fst_honomorphism_thm
$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in Group$
$\Rightarrow Fst \in Homomorphism\ (G \times_G H,\ G)$

snd_honomorphism_thm
$\vdash \forall\ G\ H$
$\bullet\ G \in Group \wedge H \in Group$
$\Rightarrow Snd \in Homomorphism\ (G \times_G H,\ H)$

product_honomorphism_thm
$\vdash \forall\ G\ H\ K$
$\bullet\ G \in Group$
$\wedge\ H \in Group$
$\wedge\ K \in Group$
$\wedge\ f \in Homomorphism\ (G,\ H)$
$\wedge\ g \in Homomorphism\ (G,\ K)$
$\Rightarrow (\lambda\ x \bullet\ (f\ x,\ g\ x)) \in Homomorphism\ (G,\ H \times_G K)$

C THE THEORY group_egs

C.1 Parents

analysis *groups*

C.2 Constants

$\mathbb{Z}\_\boldsymbol{units}$ $\mathbb{Z}$ *GROUP*
$\mathbb{Z}\_\boldsymbol{plus}$ $\mathbb{Z}$ *GROUP*
$\mathbb{R}\_\boldsymbol{pos}$ $\mathbb{R}$ *GROUP*
$\mathbb{R}_+$ $\mathbb{R}$ *GROUP*

C.3 Definitions

$\mathbb{Z}\_\boldsymbol{plus}$
$\mathbb{Z}\_\boldsymbol{units}$ $\vdash \mathbb{Z}\_plus = MkGROUP\ Universe\ \$+\ (\mathbb{NZ}\ 0)\ \sim$
$\wedge\ \mathbb{Z}\_units$
$= MkGROUP\ \{\sim\ (\mathbb{NZ}\ 1);\ \mathbb{NZ}\ 1\}\ \$*\ (\mathbb{NZ}\ 1)\ (\lambda\ x\bullet\ x)$
$\mathbb{R}_+$
$\mathbb{R}\_\boldsymbol{pos}$ $\vdash \mathbb{R}_+ = MkGROUP\ Universe\ \$+\ 0.\ \sim$
$\wedge\ \mathbb{R}\_pos = MkGROUP\ \{x|0. < x\}\ \$*\ 1.\ \$^{-1}$

C.4 Theorems

$\mathbb{Z}\_\boldsymbol{plus\_group\_thm}$
$\vdash \mathbb{Z}\_plus \in Group$
$\mathbb{Z}\_\boldsymbol{units\_group\_thm}$
$\vdash \mathbb{Z}\_units \in Group$
$\mathbb{Z}\_\boldsymbol{plus\_ops\_thm}$
$\vdash Car\ \mathbb{Z}\_plus = Universe$
$\wedge\ (\forall\ i\ j\bullet\ (i\ .\ j)\ \mathbb{Z}\_plus = i + j)$
$\wedge\ Unit\ \mathbb{Z}\_plus = \mathbb{NZ}\ 0$
$\wedge\ (\forall\ i\bullet\ (i\ ^\sim)\ \mathbb{Z}\_plus = \sim i)$
$\mathbb{Z}\_\boldsymbol{units\_ops\_thm}$
$\vdash Car\ \mathbb{Z}\_units = \{\sim\ (\mathbb{NZ}\ 1);\ \mathbb{NZ}\ 1\}$
$\wedge\ (\forall\ i\ j\bullet\ (i\ .\ j)\ \mathbb{Z}\_units = i * j)$
$\wedge\ Unit\ \mathbb{Z}\_units = \mathbb{NZ}\ 1$
$\wedge\ (\forall\ i\bullet\ (i\ ^\sim)\ \mathbb{Z}\_units = i)$
$\mathbb{Z}\_\boldsymbol{plus}\_\mathbb{Z}\_\boldsymbol{units\_homomorphism\_def}$
$\vdash \forall\ f$
$\bullet\ f \in Homomorphism\ (\mathbb{Z}\_plus,\ \mathbb{Z}\_units)$
$\Leftrightarrow\ (\forall\ x\bullet\ f\ x \in \{\sim\ (\mathbb{NZ}\ 1);\ \mathbb{NZ}\ 1\})$
$\wedge\ (\forall\ x\ y\bullet\ f\ (x + y) = f\ x * f\ y)$
$\mathbb{Z}\_\boldsymbol{plus}\_\mathbb{Z}\_\boldsymbol{units\_homomorphism\_unit\_thm}$
$\vdash \forall\ f$
$\bullet\ f \in Homomorphism\ (\mathbb{Z}\_plus,\ \mathbb{Z}\_units)$
$\Rightarrow\ f\ (\mathbb{NZ}\ 0) = \mathbb{NZ}\ 1$
$\mathbb{Z}\_\boldsymbol{plus}\_\mathbb{Z}\_\boldsymbol{units\_homomorphism\_inverse\_thm}$
$\vdash \forall\ f\ x$

$\bullet\ f \in \mathit{Homomorphism}\ (\mathbb{Z}\_\mathit{plus}, \mathbb{Z}\_\mathit{units}) \Rightarrow f\ (\sim x) = f\ x$

$\mathbb{R}$_**additive_group_thm**

$\vdash \mathbb{R}_+ \in \mathit{Group}$

$\mathbb{R}$_**pos_group_thm**

$\vdash \mathbb{R}\_\mathit{pos} \in \mathit{Group}$

$\mathbb{R}$_**additive_ops_thm**

$\vdash \mathit{Car}\ \mathbb{R}_+ = \mathit{Universe}$
$\quad \wedge\ (\forall\ x\ y\bullet\ (x\ .\ y)\ \mathbb{R}_+ = x + y)$
$\quad \wedge\ \mathit{Unit}\ \mathbb{R}_+ = 0.$
$\quad \wedge\ (\forall\ x\bullet\ (x\ ^\sim)\ \mathbb{R}_+ = \sim x)$

$\mathbb{R}$_**pos_ops_thm**

$\vdash \mathit{Car}\ \mathbb{R}\_\mathit{pos} = \{x | 0. < x\}$
$\quad \wedge\ (\forall\ x\ y\bullet\ (x\ .\ y)\ \mathbb{R}\_\mathit{pos} = x * y)$
$\quad \wedge\ \mathit{Unit}\ \mathbb{R}\_\mathit{pos} = 1.$
$\quad \wedge\ (\forall\ x\bullet\ (x\ ^\sim)\ \mathbb{R}\_\mathit{pos} = x^{-1})$

$\mathbb{R}$_**additive_**$\mathbb{R}$_**pos_homomorphism_def**

$\vdash \forall\ f$
$\quad \bullet\ f \in \mathit{Homomorphism}\ (\mathbb{R}_+, \mathbb{R}\_\mathit{pos})$
$\qquad \Leftrightarrow\ (\forall\ x\bullet\ 0. < f\ x)$
$\qquad\quad \wedge\ (\forall\ x\ y\bullet\ f\ (x + y) = f\ x * f\ y)$

$\mathbb{R}$_**additive_**$\mathbb{R}$_**pos_homomorphism_unit_thm**

$\vdash \forall\ f\bullet\ f \in \mathit{Homomorphism}\ (\mathbb{R}_+, \mathbb{R}\_\mathit{pos}) \Rightarrow f\ 0. = 1.$

$\mathbb{R}$_**additive_**$\mathbb{R}$_**pos_homomorphism_inverse_thm**

$\vdash \forall\ f\ x$
$\quad \bullet\ f \in \mathit{Homomorphism}\ (\mathbb{R}_+, \mathbb{R}\_\mathit{pos}) \Rightarrow f\ (\sim x) = f\ x^{-1}$

$\mathbb{R}$_**additive_**$\mathbb{R}$_**pos_isomorphism_def**

$\vdash \forall\ f$
$\quad \bullet\ f \in \mathit{Isomorphism}\ (\mathbb{R}_+, \mathbb{R}\_\mathit{pos})$
$\qquad \Leftrightarrow f \in \mathit{Homomorphism}\ (\mathbb{R}_+, \mathbb{R}\_\mathit{pos})$
$\qquad\quad \wedge\ (\forall\ x\ y\bullet\ f\ x = f\ y \Rightarrow x = y)$
$\qquad\quad \wedge\ (\forall\ z\bullet\ 0. < z \Rightarrow (\exists\ x\bullet\ f\ x = z))$

exp_isomorphism_thm

$\vdash \mathit{Exp} \in \mathit{Isomorphism}\ (\mathbb{R}_+, \mathbb{R}\_\mathit{pos})$

linear_homomorphism_thm

$\vdash \forall\ c\bullet\ (\lambda\ x\bullet\ c * x) \in \mathit{Homomorphism}\ (\mathbb{R}_+, \mathbb{R}_+)$

plus_$\mathbb{R}$**_additive_homomorphism_thm**

$\vdash \mathit{Uncurry}\ \$+ \in \mathit{Homomorphism}\ (\mathbb{R}_+ \times_G \mathbb{R}_+, \mathbb{R}_+)$

INDEX