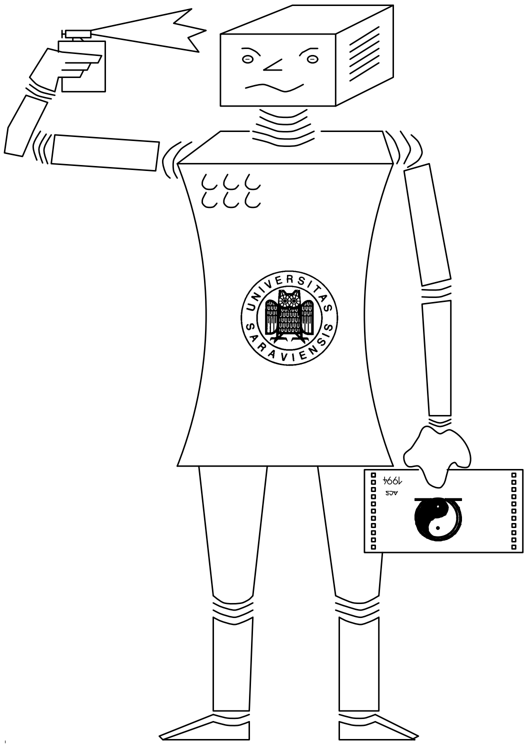
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://js-sfbsun.cs.uni-sb.de/pub/www/



SEKI Working Paper

A Mathematical Knowledge Base for Proving Theorems in Semigroup and Automata Theory
– Part I –

Barbara Schütt and Manfred Kerber

SEKI Working Paper SWP-93-02

A Mathematical Knowledge Base for Proving Theorems in Semigroup and Automata Theory – Part I –*

Barbara Schütt Manfred Kerber
Fachbereich Informatik, Universität des Saarlandes
Im Stadtwald, D-66041 Saarbrücken
Germany

Abstract

We present a mathematical knowledge base containing the factual knowledge of the first of three parts of a textbook on semi-groups and automata, namely "P. Deussen: Halbgruppen und Automaten". Like almost all mathematical textbooks this textbook is not self-contained, but there are some algebraic and set-theoretical concepts not being explained. These concepts are added to the knowledge base. Furthermore there is knowledge about the natural numbers, which is formalized following the first paragraph of "E. Landau: Grundlagen der Analysis".

The data base is written in a sorted higher-order logic, a variant of $\mathcal{POST}$, the working language of the proof development environment Ω-MKRP. We distinguish three different types of knowledge: axioms, definitions, and theorems. Up to now, there are only 2 axioms (natural numbers and cardinality), 149 definitions (like that for a semi-group), and 165 theorems. The consistency of such knowledge bases cannot be proved in general, but inconsistencies may be imported only by the axioms. Definitions and theorems should not lead to any inconsistency since definitions form conservative extensions and theorems are proved to be consequences.

*This work was supported by the Deutsche Forschungsgemeinschaft (SFB 314)

Contents

Introduction

In this paper we represent the mathematical "factual" knowledge of the first third of the textbook "Halbgruppen und Automaten" of P. Deussen [Deu71]. As mathematical textbooks in general the book is not self-contained, the additional knowledge concerns above all the natural numbers. We have formalized this knowledge following the first section of E. Landau's "Grundlagen der Analysis" [Lan30]. We present the knowledge in the form of a mathematical dictionary. The main means for our representation is a sorted higher-order logic, which is a variant of $\mathcal{POST}$ [Nes92], the working language of the proof development environment Ω-MKRP [HKK$^+$92]. Indeed one may ask why such a powerful logic is not sufficient for the description of the factual knowledge of mathematics, because logic has been developed in the last hundred years just for that purpose. The answer is that it is possible to find many extra-logical features in the presentation of mathematics and that these features and particular the structure of the knowledge are essential for mathematical activities like theorem proving.

What are the shortcomings of logic that make a particular knowledge representation necessary? One main point is that the basic notion of logic is that of a formula, but that in mathematics different kinds of formulae are distinguished, namely *axioms*, *definitions*, and *theorems* and we will subdivide these kinds even more. Furthermore in mere logic a knowledge base consists of an unstructured set of formulae, whereas text books are well-structured and mathematicians spend a lot of time on the final presentation of the mathematical content. In addition there are constraints – which are not present in logic – in the procedure of stating theorems or defining concepts. The most important constraint for definitions is that all concepts which are used in the definition – with the exception of the definiendum of course – must already be known. Analogously all concepts in theorems must be known. But even if all concepts are known, a definition has to fulfill further extra-logical requirements, for example, it is normally given in a form as abstract as possible.

Perhaps the main difference between logic and mathematics can be seen in the *conceptual* representation of mathematics. The standard schema of this procedure in mathematical text books is: "definition", "example"[1], "theorem", "proof". When we look closer at this procedure, we see that the introduction of a concept is not terminated by giving a definition, but that lemmas about the introduced concept immediately belong to the concept.

Another great difference between logic and mathematics is the assumption in mathematics – even if it cannot be proved – that a knowledge base is consistent, whereas by logic certain formulae are related, but it does not matter whether the preconditions are fulfilled or not.

Which requirements should a knowledge representation formalism for mathematics satisfy? There are the following properties we would like to see:

[1]Although very important, in this report we do not represent examples.

- The knowledge base should be consistent and the representation formalism should support to keep it consistent.
- The representation formalism should reflect the different types of knowledge, that is, axioms, definitions, and theorems should be distinguished.
- The knowledge base should be redundancy free.
- It should not be possible to use unknown concepts.
- It should be possible that the knowledge can be represented in a conceptual, structured way.
- The representation formalism should be powerful enough in order to represent the knowledge easily.
- The formalism should have a clear semantics.

The distinction of the three different kinds of knowledge is very important for the consistency: If we guarantee that definitions are really definitions and that theorems are proved, we have to be careful only with the axioms. Therefore we will use in the following three different basic knowledge units, one for axioms, one for definitions, and one for theorems. For further details see [Ker92].

Although not committed to a special system, the presented data base is intended as a tool in a proof development environment, called Ω-MKRP, which is currently developed in our group. It provides the factual knowledge the system needs to prove any theorem in a field. Of course such a data base is a changing (in particular growing) object. Therefore the presented work will be extended in the future and presents a preliminary state.

Ω-MKRP is intended as a tool for formal proofs in mathematics. It is based on the sorted higher-order logic $\mathcal{POST}$. The proofs are built up in the natural deduction calculus [Gen35]. In Ω-MKRP, a user first states a problem to be solved and then applies natural deduction inference rules in order to prove it. He can also insert a mathematical fact from the database described in this paper into the current partial proof, he can apply a domain-specific problem-solving method, or he can call an integrated automated theorem prover to solve a subproblem. The user can also pass the control to a planning component that supports and partially automates his long-range planning of a proof. Towards the important goal of user-friendliness, machine-generated proofs are transformed in several steps into much shorter, better-structured proofs that are finally translated into natural language. A detailed description of Ω-MKRP can be found in [HKK+92].

How to Read the Data Base

In this section we shortly describe the basic features of the database. Further information concerning the language $\mathcal{POST}$ can be found in [Nes92], concerning the data base format in [Ker92]. The main means for the representation of the mathematical content of the text book is a sorted higher-order logic, which is a variant of the working language of Ω-MKRP, called $\mathcal{POST}$. Furthermore we employ a format, structuring the data base in three main parts, namely axioms, definitions, and theorem. The axioms are further structured in "simple axioms" and "inductive axioms", the definitions in "simple definitions", "inductive definitions", and "implicit definitions".

The Higher-Order Language

Our sorted higher-order logic is written in a standard LISP-prefix style. Each term has a type, where types are the simple types of Alonzo Church [Chu40, And86]. There are the basic types `I` and `O` standing for the individuals and the truth values. If $\tau_1, \ldots, \tau_n, \sigma$ are types then $(\sigma\ (\tau_1\ \cdots\ \tau_n))$ is the type of the functions from $\tau_1 \times \cdots \times \tau_n$ to σ. Furthermore we use `A`, `B`, and `C` as type variables, which have to be instantiated for each concrete problem.

Terms are built up by function symbols from constants or variables of a certain type, namely if $\texttt{t}_1, \ldots, \texttt{t}_n$ are terms of types $\tau_1, \ldots, \tau_n$ and `f` is a function symbol of type $(\sigma\ (\tau_1\ \cdots\ \tau_n))$, then $(\texttt{f}\ \texttt{t}_1\ \cdots\ \texttt{t}_n)$ is a term of type σ. Each term of type `O` is an atomic formula. Formulae are either atomic or built from formulae by the connectives `not`, `and`, `or`, `impl`, `equiv` with the usual semantics or are quantified expressions. The quantifiers are the universal quantifier `all` and the existential quantifier `ex`. Each variable is bound by a quantifier and gets its type from the quantification. A quantified expression has the form

$$(<\texttt{QUANT}>(<\texttt{VAR}><\texttt{QUALIFIER}>)<\texttt{FORMULA}>)$$

where the $<\texttt{QUALIFIER}>$ is either a type or a sort.

In order to have a comfortable language for expressing mathematics, we make extensive use of a *sorted* notation, where unary predicates and sorts are used synonymously, that is, if `P` is a unary predicate of type `(O I)` then we write `(all (x P) PHI)` as synonym for `(all (x I) (impl (P x) PHI))`.

As usual we do not distinguish sets and predicates. In order to be close to the general mathematical notation, we do not use any λ-expressions in the following.

The Structure of the Texts

The data base is structured in so-called theories, which form a directed acyclic graph. In our case this structure is as follows:

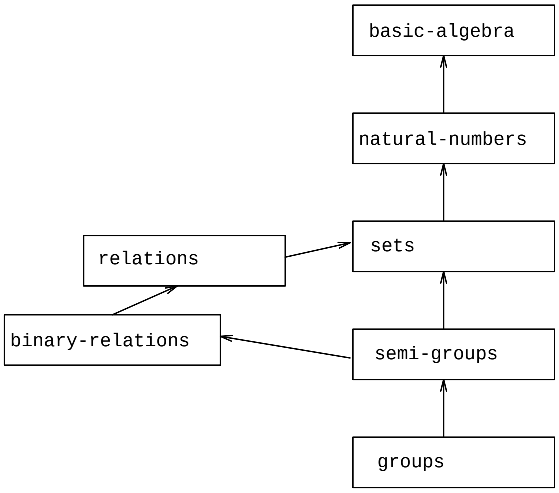


Abbildung 1: Structure of the Data Base

In this report, the data base is structured in the chapters: Basic Notions of Algebra, Natural Numbers, Set-Theory, Paragraph 1, Paragraph 2, Paragraph 3, Paragraph 4, and Paragraph 5. Each theory consists of a couple of units, namely axioms, definitions, and theorems in the form of frames. The simplest unit is that of a theorem, it consists of the slots for a formula, a status and the name of the theory it belongs to. As far as the data base is built up, no theorem is proven in Ω-MKRP, hence the status is never `"proved"` but always `"conjectured"`.

For instance our first theorem has the form

```
(theorem 1
   (theorem (all (x nat) (all (y nat)
            (impl (not= x y)
                  (not= (suc x) (suc y))))))
   (status "conjectured")
   (theory numbers))
```

Axioms are relatively rare. Up to now, we have only two axioms in the data base, a simple axiom and an inductive axiom. The simple axiom introduces the cardinality of a predicate. It consists of a formula, the extension of the signature and the name of the theory it belongs to.

```
(simple-axiom card
   (axiom (and (all (X (O A)) (all (Y (O A))
               (equiv (= (card X) (card Y))
                      (ex (f (Y X) (bijective X Y f))))))
          (all (n nat) (= (card (1-to n) n)))))
   (sign.ext card (I (O A)))
   (theory sets))
```

The inductive axiom introduces the data-type of the natural numbers by

```
(inductive-axiom nat
   (base 1 nat)
   (step suc (nat nat))
   (signature-ext (nat (O I))
                  (1 nat)
                  (suc (nat nat)))
   (theory natural-numbers))
```

It evaluates to the standard Peano axioms for the natural numbers.

The definitions have three different forms, simple definitions, inductive definitions, and implicit definitions. Simple definitions have the form:

```
(simple-definition associative
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (definition (all (x G) (all (y G) (all (z G)
               (= (f (f x y) z)
                  (f x (f y z)))))))
   (theory basic-algebra))
```

The standard slots are that of `parameters`, which form the arguments of the corresponding constant – i.e. `associative` in this case – `sort`, `definition` and `theory`. Furthermore we use slots for `preconditions`, `consequences`, `equivalences`, `superconcepts`, and `optional-parameters`. `preconditions` are formulae indicating the cases when a definition makes sense, in `consequences` lemmas concerning the defined concept are stored, the `equivalences` slot can be used to state equivalent formulae for the definition. `superconcepts` belong to the definition of a slot, but consist only of atomic formulae. `optional-parameters` need not to be given. For instance in the definition

```
(simple-definition ex-left-zero
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (z G))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f z x) z)))
   (theory semi-groups))
```

the zero `z` is an optional parameter. `(ex-left-zero G f z)` means that `z` is a zero in the semi-group `(G f)`, whereas `(ex-left-zero G f)` means that there is an element `z` such that `z` is a zero in the semi-group `(G f)`.

Inductive definitions are given in the form

```
(inductive-definition plus
   (parameters (x nat)
               (y nat))
   (sort nat)
   (base (all (x nat) (= (plus x 1)
                         (suc x))))
   (step (all (x nat) (all (y nat) (= (plus (x (suc y)))
                                      (suc (plus x y))))))
   (theory natural-numbers))
```

with separate definitions for the base and the step case.

While simple definitions are essentially abbreviations that can be easily expanded, implicit definitions describe an object implicitly in such a way that the existence and uniqueness is ensured. This must be proven however. For most cases this follows simply from the existence of the corresponding λ-expression. An example is:

```
(implicit-definition set-difference
   (parameters (X (O A))
               (Y (O A)))
   (sort (O A))
   (definition (all (x A)
               (equiv ((set-difference X Y) x)
                      (and (X x) (not (Y x))))))
   (status "conjectured")
   (theory basic-algebra))
```

Outlook and Open Problems

In this section we give a short critique of the work.

Positively it can be remarked that all relevant concepts and theorems have been stated in the given format without greater difficulties. In particular, the chosen language seems to be quite adequate for the formalization of the mathematical content presented in the two textbooks. The language is so strong that the conceptual part of the knowledge base needs no axioms, but only two, one for the axiomatization of natural numbers and one for the axiomatization of cardinality.

Negatively can be remarked that in some rare cases the type restriction is unpleasant, for instance, it is not easily possible to define pairs with arbitrary types of elements that fits well in the framework of n-tuples, so that it is not possible to have identities like $(a,(b,c)) = (a,b,c) = ((a,b),c)$ and to define n-tuples inductively as $(a_1,\ldots,a_n,a_{n+1}) = ((a_1,\ldots,a_n),a_{n+1})$. However, up to now the problems are not severe and can be bypassed.

Furthermore many definitions are a little bit cumbersome, since the symbols are atomic, that is we can not infer from the definition of a left ideal to that of a right ideal, from that of a left neutral element to that of a right neutral element and so on.

A final valuation is in work and will influence the further development of the Ω-MKRP working language.

References

[And86] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth through Proof.* Academic Press, Orlando, Florida, USA, 1986.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, **5**:56–68, 1940.

[Deu71] Peter Deussen. *Halbgruppen und Automaten*, volume 99 of *Heidelberger Taschenbücher.* Springer Verlag, Berlin, Germany, 1971.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, **39**:572–595, 1935.

[HKK+92] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, Jörn Richts, and Jörg Siekmann. Ω-MKRP – a proof development environment. SEKI Report SR-92-22, Fachbereich Informatik, Universität des Saarlandes, Im Stadtwald, Saarbrücken, Germany, 1992.

[Ker92] Manfred Kerber. *On the Representation of Mathematical Concepts and their Translation into First Order Logic.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1992.

[Lan30] Edmund Landau. *Grundlagen der Analysis.* Wissenschaftliche Buchgesellschaft, Darmstadt, Germany; second edition, 1930. Reprint of the edition, Leipzig, 1970.

[Nes92] Dan Nesmith, editor. KEIM-Manual. Version 0, 1992. Universität des Saarlandes, Im Stadtwald, Saarbrücken, Germany.

Alphabetic Index

Consecutive Index

Basic Notions of Algebra

*** $\neq$

```
(simple-definition not=
   (parameters )
   (sort O O)
   (definition (not =))
   (theory basic-algebra))
```

*** associative

```
(simple-definition associative
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (definition (all (x G) (all (y G) (all (z G)
               (= (f (f x y) z)
                  (f x (f y z)))))))
   (theory basic-algebra))
```

*** commutative

```
(simple-definition commutative
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (definition (all (x G) (all (y G) (= (f x y)
                                        (f y x)))))
   (theory basic-algebra))
```

*** subset U of G.

```
(simple-definition subset
   (parameters (U (O A))
               (G (O A)))
   (sort O)
   (definition (all (x A) (impl (U x) (G x))))
   (theory basic-algebra))
```

*** left distributive

```
(simple-definition left-distributive
   (parameters (G (O A))
               (f (G G G))
               (h (G G G)))
   (sort O)
   (definition (all (x G) (all (y G) (all (z G)
               (= (h x (f y z))
                  (f (h x y) (h x z)))))))
   (theory basic-algebra))
```

*** right distributive

```
(simple-definition right-distributive
   (parameters (G (O A))
               (f (G G G))
               (h (G G G)))
   (sort O)
   (definition (all (x G) (all (y G) (all (z G)
               (= (h (f x y) z)
                  (f (h x z) (h y z)))))))
   (theory basic-algebra))
```

*** abelian

```
(simple-definition abelian
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (definition (commutative G f))
   (theory basic-algebra))
```

*** empty set

```
(simple-definition empty-set
   (parameters )
   (sort (O A))
   (definition (all (x A) (not (empty-set x))))
   (theory basic-algebra))
```

*** non-empty

```
(simple-definition non-empty
   (parameters (U (O A)))
   (sort O)
   (definition (not= U empty-set))
   (theory basic-algebra))
```

*** empty

```
(simple-definition empty
   (parameters (U (O A)))
   (sort O)
   (definition (= U empty-set))
   (theory basic-algebra))
```

*** singleton

```
(simple-definition singleton
   (parameters (X (O A)))
   (sort O)
   (definition (and (ex (x A) (X x))
                    ((all (x X) (all (y X)
                     (= x y))))))
   (theory basic-algebra))
```

*** set-difference $X \setminus Y$

```
(implicit-definition set-difference
   (parameters (X (O A))
               (Y (O A)))
   (sort (O A))
   (definition (all (x A)
               (equiv ((set-difference X Y) x)
                      (and (X x) (not (Y x))))))
   (status "conjectured")
   (theory basic-algebra))
```

*** the composition of two mappings

```
(implicit-definition comp
   (parameters (f (A A A))
               (g (A A A)))
   (sort (A A))
   (definition (all (x A)
               (= ((comp f g) x)
                  (f (g x)))))
   (theory basic-algebra))
```

Natural Numbers

Axiom 1: 1 ist eine natürliche Zahl.

Axiom 2: Zu jedem x gibt es genau eine natürliche Zahl, die der Nachfolger von x heißt und mit x' bezeichnet werden möge.

Axiom 3: Stets ist $x' \neq 1$. D.h. es gibt keine Zahl mit dem Nachfolger 1.

Axiom 4: Aus $x' = y'$ folgt $x = y$. D.h. zu jeder Zahl gibt es keine oder genau eine, deren Nachfolger jene Zahl ist.

Axiom 5: (Induktionsaxiom) Es sei M eine Menge natürlicher Zahlen mit den Eigenschaften:
I) 1 gehört zu M.
II) Wenn x zu M gehört, so gehört x' zu M.
Dann umfaßt M alle natürlichen Zahlen.

*** inductive axiom for the definition of the natural numbers

```
(inductive-axiom nat
   (base 1 nat)
   (step suc (nat nat))
   (signature-ext (nat (O I))
                  (1 nat)
                  (suc (nat nat)))
   (theory natural-numbers))
```

Satz 1: Aus $x \neq y$ folgt $x' \neq y'$.

*** theorem 1: $\forall x, y \in \mathbb{N} x \neq y \Rightarrow x' \neq y'$

```
(theorem 1
   (theorem (all (x nat) (all (y nat)
            (impl (not= x y)
                  (not= (suc x) (suc y))))))
   (status "conjectured")
   (theory numbers))
```

Satz 2: $x' \neq x$.

*** theorem 2:$\forall x \in \mathbb{N} : x' \neq x$

```
(theorem 2
   (theorem (all (x nat) (not= (suc x) x)))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 3: Ist $x \neq 1$, so gibt es ein (also nach Axiom 4 genau ein) u mit $x = u'$.

*** theorem 3: $\forall x \in \mathbb{N} : x \neq 1 \Rightarrow \exists u : x = u'$

```
(theorem 3
   (theorem (all (x nat)
            (impl (not= x 1)
                  ((ex u nat) (= x (suc u))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 4 zugleich **Definition 1**: Auf genau eine Art läßt sich jedem Zahlenpaar x, y eine natürliche Zahl, $x + y$ genannt (+ sprich: plus), so zuordnen, daß
1) $x + 1 = x'$ für jedes x,
2) $x + y' = (x + y)'$ für jedes x und jedes y.
$x + y$ heißt die Summe von x und y oder die durch Addition von y zu x entstehende Zahl.

*** plus

```
(inductive-definition plus
   (parameters (x nat)
               (y nat))
   (sort nat)
   (base (all (x nat) (= (plus x 1)
                         (suc x))))
   (step (all (x nat) (all (y nat) (= (plus (x (suc y)))
                                      (suc (plus x y))))))
   (theory natural-numbers))
```

Satz 5: (assoziatives Gesetz der Addition) $(x + y) + z = x + (y + z)$.

*** theorem 5: The composition plus of the natural numbers is associative.

```
(theorem 5
   (theorem (associative nat plus)))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 6: (kommutatives Gesetz der Addition) $x + y = y + x$.

*** theorem 6: The composition plus of the natural numbers is commutative.

```
(theorem 6
   (theorem (commutative nat plus)))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 7: $y \neq x + y$.

*** theorem 7:$\forall x, y \in \mathbb{N} : y \neq x + y$

```
(theorem 7
   (theorem (all (x nat) (all (y nat)
            (not= y (plus x y)))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 8: Aus $y \neq z$ folgt $x + y \neq x + z$.

*** theorem 8: $\forall x, y, z \in \mathbb{N} : y \neq z \Rightarrow x + y \neq x + z$

```
(theorem 8
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (not= y z)
                  (not= (plus x y) (plus x z)))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 9: Sind x und y gegeben, so liegt genau einer der Fälle vor:
1) $x = y$.
2) Es gibt ein (also nach Satz 8 genau ein) u mit $x = y + u$.
3) Es gibt ein (also nach Satz 8 genau ein) v mit $y = x + v$.

*** theorem 9a: $\forall x, y \in \mathbb{N}$ we have one of the following cases:
$x = y \quad \vee \quad \exists u \in \mathbb{N} : x = y + u \quad \vee \quad \exists v \in \mathbb{N} : y = x + v$

```
(theorem 9a
   (theorem (all (x nat) (all (y nat)
            (or (or (= x y)
                    (ex (u nat) (= x (plus y u))))
                (ex (v nat) (= y (plus x v)))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 9b: $\forall x, y \in \mathbb{N}$ with $x = y$
$\Rightarrow \quad \neg \exists u \in \mathbb{N} : x = y + u$
$\wedge \quad \neg \exists v \in \mathbb{N} : y = x + v$

```
(theorem 9b
   (theorem (all (x nat) (all (y nat)
            (impl (= x y)
                  (and (not (ex (u nat)
                                (= x (plus y u))))
                       (not (ex (v nat)
                                (= y (plus x v)))))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 9c: $\forall x, y \in \mathbb{N}$ for which there is a $u \in \mathbb{N}$ with $x = y + u$
$\Rightarrow \quad x \neq y$
$\wedge \quad \neg \exists v \in \mathbb{N} : y = x + v$

```
(theorem 9c
   (theorem (all (x nat) (all (y nat)
            (impl (ex (u nat) (= x (plus y u)))
                  (and (not (= x y))
                       (not (ex (v nat) (= y (plus x v))
                   )))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 9d: $\forall x, y \in \mathbb{N}$ for which there is a $v \in \mathbb{N}$ with $y = x + v$

$$\begin{array}{lll} \Rightarrow & & x \neq y \\ & \wedge & \neg\exists u \in \mathbb{N} : x + y + u \end{array}$$

```
(theorem 9d
   (theorem (all (x nat) (all (y nat)
            (impl (ex (v nat) (= y (plus x v)))
                  (and (not (= x y))
                       (not (ex (u nat) (= x (plus y u))
                   ))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Def.2: Ist $x = y + u$, so ist $x > y$. ($>$ sprich: größer als.)

*** greater than

```
(simple-definition >
   (parameters (x nat)
               (y nat))
   (sort O)
   (definition (ex (u nat) (= x (plus y u))))
   (theory natural-numbers))
```

Def.3: Ist $y = x + v$, so ist $x < y$. ($<$ sprich: kleiner als.)

*** smaller than

```
(simple-definition <
   (parameters (x nat)
               (y nat))
   (sort O)
   (definition (ex (v nat) (= y (plus x v))))
```

Satz 10: Sind x, y beliebig, so liegt genau einer der Fälle $x = y, x > y, x < y$ vor.

*** theorem 10a: $\forall x, y \in \mathbb{N}$ is
$x = y \quad \vee \quad x > y \quad \vee \quad x < y.$

```
(theorem 10a
   (theorem (all (x nat) (all (y nat)
            (or (or (= x y) (> x y)) (< x y)))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 10b: $\forall x, y \in \mathbb{N}$ with $x = y$
$\Rightarrow x \not> y \wedge x \not< y$

```
(theorem 10b
   (theorem (all (x nat) (all (y nat)
            (impl (= x y)
                  (and (not (> x y)) (not (< x y)))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 10c: $\forall x, y \in \mathbb{N}$ with $x > y$
$\Rightarrow x \neq y \ \wedge \ x \not< y$.

```
(theorem 10c
   (theorem (all (x nat) (all (y nat)
            (impl (> x y)
                  (and (not (= x y)) (not (< x y)))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 10d: $\forall x, y \in \mathbb{N}$ with $x < y$
$\Rightarrow x \neq y \ \wedge \ x \not> y$.

```
(theorem 10d
   (theorem (all (x nat) (all (y nat)
            (impl (< x y)
                  (and (not (= x y)) (not (> x y)))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 11: Aus $x > y$ folgt $y < x$.

*** theorem 11:$\forall x, y \in \mathbb{N} : x > y \Rightarrow y < x$

```
(theorem 11
   (theorem (all (x nat) (all (y nat)
            (impl (> x y) (< y x)))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 12: Aus $x < y$ folgt $y > x$.

*** theorem 12: $\forall x, y \in \mathbb{N} : x < y \Rightarrow y > x$

```
(theorem 12
   (theorem (all (x nat) (all (y nat)
            (impl (< x y) (> y x)))))
   (status "conjectured")
   (theory natural-numbers))
```

Def.4: $x \geq y$ bedeutet $x > y$ oder $x = y$. ($\geq$ sprich: größer oder gleich.)

*** greater or equal

```
(simple-definition >=
   (parameters (x nat)
               (y nat))
   (sort O)
   (definition (or (> x y) (= x y)))
   (theory natural-numbers))
```

Def.5 $x \leq y$ bedeutet $x < y$ oder $x = y$. ($\leq$ sprich: kleiner oder gleich.)

*** smaller or equal

```
(simple-definition <=
   (parameters (x nat)
               (y nat))
   (sort O)
   (definition (or (< x y) (= x y)))
   (theory natural-numbers))
```

Satz 13: Aus $x \geq y$ folgt $y \leq x$.

*** theorem 13: $\forall x, y \in \mathbb{N} : x \geq y \Rightarrow y \leq x$

```
(theorem 13
   (theorem (all (x nat) (all (y nat)
            (impl (>= x y) (<= y x)))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 14: Aus $x \leq y$ folgt $y \geq x$.

*** theorem 14: $\forall x, y \in \mathbb{N} : x \leq y \Rightarrow y \geq x$

```
(theorem 14
   (theorem (all (x nat) (all (y nat)
             (impl (<= x y) (>= y x)))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 15: (Transitivität der Ordnung) Aus $x < y$, $y < z$ folgt $x < z$.

*** theorem 15: transitivity of $<$

```
(theorem 15-(transitivity-of-<)
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (and (< x y) (< y z))
                  (< x z))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 16: Aus $x \leq y$, $y < z$ oder $x < y$, $y \leq z$ folgt $x < z$.

*** theorem 16a: $\forall x, y, z \in \mathbb{N}$:
$x \leq y \;\wedge\; y < z \;\Rightarrow\; x < z$.

```
(theorem 16a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (and (<= x y) (< y z))
                  (< x z))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 16b: $\forall x, y, z \in \mathbb{N}$:
$x < y \ \wedge\ y \leq z \ \Rightarrow\ x < z.$

```
(theorem 16b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (and (< x y) (<= y z))
                  (< x z))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 17 Aus $x \leq y,\ y \leq z$ folgt $x \leq z$.

*** theorem 17: transitivity of $\leq$

```
(theorem 17-(transitivity-of-<=)
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (and (<= x y) (<= y z))
                  (<= x z))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 18: $x + y > x$.

*** theorem 18: $\forall x, y, z \in \mathbb{N}$:
$x + y > x$

```
(theorem 18
   (theorem (all (x nat) (all (y nat)
            (> (plus x y) x))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 19: Aus $x > y$ bzw. $x = y$ bzw. $x < y$ folgt
$x + z > y + z$ bzw. $x + z = y + z$ bzw. $x + z < y + z$.

*** theorem 19a: $\forall x, y, z \in \mathbb{N}$:
$x > y \ \Rightarrow\ x + z > y + z.$

```
(theorem 19a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (> x y)
                  (> (plus x z) (plus y z)))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 19b: $\forall x, y, z \in \mathbb{N}$:
$x = y \ \Rightarrow\ x + z = y + z.$

```
(theorem 19b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (= x y)
                  (= (plus x z) (plus y z)))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 19c: $\forall x, y, z \in \mathbb{N}$:
$x < y \;\Rightarrow\; x + z < y + z$.

```
(theorem 19c
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (< x y)
                  (< (plus x z) (plus y z)))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 20: Aus $x + z > y + z$ bzw. $x + z = y + z$ bzw. $x + z < y + z$ folgt $x > y$ bzw. $x = y$ bzw. $x < y$.

*** theorem 20a: $\forall x, y, z \in \mathbb{N}$:
$x + z > y + z \;\Rightarrow\; x > y$.

```
(theorem 20a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (> (plus x z) (plus y z))
                  (> x y))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 20b: $\forall x, y, z \in \mathbb{N}$:
$x + z = y + z \;\Rightarrow\; x = y$.

```
(theorem 20b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (= (plus x z) (plus y z))
                  (= x y))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 20c: $\forall x, y, z \in \mathbb{N}$:
$x + z < y + z \;\Rightarrow\; x < y$.

```
(theorem 20c
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (< (plus x z) (plus y z))
                  (< x y))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 21: Aus $x > y$, $z > u$ folgt $x + z > y + u$.

*** theorem 21: $\forall x, y, z \in \mathbb{N}$:
$x > y \wedge z > u \Rightarrow x + z > y + u$

```
(theorem 21
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (> x y) (> z u))
                  (> (plus x z) (plus y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 22: Aus $x \geq y,\ z > u$ oder $x > y,\ z \geq u$ folgt $x + z > y + u$.

*** theorem 22a: $\forall x, y, z, u \in \mathbb{N}$:
$x \geq y \ \wedge \ z > u \ \Rightarrow \ x + z > y + u$.

```
(theorem 22a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (>= x y) (> z u))
                  (> (plus x z) (plus y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 22b: $\forall x, y, z, u \in \mathbb{N}$:
$x > y \ \wedge \ z \geq u \ \Rightarrow \ x + z > y + u$.

```
(theorem 22b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (> x y) (>= z u))
                  (> (plus x z) (plus y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 23: Aus $x \geq y,\ z \geq u$ folgt $x + z \geq y + u$.

*** theorem 23: $\forall u, x, y, z \in \mathbb{N}$:
$x \geq y,\ z \geq u \Rightarrow x + z \geq y + u$

```
(theorem 23
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (>= x y) (>= z u))
                  (>= (plus x z) (plus y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 24: $x \geq 1$.

*** theorem 24: $\forall x \in \mathbb{N}$:
$x \geq 1$

```
(theorem 24
   (theorem (all (x nat) (>= x 1)))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 25: Aus $y > x$ folgt $y \geq x + 1$.

*** theorem 25: $\forall x, y \in \mathbb{N}$:
$y > x \Rightarrow y \geq x + 1$

```
(theorem 25
   (theorem (all (x nat) (all (y nat)
            (impl (> y x)
                  (>= y (plus x 1))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 26: Aus $y < x + 1$ folgt $y \leq x$.

*** theorem 26:$\forall x, y \in \mathbb{N}$:
$y < x + 1 \Rightarrow y \leq x$

```
(theorem 26
   (theorem (all (x nat) (all (y nat)
            (impl (< y (plus x 1))
                  (<= y x)))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 27: In jeder nichtleeren Menge natürlicher Zahlen gibt es eine kleinste (d.h. eine, die kleiner ist als jede etwaige andere).

*** theorem 27: Each nonempty set of natural numbers has a least element.

```
(theorem 27
   (theorem (all (U (O I))
            (impl (and (subset U nat)
                       (non-empty U))
                  (ex (n U) (all (u U) (<= n u))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 28: zugleich **Definitions 6** Auf genau eine Art läßt sich jedem Zahlenpaar x, y eine natürliche Zahl, $x \cdot y$ genannt ($\cdot$ sprich: mal; aber man schreibt den Punkt meist nicht), so zuordnen, daß
1) $x \cdot 1 = x$ für jedes x
2) $x \cdot y' = x \cdot y + x$ für jedes x und jedes y
$x \cdot y$ heißt das Produkt von x mit y oder die durch Multiplikation von x mit y entstehende Zahl.

*** product of natural numbers

```
(inductive-definition product
   (parameters (x nat)
               (y nat))
   (sort nat)
   (base (all (x nat) (= (product x 1) x)))
   (step (all (x nat) (all (y nat)
         (= (product x (suc y))
            (plus (product x y) x)))))
   (theory natural-numbers))
```

Satz 29: (kommutatives Gesetz der Multiplikation) $xy = yx$.

*** theorem 29: The multiplication of the natural numbers is commutative.

```
(theorem 29
   (theorem (commutative nat product))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 30: (distributives Gesetz) $x(y + z) = xy + xz$.

*** theorem 30: left distributive law for the natural numbers

```
(theorem 30
   (theorem (left-distributive nat plus product))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 31: (assoziatives Gesetz der Multiplikation) $(xy)z = x(yz)$.

*** theorem 31: The multiplication of the natural numbers is associative.

```
(theorem 31
   (theorem (associative nat product))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 32: Aus $x > y$ bzw. $x = y$ bzw. $x < y$ folgt $xz > yz$ bzw. $xz = yz$ bzw. $xz < yz$.

*** theorem 32a: $\forall x, y, z \in \mathbb{N}$: $x > y \;\Rightarrow\; xz > yz$.

```
(theorem 32a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (> x y)
                  (> (product x z) (product y z))
                   )))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 32b: $\forall x, y, z \in \mathbb{N}$: $x = y \;\Rightarrow\; xz = yz$.

```
(theorem 32b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (= x y)
                  (= (product x z) (product y z))
                   )))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 32c: $\forall x, y, z \in \mathbb{N}$: $x < y \;\Rightarrow\; xz < yz$.

```
(theorem 32c
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (< x y)
                  (< (product x z) (product y z))
                   )))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 33: Aus $xz > yz$ bzw. $xz = yz$ bzw. $xz < yz$ folgt $x > y$ bzw. $x = y$ bzw. $x < y$.

*** theorem 33a: $\forall x, y, z \in \mathbb{N}$: $xz > yz \;\Rightarrow\; x > y$.

```
(theorem 33a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (> (product x z) (product y z))
                  (> x y))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 33b: $\forall x, y, z \in \mathbb{N}$:
$xz = yz \;\Rightarrow\; x = y$.

```
(theorem 33b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (= (product x z) (product y z))
                  (= x y))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 33c: $\forall x, y, z \in \mathbb{N}$:
$xz < yz \;\Rightarrow\; x < y$.

```
(theorem 33c
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (impl (< (product x z) (product y z))
                  (< x y))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 34: Aus $x > y$, $z > u$ folgt $xz > yu$.

*** theorem 34: $\forall u, x, y, z \in \mathbb{N}$:
$x > y \wedge z > u \Rightarrow xz > yu$

```
(theorem 34
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (> x y) (> z u))
                  (> (product x z) (product y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 35: Aus $x \geq y$, $z > u$ oder $x > y$, $z \geq u$ folgt $xz > yu$.

*** theorem 35a: $\forall x, y, z, u \in \mathbb{N}$:
$x \geq y \;\wedge\; z > u \;\Rightarrow\; xz > yu$.

```
(theorem 35a
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (>= x y) (> z u))
                  (> (product x z) (product y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

*** theorem 35b: $\forall x, y, z, u \in \mathbb{N}$:
$x > y \ \wedge \ z \geq u \ \Rightarrow \ xz > yu.$

```
(theorem 35b
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (> x y) (>= z u))
                  (> (product x z) (product y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Satz 36: Aus $x \geq y$, $z \geq u$ folgt $xz \geq yu$.

*** theorem 36: $\forall u, x, y, z \in \mathbb{N}$:
$x \geq y \wedge z \geq u \Rightarrow xz \geq yu$

```
(theorem 36
   (theorem (all (x nat) (all (y nat) (all (z nat)
            (all (u nat)
            (impl (and (>= x y) (>= z u))
                  (>= (product x z) (product y u))))))))
   (status "conjectured")
   (theory natural-numbers))
```

Set-Theory

*** complex product UV

```
(implicit-definition complex-product
   (parameters (G (O A))
               (U (O A))
               (V (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (and (subset U G)
                       (subset V G)))
   (definition (all (x A)
               (equiv ((complex-product G U V f) x)
                      ((ex (y U) (ex (z V)
                                 (= x (f y z))))))))
   (status "conjectured")
   (theory basic-algebra))
```

*** left complex product uV

```
(implicit-definition left-complex-product
   (parameters (G (O A))
               (u G)
               (V (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (subset V G))
   (definition (all (x A)
               (equiv ((left-complex-product G u V f) x)
                      (ex (v V) (= x (f u v))))))
   (status "conjectured")
   (theory sets))
```

*** right complex product Uv

```
(implicit-definition right-complex-product
   (parameters (G (O A))
               (U (O A))
               (v G)
               (f (G G G)))
   (sort (O A))
   (preconditions (subset U G))
   (definition (all (x A)
               (equiv ((right-complex-product G U v f) x)
                      (ex (u U) (= x (f u v))))))
   (status "conjectured")
   (theory sets))
```

*** idempotent

```
(simple-definition idempotent
   (parameters (G (O A))
               (f (G G G))
               (e G))
   (sort O)
   (definition (= (f e e) e))
   (theory sets))
```

*** injective mapping

```
(simple-definition injective
   (parameters (G (O A))
               (H (O A)))
   (optional-parameters (f (H G)))
   (sort O)
   (definition (all (x G) (all (y G) (impl (= (f x) (f y))
                                           (= x y)))))
   (theory sets))
```

*** surjective mapping

```
(simple-definition surjective
   (parameters (G (O A))
               (H (O A)))
   (optional-parameters (f (H G)))
   (sort O)
   (definition (all (y H) (ex (x G) (= (f x) y))))
   (theory sets))
```

*** bijective mapping

```
(simple-definition bijective
   (parameters (G (O A))
               (H (O A)))
   (optional-parameters (f (H G)))
   (sort O)
   (definition (and (injective G H f)
                    (surjective G H f)))
   (theory sets))
```

*** set of the first n numbers

```
(implicit-definition 1-to
   (parameters (n nat))
   (sort (O I))
   (definition (all (x I) (all (n nat)
               (equiv ((1-to n) x)
                      (and (and (nat x) (<= x n))
                           (<= 1 x))))))
   (status "conjectured")
   (theory sets))
```

*** finite

```
(simple-definition finite
   (parameters (S (O A)))
   (sort O)
   (definition (ex (n nat) (bijective (1-to n) S)))
   (theory sets))
```

*** intersection $X \cap Y$

```
(implicit-definition intersection
   (parameters (X (O A))
               (Y (O A)))
   (sort (O A))
   (definition (all (a A) (equiv ((intersection X Y) a)
                                 (and (X a) (Y a)))))
   (status "conjectured")
   (theory sets))
```

*** intersection $\bigcap_{i \in J} U_i$

```
(implicit-definition fam-intersection
   (parameters (J (O A))
               (U ((O B) J)))
   (sort (O B))
   (definition (all (x B)
               (equiv ((fam-intersection J U) x)
                      (all (i J) ((U i) x)))))
   (status "conjectured")
   (theory sets))
```

*** intersection $\bigcap_{i=1}^{n} U_i$

```
(simple-definition n-intersection
   (parameters (n nat)
               (U ((O A) (1-to n))))
   (sort (O A))
   (definition (fam-intersection (1-to n) U))
   (consequences ((= (n-intersection 1 U) (U 1))
                  (status "conjectured"))
                 ((all (n nat) (all (U ((O A) (1-to (suc n))))
                  (= (n-intersection (suc n) U)
                     (intersection (U (suc n))
                                   (n-intersection n U)))))
                  (status "conjectured")))
   (theory sets))
```

*** union $X \cup Y$

```
(implicit-definition union
   (parameters (X (O A))
               (Y (O A)))
   (sort O)
   (definition (all (a A) (equiv ((union X Y) a)
                                 (or (X a) (Y a)))))
   (status "conjectured")
   (theory sets))
```

*** union $\bigcup_{i \in J} U_i$

```
(implicit-definition fam-union
   (parameters (J (O A))
               (U ((O B) J)))
   (sort (O B))
   (definition (all (x B)
               (equiv ((fam-union J U) x)
                      (ex (i J) ((U i) x)))))
   (status "conjectured")
   (theory sets))
```

*** union $\bigcup_{i=1}^{n} U_i$

```
(simple-definition n-union
   (parameters (n nat)
               (U ((O A) (1-to n))))
   (sort (O A))
   (definition (fam-union (1-to n) U))
   (consequences ((= (n-union 1 U) (U 1))
                  (status "conjectured"))
                 ((all (n nat) (all (U ((O I) (1-to (suc n))))
                  (= (n-union (suc n) U)
                    (union (U (suc n)) (n-union n U)))))
                  (status "conjectured")))
   (theory sets))
```

*** The pair (x, y) $(x, y \in A)$ is defined as a mapping from $\{1, 2\}$ to A by $\big((pair\ x\ y)\ 1\big) = x$ and $\big((pair\ x\ y)\ 2\big) = y$
The other alternative would be to define the pair by the axiom:
$(a, b) = (c, d) \quad \Leftrightarrow \quad a = c\ \wedge\ b = d$
We have not made this choice, since then we would have an additional axiom.

```
(simple-definition pair
   (parameters (x A)
               (y A))
   (sort (A (1-to (suc 1))))
   (definition (and (= x ((pair x y) 1))
                    (= y ((pair x y) (suc 1)))))
   (consequences ((all (x A) (all (y A) (all (s A) (all (t A)
                  (impl (= (pair x y) (pair s t))
                        (and (= x s) (= y t)))))))
                  (status "conjectured")))
   (theory sets))
```

*** The cartesian product of two sets X and Y $\big(X, Y \subset (O\ A)\big)$ is defined as a mapping from $\{1, 2\}$ to $(O\ A)$ by $a \in X \times Y$
$\Leftrightarrow \qquad \exists x \in X \quad \vee \quad \exists y \in Y$ with $a = (x, y) \quad \forall a \in A$

```
(implicit-definition 2-cart-prod
   (parameters (X (O A))
               (Y (O A)))
   (sort ((O A) (1-to (suc 1))))
   (definition (all (a (A (1-to (suc 1))))
               (equiv ((2-cart-prod X Y) a)
                      (ex (x X) (ex (y Y)
                          (= a (pair x y)))))))
   (theory sets))
```

*** Like the pair it's possible to define the n-tuple by the projection mappings to the different components. It's useful to have the definition of the n-tuple although x as mapping is already an n-tuple!
One consequence of the definition is that two n-tuples could only be equal iff they have the same length and each component is identical.

```
(simple-definition n-tuple
   (parameters (n nat)
               (x (A (1-to n))))
   (sort (A (1-to n)))
   (definition (all (i (1-to n))
               (= (x i) ((n-tuple n x) i))))
   (consequences ((all (n nat) (all (x (A (1-to n))
                   (all (m nat) (all (y (A (1-to m))
                   (impl (= (n-tuple n x) (n-tuple m y))
                        (and (= n m)
                             (all (i (1-to n))
                                     (= (x i) (y i))))))))))))
                   (status "conjectured")))
   (theory sets))
```

*** In analogy to the definition of the cartesian product of two sets we can also define the cartesian product of n sets $X_1 \times X_2 \times \ldots \times X_n$.

```
(implicit-definition n-cart-prod
   (parameters (n nat)
               (X ((O A) (1-to n))))
   (sort ((O A) (1-to n)))
   (definition (all (a A)
               (equiv ((n-cart-prod n X) a)
                      (all (i (1-to n))
                      (ex (x (A (1-to n)))
                      (and ((X i) (x i))
                           (= a (n-tuple n x)))))))
               (status "conjectured"))
   (theory sets))
```

*** Here we will define the cartesian product of only one set X with itself, that is $X \times X \times \ldots \times X$ (n-times).

```
(implicit-definition n-same-cart-prod
   (parameters (n nat)
               (X (O A)))
   (sort ((O A) (1-to n)))
   (definition (all (a A)
               (equiv ((n-same-cart-prod n X) a)
                      (ex (x (A (1-to n)))
                      (and (X (x i))
                           (= a (n-tuple n x))))))
               (status "conjectured")
   (theory sets))
```

*** axiom of the cardinality

```
(simple-axiom card
   (axiom (and (all (X (O A)) (all (Y (O A))
               (equiv (= (card X) (card Y))
                      (ex (f (Y X) (bijective X Y f))))))
          (all (n nat) (= (card (1-to n) n)))))
   (sign.ext card (I (O A)))
   (theory sets))
```

Paragraph 1

1.1 Def.: Eine *Halbgruppe* ist eine Menge F zusammen mit einer zweistelligen Verknüpfung $\Phi : F \times F \to F$, die üblicherweise als "Produkt" geschrieben wird $fg := \Phi(f,g)$ und für das gilt $(fg)h = f(gh)$ $(f,g,h \in G)$.

*** semigroup

```
(simple-definition semi-group
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (definition (associative G f))
   (theory semi-groups))
```

1.2 Def.: Eine Halbgruppe F heißt *kommutativ* oder *abelsch*, wenn gilt $fg = gf$ für alle $f, g \in F$.

*** abelian

```
(simple-definition abelian
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (definition (commutative G f))
   (theory basic-algebra))
```

Def.: Sind $U, V \subset F$ Teilmengen, so bezeichnet man als *Komlexprodukt* die Menge $UV := \{uv : u \in U, v \in V\}$ und schreibt insbesondere $uV := \{u\}V$ bzw. $Uv := U\{v\}$, wenn eine der Mengen einelementig ist.

*** UV is called complex product.

```
(implicit-definition complex-product
   (parameters (G (O A))
               (U (O A))
               (V (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (and (subset U G)
                       (subset V G)))
   (definition (all (x A)
               (equiv ((complex-product G U V f) x)
                      ((ex (y U) (ex (z V)
                                 (= x (f y z))))))))
   (status "conjectured")
   (theory sets))
```

*** uV is called left complex product.

```
(implicit-definition left-complex-product
   (parameters (G (O A))
               (u G)
               (V (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (subset V G))
   (definition (all (x A)
               (equiv ((left-complex-product G u V f) x)
                      (ex (v V) (= x (f u v))))))
   (status "conjectured")
   (theory sets))
```

*** Uv is called right complex product.

```
(implicit-definition right-complex-product
   (parameters (G (O A))
               (U (O A))
               (v G)
               (f (G G G)))
   (sort (O A))
   (preconditions (subset U G))
   (definition (all (x A)
               (equiv ((right-complex-product G U v f) x)
                      (ex (u U) (= x (f u v))))))
   (status "conjectured")
   (theory sets))
```

1.3 Def.: Eine nichtleere Teilmenge $U \subset F$ heißt *Unterhalbgruppe*, wenn gilt $UU \subset U$.

*** sub-semigroup U of G

```
(simple-definition sub-semi-group
   (parameters (U (O A))
               (G (O A))
               (f (G G G)))
   (definition (and (and (and (semi-group G f)
                              (subset U G))
                         (nonempty U))
                    (all(x U) (all (y U)
                    (U (f x y))))))
   (equivalences (and (and (semi-group G f)
                           (nonempty U))
                      (subset (complex-product G U U f) U))
                 (status "conjectured"))
   (theory semi-groups))
```

1.4 Def.: Ein Element n_l $[n_r]$ von F heißt *Linksnull* [*Rechtsnull*], wenn für jedes $f \in F$ gilt: $n_l f = n_l$ $[f n_r = n_r]$. Ein Element e_l $[e_r]$ von F heißt *Linkseins* [*Rechtseins*], wenn für jedes $f \in F$ gilt: $e_l f = f$ $[f e_r = f]$.

*** existence of the left zero

```
(simple-definition ex-left-zero
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (z G))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f z x) z)))
   (theory semi-groups))
```

*** existence of the right zero

```
(simple-definition ex-right-zero
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (z G))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f x z) z)))
   (theory semi-groups))
```

*** existence of the left neutral element

```
(simple-definition ex-left-neutral-element
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (n G))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f n x) x)))
   (theory semi-groups))
```

*** existence of the right neutral element

```
(simple-definition ex-right-neutral-element
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (n G))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f x n) x)))
   (theory semi-groups))
```

1.5 Def.: Ein Element $0 \in F$ heißt *Nullelement* oder *Null*, wenn für jedes $f \in F$ gilt $0f = f0 = 0$. Ein Element $1 \in F$ heißt *Einselement* oder *Eins*, wenn für jedes $f \in F$ gilt $1f = f1 = f$.

*** zero

```
(simple-definition zero
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (z G))
   (sort O)
   (superconcepts (ex-left-zero G f z)
                  (ex-right-zero G f z))
   (definition T)
   (theory semi-groups))
```

*** existence of the neutral element

```
(simple-definition ex-neutral-element
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (n G))
   (sort O)
   (superconcepts (ex-left-neutral-element G f n)
                  (ex-right-neutral-element G f n))
   (definition T)
   (theory semi-groups))
```

*** function one

```
(implicit-definition one
   (parameters (G (O A))
               (f (G G G)))
   (sort G)
   (preconditions (and (semi-group G f)
                       (ex-neutral-element G f n)))
   (definition n)
   (status "conjectured")
   (theory semi-groups))
```

Def.: Eine Halbgruppe mit Einselement wird auch *Monoid* genannt.

*** monoid

```
(simple-definition monoid
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (n G))
   (sort O)
   (definition (and (semi-group G f)
                    (ex-neutral-element G f n))))
   (theory semi-groups))
```

1.6 Def.: Eine *Gruppe* F ist eine Halbgruppe mit Eins, in der es zu jedem $f \in F$ ein Element $f^{-1} \in F$ gibt, so daß $f^{-1}f = 1$ ist (Existenz der *Linksinversen*).

*** existence of a left inverse element

```
(simple-definition ex-left-inverse-element
   (parameters (G (O A))
               (f (G G G))
               (n G))
   (optional-parameters (inv (G G)))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f (inv x) x) n)))
   (theory semi-groups))
```

*** existence of a right inverse element

```
(simple-definition ex-right-inverse-element
   (parameters (G (O A))
               (f (G G G))
               (n G))
   (optional-parameters (inv (G G)))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (all (x G) (= (f x (inv x)) n))
   (theory semi-groups))
```

*** existence of an inverse element

```
(simple-definition ex-inverse-element
   (parameters (G (O A))
               (f (G G G))
               (n G))
   (optional-parameters (inv (G G)))
   (sort O)
   (superconcepts (semi-group G f))
   (definition (and (ex-left-inverse-element G f n inv)
                    (ex-right-inverse-element G f n inv)))
   (theory semi-groups))
```

*** group

```
(simple-definition group
   (parameters (G (O A))
               (f (G G G)))
   (optional-parameters (n G)
                        (inv (G G)))
   (sort O)
   (superconcepts (associative G f)
                  (ex-neutral-element G f n)
                  (ex-inverse-element G f n inv))
   (definition T)
   (equivalences ((and (and (semi-group G f)
                            (ex-left-neutral-element G f n))
                       (ex-left-inverse-element G f n inv))
                  (status "conjectured"))
                 ((and (and (semi-group G f)
                            (ex-right-neutral-element G f n))
                       (ex-right-inverse-element G f n inv))
                  (status "conjectured"))
                 ((and (and (associative G f)
                            (ex-left-neutral-element G f n))
                       (ex-left-inverse-element G f n inv))
                  (status "conjectured"))
                 ((and (and (associative G f)
                           (ex-right-neutral-element G f n))
                     (ex-right-inverse-element G f n inv))
                  (status "conjectured"))
                 ((and (monoid G f n)
                       (ex-inverse-element G f n inv))
                  (status "conjectured")))
   (theory groups))
```

1.7 Def.: Eine Teilmenge $U \subset F$ einer Halbgruppe F heißt *Untergruppe*, wenn sie bezüglich der Halbgruppenverknüpfung in F eine Gruppe ist.

*** subgroup

```
(simple-definition sub-group
   (parameters (U (O A))
               (G (O A))
               (f (G G G)))
   (optional-parameters (n G)
                        (inv (U U)))
   (sort O)
   (preconditions (and (subset U G)
                       (semi-group G f)))
   (definition (group U f n inv))
   (theory groups))
```

1.8 Def.: Ein Element e einer Halbgruppe F heißt *idempotent*, wenn $ee = e$.

*** idempotent

```
(simple-definition idempotent
   (parameters (G (O A))
               (f (G G G))
               (e G))
   (sort O)
   (definition (= (f e e) e))
   (theory sets))
```

1.9 Satz: Ist F eine Gruppe, $U \subset F$ eine Untergruppe und sei $1_U \in U$ das Einselement von U, so ist $1 = 1_U$.

*** theorem 1.9: Let G be a group, $U \subset G$ a subgroup, $1_U \in U$ the one of U, then: $1 = 1_U$.

```
(theorem 1.9
   (theorem (all G (O A) (all U (O A) (all f (G G G)
            (all  (n G) (all (inv (G G)) (all (nu U)
            (impl (and (group G f n inv)
                       (sub-group U G f nu))
                  (= n nu))))))))
   (status "conjectured")
   (theory groups))
```

1.10 Satz: Ist F eine Gruppe und $U \subset F$ eine Teilmenge, so ist U Untergruppe genau dann, wenn $v^{-1}u \in U$ für alle $u, v \in U$. Ist ferner U eine endliche Menge, so ist U Untergruppe genau dann, wenn $UU \subset U$, d.h. wenn U Unterhalbgruppe in F ist.

*** theorem 10.a: If G is a group, $U \subset G$, then:
U is a subgroup of $G \Leftrightarrow \forall u, v \in U : v^{-1}u \in U$

```
(theorem 1.10a
   (theorem (all (G (O A)) (all (U (O A)) (all (f (G G G))
            (all (n G) (all (inv (G G))
            (impl (and (group G f n inv) (subset U G))
                  (equiv (sub-group U G f n inv)
                         (all (u U) (all (v U)
                         (U (f (inv v) u))))))))))))
   (status "conjectured")
   (theory groups))
```

*** theorem 10b: If G is a group, U a finite subset of G, then
U is subgroup of $G \Leftrightarrow U$ is a sub-semigroup of G.

```
(theorem 1.10b
   (theorem (all (G (O A)) (all (U (O A)) (all (f (G G G))
            (all (n G) (all (inv (G G))
            (impl (and (and (group G f n inv)
                                 (subset U G))
                            (finite U))
                  (equiv (sub-group U G f n inv)
                         (sub-semi-group U G f)))))))))
   (status "conjectured")
   (theory groups))
```

1.11 Satz: Ist $\{U_i : i \in I\}$ eine Familie von Unterhalbgruppen einer Halbgruppe F, so ist auch der Durchschnitt $\bigcap_{i \in I} U_i$ eine Unterhalbgruppe von F, sofern der Durchschnitt nicht leer ist.

*** theorem 1.11: If $\{U_i : i \in I\}$ is a family of sub-semigroups of a semigroup G, then $\bigcap_{i \in I} U_i$ is also a subsemigroup of G iff this intersection isn't empty.

```
(theorem 1.11
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (U ((O A) J))
            (impl (and (and (semi-group G f)
                            (all (i J) (sub-semi-group (U i) G f)))
                            (not= empty-set (fam-intersection J U)))
             (sub-semi-group (fam-intersection J U) G f)))))))
   (status "conjectured")
   (theory semi-groups))
```

1.12 Satz: Ist $\{U_i : i \in I\}$ eine Familie von Untergruppen einer Gruppe F, so ist auch der Durchschnitt $\bigcap_{i \in I} U_i$ eine Untergruppe von F.

*** theorem 1.12: If $\{U_i : i \in I\}$ is a family of subgroups of a group G, then also $\bigcap_{i \in I} U_i$ is a subgroup of G.

```
(theorem 1.12
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (U ((O A) J))
            (impl (and (group G f)
                       (all (i J) (sub-group (U i) G f)))
            (sub-group (fam-intersection J U) G f)))))))
   (status "conjectured")
   (theory groups))
```

1.13 Def.: $U \sqcup V := \bigcap_{W \supset U \cup V} W$. $U \sqcup V$ ist also die kleinste aller Unterhalbgruppen von F, die $U \cup V$ enthält.

*** intersection of all sub-semigroups W of a group G containing the union of the sub-semigroups U and V of G
$U \sqcup V := \bigcap_{W \supseteq U \cup V} W$

```
(implicit-definition sqcup-sub-semi-group
   (parameters (G (O A))
               (U (O A))
               (V (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (and (sub-semi-group U G f)
                       (sub-semi-group V G f)))
   (definition (all (x A)
               (equiv ((sqcup-sub-semi-group G U V f) x)
                      (all (W (O A))
                      (impl (and (sub-semi-group W G f)
                                 (subset (union U V) W))
                            (W x))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 1.13a: $U \sqcup V$ is a sub-semigroup of G

```
(theorem 1.13a
   (theorem (all (G (O A)) (all (f (G G G)) (all (U (O A))
            (all (V (O A))
            (impl (and (sub-semi-group U G f)
                       (sub-semi-group V G f))
                  (sub-semi-group
                     (sqcup-sub-semi-group G U V f) G f)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 1.13b: $U \sqcup V$ is the smallest sub-semigroup of G containing $U \cup V$

```
(theorem 1.13b
   (theorem (all (G (O A)) (all (f (G G G)) (all (U (O A))
            (all (V (O A)) (all (W (O A))
            (impl (and (and (and (sub-semi-group U G f)
                                 (sub-semi-group V G f))
                            (sub-semi-group W G f))
                       (subset (union U V) W))
                  (subset (sqcup-sub-semi-group G U V f)
                          W))))))))
   (status "conjectured")
   (theory semi-groups))
```

1.14 Def.: Sei $E \subset F$ eine Teilmenge und E^n induktiv definiert durch $E^1 := E$, $E^{n+1} := EE^n$, so heißt $E^* := \bigcup_{n \in \mathbb{N}} E^n$ die *von E erzeugte Unterhalbgruppe.*

*** E^n

```
(inductive-definition n-product
   (parameters (G (O A))
               (f (G G G))
               (E (O A))
               (n nat))
   (sort (O A))
   (preconditions (and (subset E G) (semi-group G f)))
   (base (all (E (O A)) (all (G (O A)) (all (f (G G G))
         (= (n-product G f E 1) E)
   (step (all (n nat) (all (E (O A)) (all (G (O A))
         (all (f (G G G))
         (= (n-product G f E (suc n))
         (complex-produkt G E (n-product G f E n) f)))))))
   (theory semi-groups))
```

*** generated sub-semigroup from E in G

```
(implicit-definition generated-sub-semi-group
   (parameters (G (O A))
               (E (O A))
               (f (G G G)))
   (preconditions (and (semi-group G f) (subset E G)))
   (definition (all (x A)
             (equiv ((generated-sub-semi-group G f E) x)
                    (ex (n nat) ((n-product G f E n) x)))))
   (status "conjectured")
   (theory semi-groups))
```

1.15 Satz: $U \sqcup V = (U \cup V)^*$.

*** theorem 1.15: $U \sqcup V = (U \cup V)^*$

```
(theorem 1.15
   (theorem (all (G (O A)) (all (U (O A)) (all (V (O A))
          (all (f (G G G))
      (impl (and (sub-semi-group U G f)
                 (sub-semi-group V G f))
         (= (sqcup-sub-semi-group G U V f)
          (generated-sub-semi-group G (union U V) f))))))))))
   (status "conjectured")
   (theory semi-groups))
```

1.16 Def.: Sei $E \subset F$ eine Teilmenge der Halbgruppe F, so heißt E ein *Erzeugendensystem* von F, wenn $E^* = F$.

*** generating system

```
(simple-definition generating-system
   (parameters (G (O A))
               (E (O A))
               (f (G G G)))
   (sort O)
   (preconditions (and (semi-group G f)
                       (subset E G)))
   (definition (= (generated-sub-semi-group G E f) G))
   (theory semi-groups))
```

1.17 Def.: Eine Halbgruppe F heißt *linkskürzbar* [*rechtskürzbar*], wenn für alle $f, g, h \in F$ aus $fg = fh$ [$gf = hf$] stets $g = h$ folgt.

*** left cancellable

```
(simple-definition left-cancellable
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (semi-group G f))
   (definition (all (x G) (all (y G) (all (z G)
               (impl (= (f (x y)) (f (x z)))
                     (= y z))))))
   (theory semi-groups))
```

*** right cancellable

```
(simple-definition right-cancellable
   (parameters (G (O A))
               (f (G G G)))
   (preconditions (semi-group G f))
   (sort O)
   (definition (all (x G) (all (y G) (all (z G)
               (impl (= (f (y x)) (f (z x)))
                     (= y z))))))
   (theory semi-groups))
```

1.18 Def.: f ist *Linksteiler* [*Rechtsteiler*] von g genau dann, wenn $f = g$ ist oder wenn es ein $h \in F$ gibt, so daß $fh = g$ [$hf = g$].

*** x is left divisor of y

```
(simple-definition left-divisor
   (parameters (G (O A))
               (f (G G G))
               (x G)
               (y G))
   (sort O)
   (definition (or (= x y)
                   (ex (z G) (= (f x z) y))))
   (theory semi-groups))
```

*** x is right divisor of y

```
(simple-definition right-divisor
   (parameters (G (O A))
               (f (G G G))
               (x G)
               (y G))
   (sort O)
   (definition (or (= x y)
                   (ex (z G) (= (f z x) y))))
   (theory semi-groups))
```

Paragraph 2

2.1 Def.: Seien F und H Halbgruppen (oder Gruppen). Eine Abbildung $\varphi : F \to H$ heißt *Homomorphismus* (von F in H), wenn gilt $\varphi(fg) = \varphi(f)\varphi(g)$ $(f, g \in F)$.

*** semigroup-homomorphism

```
(simple-definition semi-group-homomorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H))
               (phi (H G)))
   (sort O)
   (preconditions (and (semi-group G g)
                       (semi-group H h)))
   (definition (all (x G) (all (y G)
               (= (phi (g x y))
               (h (phi x) (phi y))))))
   (theory semi-groups))
```

*** group-homomorphism

```
(simple-definition group-homomorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H))
               (phi (H G)))
   (sort O)
   (preconditions (and (group G g)
                       (group H h)))
   (definition (all (x G) (all (y G)
               (= (phi (g x y))
               (h (phi x) (phi y))))))
   (theory groups))
```

noch 2.1: Spezielle Homomorphismen: Ist $F = H$, so heißt φ *Endomorphismus.*

*** semigroup-endomorphism

```
(simple-definition semi-group-endomorphism
   (parameters (G (O A))
               (g (G G G))
               (phi (G G))
   (sort O)
   (superconcepts (semi-group-homomorphism G g G g phi))
   (definition T)
   (theory semi-groups))
```

*** group-endomorphism

```
(simple-definition group-endomorphism
   (parameters (G (O A))
               (g (G G G))
               (phi (G G))
   (sort O)
   (superconcepts (group-homomorphism G g G g phi))
   (definition T)
   (theory groups))
```

noch 2.1: Ist φ injektiv, so heißt φ *Monomorphismus.*

*** semigroup-monomorphism

```
(simple-definition semi-group-monomorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (optional-parameters (phi (H G)))
   (sort O)
   (superconcepts (semi-group-homomorphism G g H h phi)
                  (injective G H phi))
   (definition T)
   (theory semi-groups))
```

*** group-monomorphism

```
(simple-definition group-monomorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (optional-parameters (phi (H G)))
   (sort O)
   (superconcepts (group-homomorphism G g H h phi)
                  (injective G H phi))
   (definition T)
   (theory groups))
```

noch 2.1: Ist φ surjektiv, so heißt φ *Epimorphismus* und H ein *homomorphes Bild* von F.

*** semigroup-epimorphism

```
(simple-definition semi-group-epimorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (optional-parameters (phi (H G)))
   (sort O)
   (superconcepts (semi-group-homomorphism G g H h phi)
                  (surjective G H phi))
   (definition T)
   (theory semi-groups))
```

*** group-epimorphism

```
(simple-definition group-epimorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (optional-parameters (phi (H G)))
   (sort O)
   (superconcepts (group-homomorphism G g H h phi)
                  (surjective G H phi))
   (definition T)
   (theory groups))
```

noch 2.1: Ist φ bijektiv, so heißt φ *Isomorphismus* und F und H sind *isomorph* ($F \simeq H$).

*** semigroup-isomorphism

```
(simple-definition semi-group-isomorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (optional-parameters (phi (H G)))
   (sort O)
   (superconcepts (semi-group-homomorphism G g H h phi)
                  (bijective G H phi))
   (definition T)
   (equivalences ((and (semi-group-monomorphism G g H h phi)
                       (semi-group-epimorphism G g H h phi))
                  (status "conjectured")))
   (theory semi-groups))
```

*** group-isomorphism

```
(simple-definition group-isomorphism
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (optional-parameters (phi (H G)))
   (sort O)
   (superconcepts (group-homomorphism G g H h phi)
                  (bijective G H phi))
   (definition T)
   (equivalences ((and (group-monomorphism G g H h phi)
                       (group-epimorphism G g H h phi))
                  (status "conjectured")))
   (theory groups))
```

noch 2.1: Ist $F = H$ und φ bijektiv, so heißt φ *Automorphismus*.

*** semigroup-automorphism

```
(simple-definition semi-group-automorphism
   (parameters (G (O A))
               (g (G G G)))
   (optional-parameters (f (G G)))
   (sort O)
   (superconcepts (semi-group-endomorphism G g f)
                  (bijective G G f))
   (definition T)
   (equivalences ((semi-group-isomorphism G g G g f)
                  (status "conjectured")))
   (theory semi-groups))
```

*** group-automorphism

```
(simple-definition group-automorphism
   (parameters (G (O A))
               (g (G G G)))
   (optional-parameters (f (G G)))
   (sort O)
   (superconcepts (group-endomorphism G g f)
                  (bijective G G f))
   (definition T)
   (equivalences ((group-isomorphism G g G g f)
                  (status "conjectured")))
   (theory groups))
```

noch 2.1: $Hom(F, H)$ bezeichnet die Menge aller Homomorphismen $\varphi : F \to H$.

*** set of semigroup-homomorphisms

```
(implicit-definition set-of-semi-group-hom
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (sort (O (H G)))
   (preconditions (and (semi-group G g)
                       (semi-group H h)))
   (definition (all (f (H G))
               (equiv ((set-of-semi-group-hom G g H h) f)
                      (semi-group-homomorphism G g H h f))))
   (theory semi-groups))
```

*** set of group-homomorphisms

```
(implicit-definition set-of-group-hom
   (parameters (G (O A))
               (g (G G G))
               (H (O A))
               (h (H H H)))
   (sort (O (H G)))
   (preconditions (and (group G g)
                       (group H h)))
   (definition (all (f (H G))
               (equiv ((set-of-group-hom G g H h) f)
                      (group-homomorphism G g H h f))))
   (theory groups))
```

2.2 Def.: Eine Teilmenge $E \subset F$ heißt *freies Erzeugendensystem*, wenn gilt:
1. $E^* = F$ und
2. Zu jeder Halbgruppe H und jeder Abbildung $\varphi_0 : E \to H$ gibt es ein $\varphi \in Hom(F; H)$, so daß $\varphi \mid E = \varphi_0$, und in diesem Fall heißt F *frei (über E)*.

*** E is a free generating system over G.

```
(simple-definition free-generating-system
   (parameters (E (O A))
               (G (O A))
               (g (G G G))
   (preconditions (and (subset E G) (semi-group G g)))
   (sort O)
   (definition (and (= (generated-sub-semi-group G E g) G))
                    (all (H (O A)) (all (h (H H H))
               (impl (semi-group H h)
                     (all (phi0 (H E)) (ex (phi (H G))
                 (and (semi-group-homomorphism G g H h phi))
                      (all (x E)
                      (= (phi x) (phi0 x))))))))))
   (theory semi-groups))
```

*** G is free over E

```
(simple-definition free-over
   (parameters (G (O A))
               (g (G G G)))
   (optional-parameters (E (O A)))
   (sort O)
   (superconcepts (free-generating-system E G g))
   (definition T)
   (theory semi-groups))
```

Def.: Wir bilden mit einer nichtleeren Menge X, dem *Alphabet*, induktiv die kartesischen Produkte $X_1 := X$, $X_{n+1} := X \times X_n$.

*** Here we will define the cartesian product of only one set X with itself, that is $X \times X \times \cdots \times X$ (n-times).

```
(implicit-definition n-same-cart-prod
   (parameters (n nat)
               (X (O A)))
   (sort ((O A) (1-to n)))
   (definition (all (a A)
               (equiv ((n-same-cart-prod n X) a)
                      (ex (x (A (1-to n)))
                      (and (X (x i))
                           (= a (n-tuple n x))))))
               (status "conjectured")
   (theory sets))
```

Def.: Die Vereinigung $W(X) := \bigcup_{n \in \mathbb{N}} X_n$ heißt dann die Menge aller *Worte über* X.

*** the set of all words over X

```
(implicit-definition set-of-all-words-over
   (parameters (X (O A)))
   (sort ((O A) nat))
   (definition (equiv
                 (all (w (A nat)) ((set-of-all-words-over X) w)
                 (ex (n nat) ((n-same-cart-prod n X) w)))))
   (theory semi-groups))
```

*** w is word over X

```
(simple-definition word-over
   (parameters (X (O A))
               (w (A nat)))
   (sort O)
   (definition ((set-of-all-words-over X) w))
   (theory semi-groups))
```

Def.: Zur Vereinfachung der Schreibweise wollen wir für ein Wort $u \in W(X)$ auch $u = x_1 x_2 \ldots x_n := (x_1, x_2, \ldots, x_n)$ schreiben und nennen dabei $\mid u \mid := n$ die *Länge* von u.

*** length of a word

```
(implicit-definition length-of-word
   (parameters (X (O A))
               (w (A nat))
   (sort nat)
   (preconditions (and (word-over X w)
                       (ex (n nat)
                       ((n-same-cart-prod n X) w))))
   (definition n)
   (status "conjectured")
   (theory semi-groups))
```

2.3 Def.: Um nun $W(X)$ zu einer Halbgruppe zu machen, wird eine Verknüpfung für je zwei Worte $u = x_1 x_2 \ldots x_n$ und $v = y_1 y_2 \ldots y_m$ ($x_i, y_k \in X$), die *Konkatenation* von u und v, durch $u^\frown v := x_1 x_2 \ldots x_n y_1 y_2 \ldots y_m$ definiert, und auch hier kürzen wir, wenn keine Verwechslungen zu befürchten sind, ab: $uv := u^\frown v$.

*** theorem: every word of length n is an n-tuple.

```
(theorem word=n-tuple
   (theorem (all (X (O A)) (all (w (A nat)) (all (n nat)
            (impl (= n (lenght-of-word X w))
                  (ex (x (A (1-to n))) (= w (n-tuple n x))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** concatenation

```
(implicit-definition concatenation
   (parameters (X (O A))
               (u (A nat))
               (v (A nat)))
   (optional-parameters (n nat)
                        (m nat))
   (sort (A nat))
   (preconditions (and (= n (length-of-word X u))
                       (= m (length-of-word X v))))
   (definition (all (w (A nat))
               (equiv (= (concatenation X u v n m) w)
                      (and (and (= (length-of-word X w)
                                   (plus n m))
                           (all (i (1-to n))
                                (= (w i) (u i))))
                           (all (k (1-to m))
                                (= (w (plus n k)) (v k)))))))
   (status "conjectured")
   (theory semi-groups))
```

2.4 Def.: Die Menge $W(X)$ versehen mit der Verknüpfung (2.3) als Halbgruppenverknüpfung heißt die *Worthalbgruppe über* X. Der *Rang* von $W(X)$ ist $card(X)$.

*** theorem 2.4: If X is not empty, then the set of all words over X forms a semigroup together with the concatenation.

```
(theorem 2.4
   (theorem (all (X (O A))
         (impl (not= X empty-set)
               (semi-group (set-of-all-words-over X)
                           concatenation ))))
   (status "conjectured")
   (theory semi-groups))
```

*** the rank of the word semigroup

```
(simple-definition rank-of-word-semi-group-over
   (parameters (X (O A)))
   (sort (A (O A))))
   (definition (card X))
   (theory semi-groups))
```

2.5a Satz: Jedes $u \in W(X)$ ist eindeutig als Produkt von Elementen $x \in X$ darstellbar.

*** theorem 2.5a: Every element $u \in W(X)$ is uniquely writeable as a product of elements $x \in X$.

```
(theorem 2.5a
   (theorem (all (X (O A)) (all (w (A nat)
            (ex (x (A (1-to (length-of-word X w))))
            (ex (y (A (1-to (length-of-word X w))))
            (impl (and (and (word-over X w)
                       (= w (n-tuple (length-of-word X w) x)))
                       (= w (n-tuple (length-of-word X w) y)))
                  (all (i (1-to (length-of-word X w)))
                  (= (x i) (y i))))))))))
   (status "conjectured")
   (theory semi-groups))
```

2.5b Satz: $W(X)$ ist sowohl links- wie rechtskürzbar.

*** theorem 2.5b1: $W(X)$ is left cancellable.

```
(theorem 2.5b1
   (theorem (all (X (O A))
          (left-cancellable (set-of-all-words-over X)
                            concatenation)))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 2.5b2: $W(X)$ is right cancellable.

```
(theorem 2.5b2
   (theorem (all (X (O A))
          (right-cancellable (set-of-all-words-over X)
                             concatenation)))
   (status "conjectured")
   (theory semi-groups))
```

2.6 Satz: $W(X) \simeq W(Y)$ genau dann, wenn $card(X) = card(Y)$.

*** theorem 2.6: $W(X) \simeq W(Y) \Leftrightarrow card(X) = card(Y)$.

```
(theorem 2.6
   (theorem (all (X (O A)) (all (Y (O A))
            (equiv (semi-group-isomorphism
                         (words-over X) concatenation
                         (words-over Y) concatenation)
                   (= (card X) (card Y))))))
   (status "conjectured")
   (theory semi-groups))
```

2.7 Satz: Die Halbgruppe F ist genau dann frei, wenn es eine Menge X gibt, so daß $F \simeq W(X)$.

*** theorem 2.7: The semigroup G is free iff there is a set X with $F \simeq W(X)$.

```
(theorem 2.7
   (theorem (all (G (O A)) (all (g (G G G))
            (impl (semi-group G g)
                  (equiv (free-over G g)
                         (ex (X (O A))
                         (semi-group-isomorphism G g
                                        (words-over X)
                                        concatenation)))))))
   (status "conjectured")
   (theory semi-groups))
```

2.8 Def.: Sei $\{F_i : i = 1, \ldots, n\}$ eine endliche Familie von Halbgruppen. Das *direkte Produkt* $\times_{i=1}^{n} F_i$ ist diejenige Halbgruppe, deren zugrunde liegende Menge das kartesische Produkt $F_1 \times F_2 \times \ldots \times F_n$ ist, mit der Verknüpfung $(f_1, f_2, \ldots, f_n)(g_1, g_2, \ldots, g_n) := (f_1 g_1, f_2 g_2, \ldots, f_n g_n)$, wobei $f_i, g_i \in F_i$ ist.

*** The direct product φ is a composition
$\varphi : \underset{i=1}{\overset{n}{\times}} F_i \times \underset{i=1}{\overset{n}{\times}} F_i \longrightarrow \underset{i=1}{\overset{n}{\times}} F_i$
It can be defined componentwise within every semigroup by:
$f_i : F_i \times F_i \longrightarrow F_i$ für $i = 1, \ldots, n$
If $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \underset{i=1}{\overset{n}{\times}} F_i$, then is
$\varphi\big((x_1, \ldots, x_n), (y_1, \ldots, y_n)\big) =$
$= \big(f_1(x_1, y_1), \ldots, f_n(x_n, y_n)\big) = (z_1, \ldots, z_n)$
That means $\forall i \in \{1, \ldots, n\}$: we have $z_i = f_i(x_i, y_i)$.

```
(implicit-definition comp-direct-product
   (parameters (n nat)
               (F ((O A) (1-to n)))
               (f (((O A) (1-to n)) ((O A) (1-to n))
                                    ((O A) (1-to n)))))
   (sort A)
   (preconditions (all (i (1-to n))
                  (and (and ((F i) (x i))
                            ((F i) (y i)))
                       (semi-group (F i) (f i)))))
   (definition (all (x (A (1-to n))) (n-tuple n x)
               (all (y (A (1-to n))) (n-tuple n y)
               (all (z (A (1-to n))) (n-tuple n z)
               (equiv (= (comp-direct-product n x y F f) z)
                      (all (i (1-to n))
                      (= (z i) ((f i) (x i) (y i)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 2.8: The n-cartesian product of F forms a semigroup together with the direct product.

```
(theorem 2.8
   (theorem (all (n nat) (all (F ((O I) (1-to n)))
                 (semi-group (n-cart-prod n F)
                             comp-direct-product))))
   (status "conjectured")
   (theory semi-groups))
```

Paragraph 3

3.1 Def.: Eine nichtleere Teilmenge $T \subset F$ einer Halbgruppe F heißt *Linksideal* [*Rechtsideal*], wenn $FT \subset T$ [$TF \subset T$] gilt. Ist T gleichzeitig Links- und Rechtsideal, so heißt T ein *zweiseitiges Ideal*.

*** left ideal

```
(simple-definition left-ideal
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (and (and (subset T G)
                            (semi-group G f))
                       (not= empty-set T)))
   (definition (subset (complex-product G G T f) T))
   (theory semi-groups))
```

*** right ideal

```
(simple-definition right-ideal
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (and (and (subset T G)
                            (semi-group G f))
                       (not= empty-set T)))
   (definition (subset (complex-product G T G f) T))
   (theory semi-groups))
```

*** two-sided ideal

```
(simple-definition two-sided-ideal
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (definition (and (right-ideal T G f)
                    (left-ideal T G f)))
   (theory semi-groups))
```

3.2 Satz: Ist T ein Links- oder Rechtsideal oder ein zweiseitiges Ideal, so ist T Unterhalbgruppe.

*** theorem 3.2a: If T is a left ideal of the semigroup G then T is a sub-semigroup of G.

```
(theorem 3.2a
   (theorem (all (G (O A)) (all (T (O A)) (all (f (G G G))
           (impl (and (semi-group G f) (left-ideal T G f))
                 (sub-semi-group T G f))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.2b: If T is a right ideal of the semigroup G then T is a sub-semigroup of G.

```
(theorem 3.2b
   (theorem (all (G (O A)) (all (T (O A)) (all (f (G G G))
            (impl (and (semi-group G f) (right-ideal T G f))
                  (sub-semi-group T G f))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.2c: If T is a two-sided ideal of the semigroup G then T is a sub-semigroup of G.

```
(theorem 3.2c
   (theorem (all (G (O A)) (all (T (O A)) (all (f (G G G))
           (impl (and (semi-group G f) (two-sided-ideal T G f))
                 (sub-semi-group T G f))))))
   (status "conjectured")
   (theory semi-groups))
```

3.3 Satz: Sei $\{T_i : i \in I\}$ eine Familie von Links- [Rechts-, zweiseitigen] Idealen der Halbgruppe F, so ist auch $\bigcup_{i \in I} T_i$ ein Links- [Rechts-, zweiseitiges] Ideal, und ebenso $\bigcap_{i \in I} T_i$, sofern dieser Durchschnitt nicht leer ist.

*** theorem 3.3a: If $\{T_i : i \in I\}$ is a family of left ideals, then $\bigcup_{i \in I} T_i$ is a left ideal too.

```
(theorem 3.3a
   (theorem (all (G (O A)) (all (f (G G G) (all (J (O A))
            (all (T ((O A) J))
            (impl (and (semi-group G f) (all (i J)
                        (left-ideal (T i) G f)))
                  (left-ideal (fam-union J T) G f))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.3b: If $\{T_i : i \in I\}$ is a family of right ideals, then $\bigcup_{i \in I} T_i$ is a right ideal too.

```
(theorem 3.3b
   (theorem (all (G (O A)) (all (f (G G G) (all (J (O A))
            (all (T ((O A) J))
            (impl (and (semi-group G f) (all (i J)
                        (right-ideal (T i) G f)))
                  (right-ideal (fam-union J T) G f))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.3c: If $\{T_i : i \in I\}$ is a family of twosided ideals, then $\bigcup_{i \in I} T_i$ is a two-sided ideal too.

```
(theorem 3.3c
   (theorem (all (G (O A)) (all (f (G G G) (all (J (O A))
            (all (T ((O A) J))
            (impl (and (semi-group G f) (all (i J)
                        (two-sided-ideal (T i) G f)))
                  (two-sided-ideal (fam-union J T) G f))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.3d: If $\{T_i : i \in I\}$ is a family of left ideals, then $\bigcap_{i \in I} T_i$ is a left ideal too if this intersection isn't empty.

```
(theorem 3.3d
   (theorem (all (G (O A)) (all (f (G G G) (all (J (O A))
            (all (T ((O A) J))
            (impl (and (and (semi-group G f) (all (i J)
                             (left-ideal (T i) G f)))
                      (not= empty-set (fam-intersection J T)))
                  (left-ideal (fam-intersection J T) G f))
                                                     ))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.3e: If $\{T_i : i \in I\}$ is a family of right ideals, then $\bigcap_{i\in I} T_i$ is a left ideal too if this intersection isn't empty.

```
(theorem 3.3e
   (theorem (all (G (O A)) (all (f (G G G) (all (J (O A))
            (all (T ((O A) J))
            (impl (and (and (semi-group G f) (all (i J)
                            (right-ideal (T i) G f)))
                      (not= empty-set (fam-intersection J T)))
                  (right-ideal (fam-intersection J T) G f))
                                                      ))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.3f: If $\{T_i : i \in I\}$ is a family of twosided ideals, then $\bigcap_{i\in I} T_i$ is a two-sided ideal too if this intersection isn't empty.

```
(theorem 3.3f
   (theorem (all (G (O A)) (all (f (G G G) (all (J (O A))
            (all (T ((O A) J))
            (impl (and (and (semi-group G f) (all (i J)
                            (two-sided-ideal (T i) G f)))
                     (not= empty-set (fam-intersection J T)))
                 (two-sided-ideal (fam-intersection J T) G f))
                                                        ))))))
   (status "conjectured")
   (theory semi-groups))
```

3.4 Satz: $R \cap L \subset RL$.

*** theorem 3.4: $R \cap L \subset RL$

```
(theorem 3.4
   (theorem (all (G (O A)) (all (f (G G G)) (all (R (O A))
            (all (L (O A))
            (impl (and (and (semi-group G f)
                            (left-ideal L G f))
                       (right-ideal R G f))
                  (subset (intersection R L)
                          (complex-product G R L f))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.5 Def.: Ein Linksideal [Rechtsideal, zweiseitiges Ideal] $T \subset F$ heißt *minimal*, wenn für jedes Linksideal [Rechtsideal, zweiseitiges Ideal] M gilt: aus $M \subset T$ folgt $M = T$.

*** minimal left ideal

```
(simple-definition min-left-ideal
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (left-ideal T G f))
   (definition (all (M (O A)) (left-ideal M G f)
               (impl (subset M T)
                     (= M T))))
   (theory semi-groups))
```

*** minimal right ideal

```
(simple-definition min-right-ideal
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (right-ideal T G f))
   (definition (all (M (O A)) (right-ideal M G f)
               (impl (subset M T)
                     (= M T))))
   (theory semi-groups))
```

*** minimal two-sided ideal

```
(simple-definition min-two-sided-ideal
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (two-sided-ideal T G f))
   (definition (all (M (O A)) (two-sided-ideal M G f)
               (impl (subset M T)
                     (= M T))))
   (theory semi-groups))
```

3.6 Satz: Für das Links- [Rechts-] Ideal $T \subset F$ sind die folgenden Aussagen äquivalent:
a) T ist minimal
b) für jedes $t \in T$ gilt $Ft = T$ $[tF = T]$
c) zu je zwei $t_1, t_2 \in T$ gibt es ein $f \in F$ mit $ft_1 = t_2$ $[t_1 f = t_2]$.

*** theorem 3.6a: For a left ideal $T \subset G$ are the following statements equivalent:
T is minimal $\Leftrightarrow$ $\forall t \in T : Gt = T$

```
(theorem 3.6a
   (theorem (all (G (O A)) (all (f (G G G)) (all (T (O A))
         (impl (and (semi-group G f)
                    (left-ideal T G f))
               (equiv (minimal-left-ideal T G f)
                      (all (t T)
                      (= (right-complex-product G G t f)
                         T))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.6b: For a left ideal $T \subset G$ are the following statements equivalent:
T is minimal $\quad\Leftrightarrow\quad \forall t_1 \in T \forall t_2 \in T\ \exists x \in G : x t_1 = t_2$.

```
(theorem 3.6b
   (theorem (all (G (O A)) (all (f (G G G)) (all (T (O A))
            (impl (and (semi-group G f)
                       (left-ideal T G f))
                  (equiv (minimal-left-ideal T G f)
                         (all (t1 T) (all (t2 T)
                              (ex (x G)
                              (= (f x t1) t2)))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.6c: For a right ideal $T \subset G$ are the following statements equivalent:
T is minimal $\quad\Leftrightarrow\quad \forall t \in T : tG = T$.

```
(theorem 3.6c
   (theorem (all (G (O A)) (all (f (G G G)) (all (T (O A))
            (impl (and (semi-group G f)
                       (right-ideal T G f))
                  (equiv (minimal-right-ideal T G f)
                         (all (t T)
                         (= (left-complex-product G t G f)
                            T))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.6d: For a right ideal $T \subset G$ are the following statements equivalent:
T is minimal $\quad\Leftrightarrow\quad \forall t_1 \in T \forall t_2 \in T\ \exists x \in G : t_1 x = t_2$.

```
(theorem 3.6d
   (theorem (all (G (O A)) (all (f (G G G)) (all (T (O A))
            (impl (and (semi-group G f)
                       (right-ideal T G f))
                  (equiv (minimal-right-ideal T G f)
                         (all (t1 T) (all (t2 T)
                              (ex (x G)
                              (= (f t1 x) t2)))))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.7 Def.: F erfüllt die *Minimalbedingung für Links- [Rechts-, zweiseitige] Ideale*, wenn jede nichtleere Teilmenge M von Links- [Rechts-, zweiseitigen] Idealen ein minimales Element enthält, d.h. ein Links- [Rechts-, zweiseitiges] Ideal, das kein Element von M echt umfaßt.

*** class of the minimal left ideals

```
(implicit-definition class-of-min-left-ideals
   (parameters (G (O A))
               (f (G G G)))
   (sort (O (O A)))
   (preconditions (semi-group G f))
   (definition (all (X (O A))
               (equiv ((class-of-min-left-ideals G f) X)
                      (min-left-ideal X G f))))
   (status "conjectured")
   (theory semi-groups))
```

*** fulfilment of the minimal condition for left ideals

```
(simple-definition fulfil-min-cond-for-left-ideals
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (semi-group G f))
   (definition (all (M (O (O A)))
                    (subset M (class-of-min-left-ideals G f))
               (impl (not= empty-set M)
                     (ex (T M)
                       (all (U (class-of-min-left-ideals G f))
                       (impl (subset U T)
                             (= U T)))))))
   (theory semi-groups))
```

*** class of the minimal right ideals

```
(implicit-definition class-of-min-right-ideals
   (parameters (G (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (semi-group G f))
   (definition (all (X (O A))
               (equiv ((class-of-min-right-ideals G f) X)
                      (min-right-ideal X G f))))
   (status "conjectured")
   (theory semi-groups))
```

*** fulfilment of the minimal condition for right ideals

```
(simple-definition fulfil-min-cond-for-right-ideals
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (semi-group G f))
   (definition (all (M (O (O A)))
                    (subset M (class-of-min-right-ideals G f))
               (impl (not= empty-set M)
                     (ex (T M)
                       (all (U (class-of-min-right-ideals G f))
                       (impl (subset U T)
                             (= U T)))))))
   (theory semi-groups))
```

*** class of the minimal two-sided ideals

```
(implicit-definition class-of-min-two-sided-ideals
   (parameters (G (O A))
               (f (G G G)))
   (sort (O A))
   (preconditions (semi-group G f))
   (definition (all (X (O A))
               (equiv ((class-of-min-two-sided-ideals G f) X)
                      (min-two-sided-ideal X G f))))
   (status "conjectured")
   (theory semi-groups))
```

*** fulfilment of the minimal condition for two-sided ideals

```
(simple-definition fulfil-min-cond-for-two-sided-ideals
   (parameters (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (semi-group G f))
   (definition (all (M (O (O A)))
                    (subset M
                          (class-of-min-two-sided-ideals G f))
          (impl (not= empty-set M)
                (ex (T M)
                  (all (U (class-of-min-two-sided-ideals G f))
                  (impl (subset U T)
                        (= U T)))))))

   (theory semi-groups))
```

3.8 Satz: Sind L_i und L_j zwei minimale Linksideale, so ist $L_i l = L_j$ $(l \in L_j)$.

*** theorem 3.8: If L_i and L_j are two minimal left ideals then $\forall l \in L_j$ holds $L_i l = L_j$.

```
(theorem 3.8
   (theorem (all (G (O A)) (all (f (G G G)) (all (Li (O A))
            (all (Lj (O A))
            (impl (and (min-left-ideal Li G f)
                       (min-left-ideal Lj G f))
                  (all (l Lj)
                    (= Lj (right-complex-product G Li l f)
                  ))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.9 Satz: Die Halbgruppe F erfülle die Minimalbedingung für Links- [Rechts-] Ideale. Dann ist $K = \bigcup_{\lambda \in \Lambda} L_\lambda$ [$= \bigcup_{\rho \in P} R_\rho$] das kleinste zweiseitige Ideal von F.

*** For the sake of shortness we will not use for the rest of this paragraph - specially at 3.10 - greek letters. We make the following transcriptions:
$K \mapsto M \qquad \lambda \mapsto i \qquad \Lambda \mapsto J \qquad \rho \mapsto k \qquad P \mapsto K$

theorem 3.9a: If the semigroup G fulfils the minimal condition for leftideals, then $M = \bigcup_{i \in J} L_i$ is the minimal two-sided ideal of G.

```
(theorem 3.9a
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (L ((O A) J))
        (impl (and (and (and (semi-group G f)
                   (fulfil-min-cond-for-left-ideals G f))
                   (all (i J) (min-left-ideal (L i) G f)))
                   (all (X (class-of-min-left-ideals G f))
                        (ex (i J) (= X (L i)))))
              (and (two-sided-ideal (fam-union J L) G f)
                   (impl (all (M (O A))
                              (and (two-sided-ideal M G f)
                                   (subset M (fam-union J L)))
                         (= (fam-union J L) M))))))))))
   (status "conjectured")
   (definition (all (X (O A))
               (equiv ((class-of-min-left-ideals G f) X)
                      (min-left-ideal X G f))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.9b: If the semigroup G fulfils the minimal condition for rightideals, then $M = \bigcup_{k \in K} R_k$ is the minimal two-sided ideal of G.

```
(theorem 3.9b
   (theorem (all (G (O A)) (all (f (G G G)) (all (K (O B))
            (all (R ((O A) K))
        (impl (and (and (and (semi-group G f)
                   (fulfil-min-cond-for-right-ideals G f))
                   (all (k K) (min-right-ideal (R k) G f)))
                   (all (X (class-of-min-right-ideals G f))
                        (ex (k K) (= X (R k)))))
              (and (two-sided-ideal (fam-union K R) G f)
                   (impl (all (M (O I))
                              (and (two-sided-ideal M G f)
                                   (subset M (fam-union K R)))
                   (= (fam-union K R) M))))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.10 Satz: Die Halbgruppe F erfülle die Minimalbedingung für Rechts- und Linksideale. Dann gilt:

1. $K = \bigcup_{\lambda \in \Lambda} L_\lambda = \bigcup_{\rho \in P} R_\rho$ ist das kleinste zweiseitige Ideal in F.
2. $R_\rho \cap L_\lambda = R_\rho L_\lambda$
3. $G_{\rho\lambda} := R_\rho L_\lambda$ sind Unterhalbgruppen (in the original erroneously: Untergruppen)
4. Die Gruppen $G_{\rho\lambda}$ sind paarweise isomorph. Die Isomorphismen erhält man wie folgt: Sind $e_{\rho\lambda} \in G_{\rho\lambda}$ die Einselemente der Gruppen $G_{\rho\lambda}$, so gilt: $G_{\rho\lambda} e_{\rho\mu} = G_{\rho\mu} \qquad e_{\sigma\lambda} G_{\rho\lambda} = G_{\sigma\lambda}$ und die Zuordnungen $g \to g e_{\rho\mu} \qquad g \to e_{\sigma\lambda} g$ $(g \in G_{\rho\lambda})$ sind Isomorphismen.

*** attention! $\lambda \mapsto i, \quad \rho \mapsto k, \quad \mu \mapsto m, \quad \sigma \mapsto l, \quad \Lambda \mapsto J, \quad P \mapsto K$

theorem 3.10a: $\bigcup_{i \in J} L_i = \bigcup_{k \in K} R_k$

```
(theorem 3.10a
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K))
      (impl (and (and (and (and (and (and (semi-group G f)
                 (fulfil-min-cond-for-left-ideals G f))
                 (fulfil-min-cond-for right-ideals G f))
                 (all (i J) (min-left-ideal (L i) G f)))
                 (all (X (class-of-min-left-ideals G f))
                        (ex (i J) (= X (L i)))))
                 (all (k K) (min-right-ideal (R k) G f)))
                 (all (X (class-of-min-right-ideals G f))
                        (ex (k K) (= X (R k))))))
            (and (min-two-sided-ideal (fam-union J L) G f)
                 (= (fam-union J L) (fam-union K R))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.10b: $R_k \cap L_i = R_k L_i$

```
(theorem 3.10b
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K))
      (impl (and (and (and (and (and (and (semi-group G f)
                 (fulfil-min-cond-for-left-ideals G f))
                 (fulfil-min-cond-for-right-ideals G f))
                 (all (i J) (min-left-ideal (L i) G f)))
                 (all (X (class-of-min-left-ideals G f))
                        (ex (i J) (= X (L i)))))
                 (all (k K) (min-right-ideal (R k) G f)))
                 (all (X (class-of-min-right-ideals G f))
                        (ex (k K) (= X (R k)))))
             (all (i J) (all (k K)
                  (= (intersection (R k) (L i))
                     (complex-product G (R k) (L i) f)))
                       ))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.10c: The sets $R_k L_i$ are sub-semigroups of G.

```
(theorem 3.10c
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K))
      (impl (and (and (and (and (and (and (semi-group G f)
                 (fulfil-min-cond-for-left-ideals G f))
                 (fulfil-min-cond-for right-ideals G f))
                 (all (i J) (min-left-ideal (L i) G f)))
                 (all (X (class-of-min-left-ideals G f))
                        (ex (i J) (= X (L i)))))
                 (all (k K) (min-right-ideal (R k) G f)))
                 (all (X (class-of-min-right-ideals G f))
                        (ex (k K) (= X (R k)))))
            (all (i J) (all (k K)
                 (sub-semi-group
                     (complex-product G (R k) (L i) f) G f)
                         ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** attention! $\lambda \mapsto i, \quad \rho \mapsto k, \Lambda \mapsto J, P \mapsto K$
$G_{\rho\lambda} \mapsto$ **`G-ki G f J K L R k i`**

definition: $G_{ki} = R_k L_i$

```
(simple-definition G-ki
   (parameters (G (O A))
               (f (G G G))
               (J (O B))
               (K (O C))
               (L ((O A) J))
               (R ((O A) K))
               (k K)
               (i J))
   (sort (O A))
   (preconditions (and (and (and
                  (fulfil-min-cond-for-left-ideals G f))
                  (fulfil-min-cond-for right-ideals G f))
                  (min-left-ideal (L i) G f))
                  (min-right-ideal (R k) G f))
   (definition (complex-product G (R k) (L i) f))
   (consequences ((sub-group (G-ki G f J K L R k i) G f)
                  (status "conjectured")))
   (theory semi-groups))
```

*** theorem 3.10d1: $G_{ki}e_{km} = G_{km}$.

attention! $\lambda \mapsto i, \rho \mapsto k, \mu \mapsto m, \Lambda \mapsto J, P \mapsto K$
$G_{\rho\lambda} \mapsto$ `G-ki G f J K L R k i`
$G_{\rho\mu} \mapsto$ `G-ki G f J K L R k m`

```
(theorem 3.10d1
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K)) (all (i J) (all (k K)
            (all (m J)
            (impl (and (and (and (and
                      (fulfil-min-cond-for-left-ideals G f)
                      (fulfil-min-cond-for right-ideals G f))
                      (min-left-ideal (L i) G f))
                      (min-left-ideal (L m) G f))
                      (min-right-ideal (R k) G f))
                   (= (G-ki G f J K L R k m)
                      (right-complex-product G
                          (G-ki G f J K L R k i)
                          (one (G-ki G f J K L R k m) f)
                           f )))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.10d2: $e_{li}G_{ki} = G_{li}$

attention! $\lambda \mapsto i, \mu \mapsto m, \sigma \mapsto l, \Lambda \mapsto J, P \mapsto K$
$G_{\sigma\lambda} \mapsto$ `G-ki G f J K L R l i`

```
(theorem 3.10d2
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K)) (all (i J) (all (k K)
            (all (l K)
      (impl (and (and (and (and
                  (fulfil-min-cond-for-left-ideals G f)
                  (fulfil-min-cond-for right-ideals G f))
                  (min-left-ideal (L i) G f))
                  (min-right-ideal (R k) G f))
                  (min-right-ideal (R l) G f))
            (= (G-ki G f J K L R l i)
               (left-complex-product G
                     (one (G-ki G f J K L R l i) f)
                     (G-ki G f J K L R k i)
                      f)))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** φ_{kl} is a mapping: $G_{ki} \mapsto G_{li}$ defined by $x \rightarrow xe_{li}(x \in G_{ki})$

```
(implicit-definition phi-kl
   (parameters (G (O A))
               (f (G G G))
               (J (O B))
               (K (O C))
               (L ((O A) J))
               (R ((O A) K))
               (k K)
               (i J)
               (l K))
   (sort (G G))
   (preconditions (and (and (and (and
                  (fulfil-min-cond-for-left-ideals G f)
                  (fulfil-min-cond-for right-ideals G f))
                  (min-left-ideal (L i) G f))
                  (min-right-ideal (R k) G f))
                  (min-right-ideal (R l) G f)))
   (definition (all (x (G-ki G f J K L R l i))
               (= ((phi-kl G f J K L R k i l) x)
                  (f x (one (G-ki G f J K L R l i) f)))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.10d3: $\varphi_{kl} : G_{ki} \rightarrow G_{li}$ is a group-isomorphism.

```
(theorem 3.10d3
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K)) (all (i J) (all (k K)
            (all (l K)
      (impl (and (and (and (and
                 (fulfil-min-cond-for-left-ideals G f)
                 (fulfil-min-cond-for right-ideals G f))
                 (min-left-ideal (L i) G f))
                 (min-right-ideal (R k) G f))
                 (min-right-ideal (R l) G f))
            (group-isomorphism (G-ki G f J K L R k i) f
                               (G-ki G f J K L R l i) f
                     (phi-kl G f J K L R k i l))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** φ_{im} is a mapping: $G_{ki} \to G_{km}$ defined by $x \mapsto e_{km}x$ $(x \in G_{ki})$

```
(implicit-definition phi-im
   (parameters (G (O A))
               (f (G G G))
               (J (O B))
               (K (O C))
               (L ((O A) J))
               (R ((O A) K))
               (i J)
               (m J)
               (k K))
   (sort (G G))
   (preconditions (and (and (and (and
                     (fulfil-min-cond-for-left-ideals G f)
                     (fulfil-min-cond-for right-ideals G f))
                     (min-left-ideal (L i) G f))
                     (min-left-ideal (L m) G f))
                     (min-right-ideal (R k) G f)))
   (definition (all (x (G-ki G f J K L R k m))
               (= ((phi-im G f J K L R i m k) x)
                  (f (one (G-ki G f J K L R k m) f) x))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.10d4: $\varphi_{im} : G_{ki} \to G_{km}$ is a group-isomorphism.

```
(theorem 3.10d4
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K)) (all (i J) (all (m J)
            (all (k K)
      (impl (and (and (and (and
                         (fulfil-min-cond-for-left-ideals G f)
                         (fulfil-min-cond-for right-ideals G f))
                         (min-left-ideal (L i) G f))
                         (min-left-ideal (L m) G f))
                         (min-right-ideal (R k) G f))
            (group-isomorphism (G-ki G f J K L R k i) f
                               (G-ki G f J K L R m k) f
                         (phi-im G f J K L R i m k)))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.13 Satz: Aus der ersten Behauptung des letzten Satzes folgt nun

$L_\lambda = \bigcup_{\rho \in P} (R_\rho \cap L_\lambda) = \bigcup_{\rho \in P} G_{\rho\lambda}$

$R_\rho = \bigcup_{\lambda \in \Lambda} (R_\rho \cap L_\lambda) = \bigcup_{\lambda \in \Lambda} G_{\rho\lambda}$

Da außerdem für verschiedene λ und μ bzw. ρ und σ $L_\lambda \cap L_\mu = \emptyset$ bzw. $R_\rho \cap R_\sigma = \emptyset$ ist, gilt $G_{\rho\lambda} \cap G_{\mu\sigma} = \emptyset$.

*** theorem 3.13a: $L_i = \bigcup_{k \in K} (R_k \cap L_i)$

```
(theorem 3.13a
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K))
        (impl (and (and (and (and
                    (all (i J) (min-left-ideal (L i) G f))
                    (all (k K) (min-right-ideal (R k) G f)))
                    (fulfil-min-cond-for-left-ideals G f))
                    (fulfil-min-cond-for-right-ideals G f))
                    (all (X (class-of-min-right-ideals G f))
                         (ex (k K) (= X (R k)))))
              (all (i J) (=  (L i)
                             (fam-union K
                                 (intersection (R k) (L i))
                         )))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.13b: $R_k = \bigcup_{i \in J} (R_k \cap L_i)$

```
(theorem 3.13b
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O A))
            (all (K (O A)) (all (L ((O A) J))
            (all (R ((O A) K))
      (impl (and (and (and (and
                 (all (k K) (min-right-ideal (R k) G f))
                 (all (i J) (min-left-ideal (L i) G f)))
                 (fulfil-min-cond-for-right-ideals G f))
                 (fulfil-min-cond-for-left-ideals G f))
                 (all (X (class-of-min-left-ideals G f))
                      (ex (i J) (= X (L i)))))
            (all (k K) (= (R k)
                          (fam-union K
                               (intersection (R k) (L i))
                                              )))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.13c: $i_1 \neq i_2 \Rightarrow L_{i_1} \cap L_{i_2} = \emptyset$

```
(theorem 3.13c
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (L ((O A) J))
       (impl (and (and
                  (all (i J) (min-left-ideal (L i) G f))
               (fulfil-min-cond-for-left-ideals G f))
              (all (i1 J) (all (i2 J) (not= i1 i2))))
          (= empty-set
             (intersection (min-left-ideal (L i1) G f)
                           (min-left-ideal (L i2) G f)
               ))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.13d: $k_1 \neq k_2 \Rightarrow R_{k_1} \cap R_{k_2} = \emptyset$

```
(theorem 3.13d
   (theorem (all (G (O A)) (all (f (G G G)) (all (K (O B))
            (all (R ((O A) K))
        (impl (and (and
                 (all (k K) (min-right-ideal (R k) G f))
               (fulfil-min-cond-for-right-ideals G f))
                 (all (k1 K) (all (k2 K) (not= k1 k2))))
          (= empty-set
             (intersection (min-right-ideal (R k1) G f)
                           (min-right-ideal (R k2) G f)
                             ))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.13e: $i_1 \neq i_2 \wedge k_1 \neq k_2 \Rightarrow G_{k_1 i_1} \cap G_{k_2 i_2} = \emptyset$

```
(theorem 3.13e
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K))
      (impl (and (and (and (and (and
                 (all (i1 J) (all (i2 J) (not= i1 i2)))
                 (all (k1 K) (all (k2 K) (not= k1 k2))))
                 (all (k K) (min-right-ideal (R k) G f)))
                 (all (i J) (min-left-ideal (L i) G f)))
              (fulfil-min-cond-for-right-ideals G f))
              (fulfil-min-cond-for-right-ideals G f))
         (= empty-set
            (intersection (complex-product G (R k1) (L i1) f)
                          (complex-product G (R k2) (L i2) f)
                            ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.14 Satz: Zeichnet man ferner willkürlich je ein Element der Indexmengen Λ und P aus, etwa λ_0 und ρ_0, und kürzt ab $G := G_{\rho_0\lambda_0}$, so kann man u.a. aufgrund der vierten Behauptung des letzten Satzes schreiben:
$G_{\rho\lambda} = e_{\rho\lambda} G e_{\rho_0\lambda} = e_{\rho\lambda_0} G e_{\rho\lambda}$
womit $L_\lambda = \{e_{\rho\lambda} : \rho \in P\} \cdot G \cdot e_{\rho_0\lambda}$
$R_\rho = e_{\rho\lambda_0} \cdot G \cdot \{e_{\rho\lambda} : \lambda \in \Lambda\}$.

*** attention! $\lambda \mapsto i, \rho \mapsto k, \Lambda \mapsto J, P \mapsto K$
The abbreviation G of Deussen is not identical with the G on the right side!

$\{e_{ki} : k \in K\}$

```
(implicit-definition k-set-e-ki
   (parameters (G (O A))
               (f (G G G))
               (J (O B))
               (K (O C))
               (L ((O A) J))
               (R ((O A) K))
               (i J))
   (sort (O A))
   (preconditions (and (and (and
                  (fulfil-min-cond-for-left-ideals G f)
                  (fulfil-min-cond-for right-ideals G f))
                  (min-left-ideal (L i) G f))
                  (all (k K) (min-right-ideal (R k) G f))))
   (definition (all (x I)
               (equiv ((k-set-e-ki G f J K L R i) x)
                      (ex (k K)
                      (= x
                         (one (G-ki G f J K L R k i) f))))))
   (status "conjectured")
   (theory semi-groups))
```

*** $\{e_{ki} : i \in J\}$

```
(implicit-definition i-set-e-ki
   (parameters (G (O A))
               (f (G G G))
               (J (O B))
               (K (O C))
               (L ((O A) J))
               (R ((O A) K))
               (k K))
   (sort (O A))
   (preconditions (and (and (and
                  (fulfil-min-cond-for-left-ideals G f)
                  (fulfil-min-cond-for right-ideals G f))
                  (all (i J) (min-left-ideal (L i) G f)))
                  (min-right-ideal (R k) G f)))
   (definition (all (x I)
               (equiv ((i-set-e-ki) x)
                      (ex (i J)
                      (= x
                        (one (G-ki G f J K L R k i) f))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.14a: $L_i = \{e_{ki} : k \in K\} \cdot G_{k_0 i_0} \cdot e_{k_0 i}$

```
(theorem3.14a
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K)) (all (i J) (all (i0 J)
            (all (k0 K)
            (impl (and (and (and (and
                     (fulfil-min-cond-for-left-ideals G f)
                     (fulfil-min-cond-for-right-ideals G f))
                     (min-left-ideal (L i0) G f))
                     (min-right-ideal (R k0) G f))
                     (min-left-ideal (L i) G f))
                  (= (L i)
                     (complex-product G
                           (k-set-e-ki G f J K L R i)
                           (right-complex-product G
                              (G-ki G f J K L R k0 i0)
                              (one (G-ki G f J K L R k0 i) f)
                               f) f)))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.14b: $R_k = e_{ki_0} \cdot G_{k_0 i_0} \cdot \{e_{ki} : i \in J\}$

```
(theorem3.14b
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (K (O C)) (all (L ((O A) J))
            (all (R ((O A) K)) (all (k K) (all (i0 J)
            (all (k0 K)
            (impl (and (and (and (and
                     (fulfil-min-cond-for-left-ideals G f)
                     (fulfil-min-cond-for right-ideals G f))
                     (min-left-ideal (L i0) G f))
                     (min-right-ideal (R k0) G f))
                     (min-right-ideal (R k) G f))
                  (= (R k)
                     (complex-product G
                          (left-complex-product G
                              (one (G-ki G f J K L R k i0) f)
                              (G-ki G f J K L R k0 i0) f)
                          (i-set-e-ki G f J K L R k)
                           f)))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

3.15 Satz: Erfülle F die Minimalbedingung für Links- und Rechtsideale, so gilt:
Für beliebiges $f \in F$ und $\lambda \in \Lambda$ $[\rho \in P]$ ist $L_\lambda f = L_\mu$ $[fR_\rho = R_\sigma]$ für ein $\mu \in \Lambda$ $[\sigma \in P]$, und die Zuordnung $l \mapsto lf (l \in L_\lambda)$ $[r \mapsto fr (r \in R_\rho)]$ ist eine bijektive Abbildung von L_λ $[R_\rho]$ auf L_μ $[R_\sigma]$.
Es gilt daher für jedes $f \in F$ die Kürzungsregel:
$l_1 f = l_2 f \rightarrow l_1 = l_2 (l_1, l_2 \in L_\lambda)$
$[fr_1 = fr_2 \rightarrow r_1 = r_2 (r_1, r_2 \in R_\rho)]$

*** theorem 3.15a: Let the semigroup G fulfil the minimal condition for left and right ideals. Then $\forall x \in G\ \forall i \in J\ \exists m \in J : L_i \cdot x = L_m$.

```
(theorem 3.15a
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (L ((O A) J)) (all (x G) (all (i J)
            (impl (and (min-left-ideal (L i) G f)
                       (fulfil-min-cond-for-left-ideals G f))
                  (ex (m J)
                  (and (min-left-ideal (L m) G f)
                    (= (L m)
                       (right-complex-product G (L i) x f)
                        ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.15b: Let the semi-group G fulfil the minimal condition for left and right ideals. Then $\forall x \in G\ \forall k \in K\ \exists l \in K : x \cdot R_k = R_l$.

```
(theorem 3.15b
   (theorem (all (G (O A)) (all (f (G G G)) (all (K (O B))
            (all (R ((O A) K)) (all (x G) (all (k K)
            (impl (and (min-right-ideal (R k) G f)
                       (fulfil-min-cond-for-right-ideals G f))
                  (ex l K)
                  (and (min-right-ideal (R l) G f)
                    (= (R l)
                       (left-complex-product G x (R k) f)
                        ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** $L\text{-}ix : L_i \mapsto L_m$ with $l \mapsto l \cdot x\ (l \in L_i, x \in G)$

```
(implicit-definition L-ix
   (parameters (G (O A))
               (f (G G G))
               (J (O A))
               (L ((O A) J))
               (i J)
               (x G))
   (sort (G G))
   (preconditions (and (min-left-ideal (L i) G f)
                  (fulfil-min-condition-for-left-ideals G f)))
   (definition (all (l (L i))
               (= ((L-ix G f J L i x) l)
                  (f l x)))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.15c: The mapping $L\text{-}ix$ is a bijective mapping into a minimal left ideal L_m.

```
(theorem 3.15c
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O B))
            (all (L ((O A) J)) (all (x G) (all (i J)
            (all (l (L i))
            (impl (and (min-left-ideal (L i) G f)
                       (fulfil-min-cond-for-left-ideals G f))
                  (ex (m J)
                  (and (and (min-left-ideal (L m) G f)
                            ((L m) (f l x)))
                       (bijective (L i) (L m)
                                 (L-ix G f J L i x))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** $R\text{-}kx : R_k \longrightarrow R_l$ with $r \mapsto x \cdot r (r \in R_k, x \in G)$

```
(implicit-definition R-kx
   (parameters (G (O A))
               (f (G G G))
               (K (O B))
               (R ((O A) J))
               (k K)
               (x G))
   (sort (G G))
   (preconditions (and (and (min-right-ideal (R k) G f)
                       (min-right-ideal (R l) G f))
                  (fulfil-min-cond-for-right-ideals G f)))
   (definition (all (r (R k))
               (= ((R-kx G f K R k x) r)
                  (f X r))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.15d: The mapping $R\text{-}kx$ is a bijective mapping into a minimal right ideal R_l.

```
(theorem 3.15d
   (theorem (all (G (O A)) (all (f (G G G)) (all (K (O B))
            (all (R ((O A) J)) (all (x G) (all (k K)
            (all (r (R k))
            (impl (and (min-right-ideal (R k) G f)
                       (fulfil-min-cond-for-right-ideals G f))
                  (ex (l K)
                  (and (and (min-right-ideal (R l) G f)
                            ((R l) (f x r)))
                       (bijective (R k) (R l)
                                  (R-kx G f K R k x))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.15e: $l_1 \cdot x = l_2 \cdot x \;\Rightarrow\; l_1 = l_2 \; (l_1, l_2 \in L_i, x \in G)$

```
(theorem 3.15e
   (theorem (all (G (O A)) (all (F (G G G)) (all (J (O B))
            (all (L ((O A) J)) (all (i J) (all (l1 (L i))
            (all (l2 (L i)) (all (x G)
            (impl (and (and (and (min-left-ideal (L i) G f))
                       (fulfil-min-cond-for-left-ideals G f))
                       (= (f l1 x) (f l2 x)))
                  (= l1 l2))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 3.15f: $x \cdot r_1 = x \cdot r_2 \;\Rightarrow\; r_1 = r_2 \; (r_1, r_2 \in R_k, x \in G)$

```
(theorem 3.15f
   (theorem (all (G (O A)) (all (F (G G G)) (all (K (O B))
            (all (R ((O A) J)) (all (k K) (all (r1 (R k))
            (all (r2 (R k)) (all (x G)
            (impl (and (and (min-right-ideal (R k) G f)
                       (fulfil-min-cond-for-right-ideals G f))
                       (= (f x r1) (f x r2)))
                  (= r1 r2))))))))))
   (status "conjectured")
   (theory semi-groups))
```

Paragraph 4

4.1 Def.: Eine n-stellige *Relation* ρ auf der Menge S ist eine Teilmenge $\rho \subset S \times S \times \ldots \times S$ und man sagt: $s_1, s_2, \ldots, s_n$ stehen in der Relation ρ, wenn $(s_1, s_2, \ldots, s_n) \in \rho$. Ist $n = 2$, so schreibt man auch $s_1 \rho s_2$ für $(s_1, s_2) \in \rho$.

```
(simple-definition n-ary-rel
   (parameters (S (O A))
               (rho (O A))
               (n nat))
   (sort O)
   (definition (subset rho (n-same-cart-prod n S)))
   (theory relations))
```

*** It would be possible to define a binary relation similar to the n-ary relations:

```
(simple-definition bin-rel
      (parameters    (S (O A))
                     (rho (O A))
      (sort O)
      (definition    (subset rho (2-cart-prod S S)))
      (equivalences  (n-ary-relation S rho (suc 1))
                     (status "conjectured"))
      (theory        binary-relations))
```

but on the other hand it is not necessary to define binary relations at all since they can easily be considered as binary predicates on the set S.

Def.: Die Relation., die auf jedes Paar (s, s') zutrifft heißt *Allrelation* $\alpha = \{(s, s') : s, s' \in S\} = S \times S$.

```
(implicit-definition bin-all-rel
   (parameters (S (O A)))
   (sort (O A A))
   (definition (all (s S) (all (t S)
               ((bin-all-rel S) s t))))
   (status "conjectured")
   (theory binary-relations))
```

*** universal relation

Def.: Die *identische Relation* oder die *Gleichheitsrelation* trifft nur auf die Paare (s,s) zu:
$\imath = \{(s,s) : s \in S\}$.

*** identical relation

```
(implicit-definition bin-id-rel
   (parameters (S (O A)))
   (sort (O S S))
   (definition (all (x S) (all (y S)
               (equiv ((bin-id-rel S) x y)
                      (= x y)))))
   (status "conjectured")
   (theory binary-relations))
```

4.2 Def.: Seien ρ, σ zwei Relationen. Die Relation $\rho \circ \sigma$, definiert durch
$(s,t) \in \rho \circ \sigma :\Leftrightarrow$
$\exists r \in S : (s,r) \in \rho \wedge (r,t) \in \sigma$ heißt die *Komposition* von ρ und σ.

*** the composition of two relations

```
(implicit-definition comp-rel
   (parameters (S (O A))
               (rho (O S S))
               (sigma (O S S)))
   (sort (O S S))
   (definition (all (s S) (all (t S)
               (equiv ((comp-rel S rho sigma) s t)
                      (ex (r S) (and (rho (s r))
                                     (sigma (r t))))))))
   (status "conjectured")
   (theory binary-relations))
```

4.3 Satz.: Die Komposition ist eine assoziative Verknüpfung:
$(\rho \circ \sigma) \circ \tau = \rho \circ (\sigma \circ \tau)$.

*** theorem 4.3: associativity of the composition of relations

```
(theorem 4.3
   (theorem (all (S (O A)) (all (rho (O S S))
            (all (sigma (O S S)) (all (tau (O S S))
            (= (comp-rel S (comp-rel S rho sigma) tau)
               (comp-rel S rho (comp-rel sigma tau))))))))
   (status "conjectured")
   (theory binary-relations))
```

Def.: Potenzen einer Relation ρ werden induktiv definiert durch

$\rho^1 := \rho, \qquad \rho^{n+1} := \rho^n \circ \rho.$

*** ρ^n

```
(inductive-definion power-rel
   (parameters (S (O A))
               (rho (O S S))
               (n nat))
   (sort (O S S))
   (base (= (power-rel S rho 1) rho))
   (step (= (power-rel S rho (suc n))
            (comp-rel S (power-rel S rho n) rho)))
   (theory binary-relations))
```

4.4 Def.: Ist ρ eine Relation, so ist die zu ρ *konverse* Relation ρ^c definiert durch

$(s,t) \in \rho^c \quad \Leftrightarrow \quad (t,s) \in \rho.$

*** converse relation

```
(implicit-definition converse-rel
   (parameters (S (O A))
               (rho (O S S)))
   (sort O S S)
   (definition (all (s S) (all (t S)
               (equiv ((converse-rel S rho) s t)
                      (rho t s)))))
   (status "conjectured")
   (theory binary-relations))
```

4.5 Def.: Ist ρ eine Relation, so heißen die Mengen

$\rho s \quad := \{t \in S : t\rho s\}$ *Vorbereich* von s (unter ρ),

$t\rho \quad := \{s \in S : t\rho s\}$ *Nachbereich* von t (unter ρ),

$V(\rho) := \bigcup_{s\in S} \rho s$ *Vorbereich* von ρ,

$N(\rho) := \bigcup_{t\in S} t\rho$ *Nachbereich* von ρ.

*** left domain of s

```
(implicit-definition left-domain-of-s
   (parameters (S (O A))
               (rho (O S S))
               (s S))
   (sort (O S))
   (definition (all (t S)
               (equiv ((left-domain-of-s S rho s) t)
                      (rho t s))))
   (status "conjectured")
   (theory binary-relations))
```

*** right domain of t

```
(implicit-definition right-domain-of-t
   (parameters (S (O A))
               (rho (O S S))
               (t S))
   (sort (O S))
   (definition (all (s S)
               (equiv ((right-domain-of-t S rho t) s)
                      (rho t s))))
   (status "conjectured")
   (theory binary-relations))
```

*** left domain of ρ

```
(implicit-definition left-domain-of
   (parameters (S (O A))
               (rho (O S S)))
   (sort (O S))
   (definition (all (x S)
               (equiv ((left-domain-of S rho) x)
                      (ex (s S)
                      ((left-domain-of-s S rho s) x)))))
   (status "conjectured")
   (theory binary-relations))
```

*** right domain of ρ

```
(implicit-definition right-domain-of
   (parameters (S (O A))
               (rho (O S S)))
   (sort (O S))
   (definition (all (x S)
               (equiv ((right-domain-of S rho) x)
                      (ex (t S)
                      ((right-domain-of-t S rho t) x)))))
   (status "conjectured")
   (theory binary-relations))
```

4.6 Def.: Eine Relation ρ heißt
reflexiv, wenn $\imath \subset \rho$;
symmetrisch, wenn $\rho \subset \rho^c$;
transitiv, wenn $\rho \circ \rho \subset \rho$;
antisymmetrisch, wenn $\rho \cap \rho^c \subset \imath$ gilt.

*** reflexive

```
(simple-definition reflexive
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (subset (bin-id-rel S) rho))
   (equivalences ((all (s S) (rho s s))
                 (status "conjectured")))
   (theory binary-relations))
```

*** symmetric

```
(simple-definition symmetric
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (subset rho (converse-rel S rho)))
   (equivalences ((all (s S) (all (t S)
                 (impl (rho s t)
                       (rho t s))))
                 (status "conjectured")))
   (theory binary-relations))
```

*** transitive

```
(simple-definition transitive
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (subset (comp-rel S rho rho) rho))
   (equivalences ((all (r S) (all (s S) (all (t S)
                 (impl (and (rho r s) (rho s t))
                       (rho r t)))))
                  (status "conjectured")))
   (theory binary-relations))
```

*** antisymmetric

```
(simple-definition antisymmetric
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (subset (intersection rho (converse-rel S rho))
                       (bin-id-rel S)))
   (theory binary-relations))
```

Def.: Eine reflexive, transitive und symmetrische Relation heißt *Äquivalenzrelation*.

*** equivalence relation

```
(simple-definition equiv-rel
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (and (and (reflexive S rho)
                         (symmetric S rho))
                    (transitive S rho)))
   (theory binary-relations))
```

Def.: Eine reflexive, transitive und antisymmetrische Relation heißt *Ordnungsrelation*.

*** order relation

```
(simple-definition order-rel
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (and (and (reflexive S rho)
                         (antisymmetric S rho))
                    (transitive S rho)))
   (theory binary-relations))
```

Satz: Für die Vorbereiche einer Äquivalenzrelation ρ gilt also entweder $\rho s \cap \rho t = \emptyset$ oder $\rho s = \rho t$.

*** theorem 4.6a: For the left domains of an equivalence relation ρ there is neither $\rho s \cap \rho t = \emptyset$ nor $\rho s = \rho t$.

```
(theorem 4.6a
   (theorem (all (S (O A)) (all (rho (O S S)) (all (s S)
            (all (t S)
        (impl (and (equiv-rel S rho)
                   ((left-domain-of-s S rho s) t))
              (or (= (left-domain-of-s S rho s)
                     (left-domain-of-s S rho t))
                  (= empty-set
                     (intersection (left-domain-of-s S rho s)
                         (left-domain-of-s S rho t))))))))))
   (status "conjectured")
   (theory binary-relations))
```

Def.: Jede Äquivalenzrelation bestimmt durch ihre Vorbereiche (oder Nachbereiche), die man auch *Äquivalenzklassen, Klassen* oder *Restklassen* nennt, eine Partition oder Zerlegung von S in eben diese Klassen. Die Anzahl dieser Klassen heißt *Index* von ρ ($ind(\rho)$).

*** equivalence class

```
(simple-def equiv-class
   (parameters (S (O A))
               (rho (O S S))
               (s S))
   (sort (O S))
   (preconditions (equiv-rel S rho))
   (definition (left-domain-of-s S rho s))
   (equivalences ((right-domain-of-s S rho s)
                  (status "conjectured")))
   (theory binary-relations))
```

*** set of all equivalence classes

```
(implicit-definition set-of-all-equiv-classes
   (parameters (S (O A))
               (rho (O S S)))
   (sort (O (O A)))
   (definition (all (X (O A))
               (equiv ((set-of-all-equiv-classes S rho) X)
                      (ex (s S) (= (equiv-class S rho s) X)))))
   (status "conjectured")
   (theory binary-relations))
```

*** the index of an equivalence class

```
(simple-definition index-of-equiv-rel
   (parameters (S (O A))
               (rho (O S S)))
   (sort I)
   (definition (card (set-of-all-equiv-classes S rho)))
   (theory binary-relations))
```

Satz: Für beliebiges ρ ist $\rho \cup \imath$ reflexiv.

*** theorem 4.6c: For each relation ρ is $\rho \cup \imath$ reflexive.

```
(theorem 4.6c
   (theorem (all (S (O A)) (all (rho (O S S))
            (reflexive S (union rho (bin-id-rel S))))))
   (status "conjectured")
   (theory binary-relations))
```

Satz: Für beliebiges ρ ist $\rho \cup \rho^c$ symmetrisch.

*** theorem 4.6d: For each relation ρ is $\rho \cup \rho^c$ symmetric.

```
(theorem 4.6d
   (theorem (all (S (O A)) (all (rho (O S S))
            (symmetric S (union rho (converse-rel S rho))))))
   (status "conjectured")
   (theory binary-relations))
```

Def.: die *transitive Hülle* $\rho^t := \bigcup_{n \in \mathbb{N}} \rho^n$

*** transitive closure

```
(implicit-definition transitive-closure
   (parameters (S (O A))
               (rho (O S S)))
   (sort (O S S))
   (definition (all (s S) (all (t S)
               (equiv ((transitive-closure S rho) s t)
                      (ex (n nat)
                          ((power-rel S rho n) s t))))))
   (status "conjectured")
   (theory binary-relations))
```

4.7 Def.: Die Relation $\hat{\rho} := (\rho \cup \rho^c \cup \imath)^t$ heißt die von ρ *erzeugte Äquivalenzrelation.*

*** $\hat{\rho}$

```
(simple-definition generated-equiv-rel
   (parameters (S (O A))
               (rho (O S S)))
   (sort O)
   (definition (transitive-closure S
                     (union rho (union
                                 (converse-rel S rho)
                                       (bin-id-rel S)))))
   (theory binary-relations))
```

*** theorem 4.7: $\hat{\rho}$ is an equivalence relation

```
(theorem 4.7
   (theorem (all (S (O A)) (all (rho (O S S))
            (equiv-rel S (generated-equiv-rel S rho)))))
   (status "conjectured")
   (theory binary-relations))
```

4.8 Satz: Seien ρ, σ zwei Äquivalenzrelationen, so ist

1. $\rho \cap \sigma$ Äquivalenzrelation
2. $\rho \sqcup \sigma := (\rho \cup \sigma)^t$ die kleinste Äquivalenzrelation,die ρ und σ enthält.

*** theorem 4.8a: ρ, σ equivalence relations $\Rightarrow$ $\rho \cap \sigma$ is an equivalence relation.

```
(theorem 4.8a
   (theorem (all (S (O A)) (all (rho (O S S))
            (all (sigma (O S S))
            (impl (and (equiv-rel S rho)
                       (equiv-rel S sigma))
                  (equiv-rel S (intersection rho sigma)))))))
   (status "conjectured")
   (theory binary-relations))
```

*** $\rho \sqcup \sigma$

```
(simple-definition sqcup-rel
   (parameters (S (O A))
               (rho (O S S))
               (sigma (O S S)))
   (sort (O S S))
   (definition (transitive-closure S (union rho sigma)))
   (theory binary-relations))
```

*** theorem 4.8b: $\rho \sqcup \sigma$ is an equivalence relation.

```
(theorem 4.8b
   (theorem (all (S (O A)) (all (rho (O S S))
            (all (sigma (O S S))
            (impl (and (equiv-rel S rho)
                       (equiv-rel S sigma))
                  (equiv-rel S (sqcup-rel S rho sigma)))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.8c: $\rho \subset \rho \sqcup \sigma$ and $\sigma \subset \rho \sqcup \sigma$.

```
(theorem 4.8c
   (theorem (all (S (O A)) (all (rho (O S S))
            (all (sigma (O S S))
            (impl (and (equiv-rel S rho)
                       (equiv-rel S sigma))
                  (and (subset rho (sqcup-rel S rho sigma))
                       (subset sigma (sqcup-rel S rho sigma))
                                                        ))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.8d: $\rho \sqcup \sigma$ is the smallest equivalence relation containing the equivalence relations $\rho \cup \sigma$.

```
(theorem 4.8d
   (theorem (all (S (O A)) (all (rho (O S S))
            (all (sigma (O S S))
            (all (mu (O S S))
        (impl (and (and (and (and (equiv-rel S rho)
                                  (equiv-rel S sigma))
                             (equiv-rel S mu))
                        (subset rho mu))
                   (subset sigma mu))
              (subset (sqcup-rel S rho sigma) mu)))))))
   (status "conjectured")
   (theory binary-relations))
```

4.9 Def.: Ist $\phi : S \to U$ eine Abbildung, so induziert ϕ in natürlicher Weise eine Äquivalenzrelation auf S gemäß:
$s\rho t \quad :\Leftrightarrow \quad \phi(s) = \phi(t).$

*** the induced equivalence relation

```
(implicit-definition induced-equiv-rel
   (parameters (S (O A))
               (U (O B))
               (phi (U S)))
   (sort (O S S))
   (definition (all (s S) (all (t S) (equiv
               ((induced-equiv-rel S U phi) s t)
                      (= (phi s) (phi t))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.9: the induced equivalence relation is indeed an equivalence relation.

```
(theorem 4.9
   (theorem (all (S (O A)) (all (U (O B))
            (all (phi (U S))
            (equiv-rel S (induced-equiv-rel S U phi))))))
   (status "conjectured")
   (theory binary-relations))
```

Def.: Als die *Quotientenmenge von S nach ρ* bezeichnet man die Menge $S/\rho := \{\rho s : s \in S\}$ der Klassen von ρ in S.

quotient set

```
(implicit-definition quotient-set
   (parameters (S (O A))
               (rho (O S S)))
   (sort (O (O A)))
   (preconditions (equiv-rel S rho))
   (definition (all (X (O A))
               (equiv ((quotient-set S rho) X)
                      (ex (s S)
                      (= X (left-domain-of-s S rho s))))))
   (status "conjectured")
   (theory binary-relations))
```

4.10 Satz: Sei $\varphi : S \to U$ eine surjektive Abbildung und ρ die von φ in S induzierte Äquivalenzrelation gemäß (4.9). Seien die Abbildungen

$\natural : S \to S/\rho$ und $\Phi : S/\rho \to U$

definiert durch

$\natural(s) := \rho s$ und $\Phi(\rho s) := \varphi(s)$

so gilt: $\natural$ (die *kanonische Abbildung*) ist surjektiv und φ ist bijektiv, ferner ist das Diagramm

$$\begin{tikzcd} S \arrow[rr, "\natural"] \arrow[dr, "\varphi"'] & & S/\rho \arrow[dl, "\Phi"] \\ & U & \end{tikzcd}$$

kommutativ, d.h. $\Phi\natural = \varphi$.

*** the canonical mapping $\natural$

```
(implicit-definition canonical-mapping
   (parameters (S (O A))
               (rho (O S S)))
   (sort ((O (O A)) (O A)))
   (definition (all (s S)
               (= ((canonical-mapping S rho) s)
                  (left-domain-of-s S rho s))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.10a: the canonical mapping $\natural : S \to S/\rho$ is surjective.

```
(theorem 4.10a
   (theorem (all (S (O A)) (all (U (O B)) (all (phi (U S))
            (impl (surjective S U phi)
                  (surjective S (quotient-set S
                                   (induced-equiv-rel S U phi))
                           (canonical-mapping S
                             (induced-equiv-rel S U phi))))))))
   (status "conjectured")
   (theory binary-relations))
```

*** the induced-mapping $\Phi : S/\rho \to U$ defined by $\Phi(\rho s) := \varphi(s)$

```
(implicit-definition induced-mapping
   (parameters (S (O A))
               (U (O B))
               (phi (U S)))
   (sort ((O A) (O (O A))))
   (definition (all (s S)
               (= ((induced-mapping S U phi)
                        (left-domain-of-s S
                           (induced-equiv-rel S U phi) s))
                  (phi s))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.10b: $\Phi : S/\rho \to U$ is bijective.

```
(theorem 4.10b
   (theorem (all (S (O A)) (all (U (O B)) (all (phi (U S))
            (impl (surjective S U phi)
                  (bijective U (quotient-set S
                                  (induced-equiv-rel S U phi))
                        (induced-mapping S U phi)))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.10c: $\Phi\natural = \varphi$.

```
(theorem 4.10c
   (theorem (all (S (O A)) (all (U (O B)) (all (phi (U S))
            (= phi
               (comp (induced-mapping S U phi)
                     (canonical mapping S
                        (induced-equiv-rel S U phi))))))))
   (status "conjectured")
   (theory binary-relations))
```

4.11 Satz: Seien $\varphi_1 : S \to U$, $\varphi_2 : S \to V$ zwei Abbildungen und ρ_1 bzw. ρ_2 die induzierten Äquivalenzrelationen.
1. Gibt es eine Abbildung $\Phi : U \to V$ mit $\Phi\varphi_1 = \varphi_2$, so ist $\rho_1 \subset \rho_2$.
2. Sei $\rho_1 \subset \rho_2$ und φ_1 surjektiv, so gibt es eine Abbildung $\Phi : U \to V$ mit $\Phi\varphi_1 = \varphi_2$, und diese Abbildung ist eindeutig bestimmt. Ist überdies φ_2 surjektiv, so ist auch Φ surjektiv.

*** theorem 4.11a: If there is a mapping $\Phi : U \mapsto V$ with $\Phi\varphi_1 = \varphi_2$ then $\rho_1 \subset \rho_2$.

```
(theorem 4.11a
   (theorem (all (S (O A)) (all (U (O B)) (all (V (O C))
            (all (phi1 (U S)) (all (phi2 (V S))
            (impl (ex (Phi (V U))
                        (= (comp Phi phi1) phi2))
                  (subset (induced-equiv-rel S phi1)
                          (induced-equiv-rel S phi2))))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.11b: If $\rho_1 \subset \rho_2$ and φ_1 is surjective, then there exists a mapping $\Phi : U \to V : \Phi\varphi_1 = \varphi_2$.

```
(theorem 4.11b
   (theorem (all (S (O A)) (all (U (O B)) (all (V (O C))
            (all (phi1 (U S)) (all (phi2 (V S))
            (impl (and (subset (induced-equiv-rel S phi1)
                               (induced-equiv-rel S phi2))
                        (surjective S U phi1))
                  (ex (Phi (V U))
                  (= (comp Phi phi1) phi2))))))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.11c: The mapping Φ of 4.11b is unique.

```
(theorem 4.11c
   (theorem (all (S (O A)) (all (U (O B)) (all (V (O C))
            (all (phi1 (U S)) (all (phi2 (V S))
            (impl (and (and (and
                        (subset (induced-equiv-rel S phi1)
                                (induced-equiv-rel S phi2))
                        (surjective S U phi1))
                        (ex (Phi1 (V U))
                            (= (comp Phi1 phi1) phi2))
                        (ex (Phi2 (V U))
                            (= (comp Phi2 phi1) phi2))
                  (= Phi1 Phi2))))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 4.11d: Is φ_2 surjective in addition then Φ is surjective too.

```
(theorem 4.11d
   (theorem (all (S (O A)) (all (U (O )) (all (V (O ))
            (all (phi1 (U S)) (all (phi2 (V S))
            (impl (and (and (and
                        (subset (induced-equiv-rel S phi1)
                                (induced-equiv-rel S phi2))
                        (surjective S U phi1))
                        (surjective S V phi2))
                        (ex (Phi (V U))
                        (= (comp Phi phi1) phi2))
                  (surjective U V Phi))))))))
   (status "conjectured")
   (theory binary-relations))
```

4.12 Satz: Sind nun ρ_1, ρ_2 irgendwelche Äquivalenzrelationen und setzt man $U = S/\rho_1, V = S/\rho_2, \varphi_1 = \natural_1, \varphi_2 = \natural_2$, so erhält man als Korollar zum letzten Satz:
Es gibt eine surjektive Abbildung $\Phi : S/\rho_1 \to S/\rho_2$ mit $\Phi\natural_1 = \natural_2$ genau dann, wenn $\rho_1 \subset \rho_2$.

*** theorem 4.12: Let ρ_1, ρ_2 be equivalence relations and $U = S/\rho_1, V = S/\rho_2, \varphi_1 = \natural_1, \varphi_2 = \natural_2$, then:
there is a surjective mapping $\Phi : S/\rho_1 \to S/\rho_2$ with $\Phi\natural_1 = \natural_2$ iff $\rho_1 \subset \rho_2$.

```
(theorem 4.12
   (theorem (all (S (O A)) (all (rho1 (O S S))
            (all (rho2 (O A))
      (impl (and (equiv-rel S rho1)
                 (equiv-rel S rho2))
            (equiv (ex (Phi ((quotient-set S rho2)
                             (quotient-set S rho1)))
                        (and (surjective (quotient-set S rho1)
                                    (quotient-set S rho2) Phi)
                             (= (comp Phi
                                      (canonical-mapping S rho1))
                                (canonical-mapping S rho2))))
                   (subset rho1 rho2)))))))
   (status "conjectured")
   (theory binary-relations))
```

Paragraph 5

5.1 Def.: Sei ρ eine Äquivalenzrelation auf der Halbgruppe $F.\rho$ heißt
Linkskongruenz, wenn für beliebige $g, f_1, f_2 \in F$ aus $f_1 \rho f_2$ folgt $(gf_1)\rho(gf_2)$,
Rechtskongruenz, wenn für beliebige $g, f_1, f_2 \in F$ aus $f_1 \rho f_2$ folgt $(f_1 g)\rho(f_2 g)$, und
Kongruenz, wenn ρ zugleich Links- und Rechtskongruenz ist.

*** left congruence

```
(simple-definition left-congruence
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G)))
   (sort O)
   (superconcepts (semi-group G f)
                  (equiv-rel G rho))
   (definition (all (g G) (all (g1 G) (all (g2 G)
               (impl (rho g1 g2)
                     (rho (f g g1) (f g g2)))))))
   (theory semi-groups))
```

*** right congruence

```
(simple-definition right-congruence
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G)))
   (sort O)
   (superconcepts (semi-group G f)
                  (equiv-rel G rho))
   (definition (all (g G) (all (g1 G) (all (g2 G)
               (impl (rho g1 g2)
                     (rho (f g1 g) (f g2 g)))))))
   (theory semi-groups))
```

*** congruence

```
(simple-definition congruence
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G)))
   (sort O)
   (superconcepts (semi-group G f)
                  (equiv-rel G rho)
                  (left-congruence G f rho)
                  (right-congruence G f rho))
   (definition T)
   (theory semi-groups))
```

Danach induziert jedes $\varphi \in Hom(F, H)$ eine Kongruenz in F.

*** theorem 5.1: Each semigroup-homomorphism $\varphi : F \to H$ induces a conguence in F.

```
(theorem 5.1
   (theorem (all (F (O A)) (all (f (F F F)) (all (H (O A))
            (all (h (H H H)) (all (phi (H F))
            (impl (semi-group-homorphism F f H h phi)
                  (congruence F f
                     (induced-equiv-rel F H phi)))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.2 Satz: Sei ρ eine Äquivalenzrelation auf F, dann sind äquivalent:

1. ρ ist Linkskongruenz [Rechtskongruenz]
2. Für jedes $f \in F$ und $h \in F$ gilt
   $h \cdot (\rho f) \subset \rho(hf) \qquad [(\rho f)h \subset \rho(fh)]$.

*** theorem 5.2a: If ρ is an equivalence relation on G then:
ρ is a left congruence $\Leftrightarrow$ $\forall x \in G$ and $\forall y \in G$: $y \cdot (\rho\ x) \subset \ \rho\ (yx)$.

```
(theorem 5.2a
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
        (impl (and (semi-group G f) (equiv-rel G rho))
              (equiv (left-congruence G f rho)
                     (all (x G) (all (y G)
                      (subset (left-complex-product G y
                                    (left-domain-of-s G rho x) f)
                              (left-domain-of-s G rho (f y x))
     )))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.2b: If ρ is an equivalence relation on G then:
ρ is a right congruence $\Leftrightarrow$ $\forall x \in G$ and $\forall y \in G$:
$(\rho\ x)\ y \subset\ \rho\ (xy)$.

```
(theorem 5.2b
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
        (impl (and (semi-group G f) (equiv-rel G rho))
              (equiv (right-congruence G f rho)
                     (all (x G) (all (y G)
                     (subset (right-complex-product G
                               (right-domain-of-t G rho x) y f)
                              (right-domain-of-t G rho (f x y))
                                 )))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.3 Satz: Seien ρ, σ Links- oder Rechtskongruenzen oder Kongruenzen, so ist
1. $\rho \cap \sigma$ Links- bzw. Rechtskongruenz bzw. Kongruenz
2. $\rho \sqcup \sigma := (\rho \cup \sigma)^t$ die kleinste Links- bzw. Rechtskongruenz bzw. Kongruenz, die σ und ρ enthält.

*** theorem 5.3a1: ρ, σ left congruences $\Rightarrow$ $\rho \cap \sigma$ is a left congruence.

```
(theorem 5.3a1
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G))
            (impl (and (left-congruence G f rho)
                       (left-congruence G f sigma))
                  (left-congruence G f
                        (intersection rho sigma))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3a2: ρ, σ right congruences $\Rightarrow$ $\rho \cap \sigma$ is a right congruence.

```
(theorem 5.3a2
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G))
            (impl (and (right-congruence G f rho)
                       (right-congruence G f sigma))
                  (right-congruence G f
                        (intersection rho sigma))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3a3: ρ, σ congruences $\Rightarrow$ $\rho \cap \sigma$ is a congruence.

```
(theorem 5.3a3
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G))
            (impl (and (congruence G f rho)
                       (congruence G f sigma))
                  (congruence G f
                       (intersection rho sigma)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3b1: ρ, σ left congruences $\Rightarrow$ $\rho \sqcup \sigma$ is a left congruence.

```
(theorem 5.3b1
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G))
            (impl (and (left-congruence G f rho)
                       (left-congruence G f sigma))
                  (left-congruence G f
                       (sqcup-rel G rho sigma)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3b2: ρ, σ right congruences $\Rightarrow$ $\rho \sqcup \sigma$ is a right congruence.

```
(theorem 5.3b2
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G))
            (impl (and (right-congruence G f rho)
                       (right-congruence G f sigma))
                  (right-congruence G f
                       (sqcup-rel G rho sigma)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3b3: ρ, σ congruences $\Rightarrow$ $\rho \sqcup \sigma$ is a congruence.

```
(theorem 5.3b3
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G))
            (impl (and (congruence G f rho)
                       (congruence G f sigma))
                  (congruence G f
                       (sqcup-rel G rho sigma))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3c1: ρ, σ left congruences $\Rightarrow$ $\rho \sqcup \sigma$ is the smallest left congruence containing ρ and σ.

```
(theorem 5.3c1
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G)) (all (mu (O G G))
        (impl (and (and (and (and
                   (left-congruence G f rho)
                   (left-congruence G f sigma))
                   (left-congruence G f mu))
                   (subset rho mu))
                   (subset sigma mu))
              (subset (sqcup-rel G rho sigma) mu))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3c2: ρ, σ right congruences $\Rightarrow$ $\rho \sqcup \sigma$ is the smallest right congruence containing ρ and σ.

```
(theorem 5.3c2
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G)) (all (mu (O G G))
        (impl (and (and (and (and
                   (right-congruence G f rho)
                   (right-congruence G f sigma))
                   (right-congruence G f mu))
                   (subset rho mu))
                   (subset sigma mu))
              (subset (sqcup-rel G rho sigma) mu))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.3c3: ρ, σ congruences $\Rightarrow$ $\rho \sqcup \sigma$ is the smallest congruence containing ρ and σ.

```
(theorem 5.3c3
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (sigma (O G G)) (all (mu (O G G))
        (impl (and (and (and (and
                   (congruence G f rho)
                   (congruence G f sigma))
                   (congruence G f mu))
                   (subset rho mu))
                   (subset sigma mu))
             (subset (sqcup-rel G rho sigma) mu))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.4 Def.: Ist ρ eine Kongruenz, so kann man die Quotientenmenge F/ρ zu einer Halbgruppe mit der Verknüpfung $\circ$ machen, wenn man definiert

$\rho f \circ \rho g \quad := \quad \rho(fg)$

*** composition inside of the quotient set G/ρ.

```
(implicit-definition comp-cong-quot-set
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G))
   (sort ((O (O A)) (O (O A)) (O (O A))))
   (preconditions (and (semi-group G f)
                       (congruence G f rho)))
   (definition (all (x G) (all (y G)
               (= ((comp-cong-quot-set G f rho)
                          (left-domain-of-s G rho x)
                          (left-domain-of-s G rho y))
                  (left-domain-of-s G rho (f x y))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.4: If ρ is a congruence on G with
$\rho x_1 = \rho x_2$ and $\rho y_1 = \rho y_2 \quad \Rightarrow \rho\ (x_1 y_1)\ =\ \rho\ (x_2 y_2)$.
By this theorem the well-definedness of the implicit definition is given.

```
(theorem 5.4
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (x1 G) (all (x2 G) (all (y1 G) (all (y2 G)
        (impl (and (and (congruence G f rho)
                   (= (left-domain-of-s G rho x1)
                      (left-domain-of-s G rho x2)))
                   (= (left-domain-of-s G rho y1)
                      (left-domain-of-s G rho y2)))
              (= (left-domain-of-s G rho (f x1 y1))
                 (left-domain-of-s G rho (f x2 y2))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.5 Def.: Sei ρ eine Kongruenz auf der Halbgruppe F. Die Quotientenmenge F/ρ versehen mit der Verknüpfung gemäß (5.4) heißt *Quotientenhalbgruppe von F nach ρ.*

*** theorem 5.5: The quotient set G/ρ together with the composition forms a semigroup.

```
(theorem 5.5
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (impl (and (semi-group G f)
                       (congruence G f rho))
                  (semi-group (quotient-set G rho)
                              (comp-cong-quot-set G f rho))
                                                       )))))
   (status "conjectured")
   (theory semi-groups))
```

*** quotient semigroup
This problem cannot be easily encoded in Post because of the limitations of the language.

```
(simple-definition quot-semi-group
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G)))
   (sort (O (O A)))
   (preconditions (and (semi-group G f)
                       (congruence G f rho)))
   (definition (*pair* (quotient-set G rho)
                       (comp-cong-quot-set G f rho)))
   (theory semi-groups))
```

5.6 Satz: Sei $\varphi : F \to H$ ein Epimorphismus und ρ die von φ induzierte Kongruenz, so gilt für die Abbildungen

$\natural : F \to F/\rho$	und	$\Phi : F/\rho \to H$ mit
$\natural(f) := \rho f$	und	$\Phi(\rho f) := \varphi(f)$

$\natural$ ist ein Epimorphismus und Φ ein Isomorphismus auf H, ferner ist das Diagramm

$$\begin{tikzcd} F \arrow[rr, "\natural"] \arrow[dr, "\varphi"'] & & F/\rho \arrow[dl, "\Phi"] \\ & H & \end{tikzcd}$$

kommutativ: $\Phi\natural = \varphi$.

*** theorem 5.6a: the mapping $\natural : G \to G/\rho$ is a semigroup-epimorphism.

```
(theorem 5.6a
   (theorem (all (G (O A)) (all (g (G G G)) (all (H (O A))
            (all (h (H H H)) (all (phi (H G))
            (impl (semi-group-epimorphism G g H h phi)
                  (semi-group-epimorphism G g
                              (quotient-set G
                                (induced-equiv-rel G H phi))
                              (comp-cong-quot-set G g
                                (induced-equiv-rel G H phi))
                              (canonical-mapping G
                                (induced-equiv-
                                      rel G H phi))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.6b: The mapping $\Phi : G/\rho \to H$ is a semigroup-isomorphism.

```
(theorem 5.6b
   (theorem (all (G (O A)) (all (g (G G G)) (all (H (O A))
            (all (h (H H H)) (all (phi (H G))
            (impl (semi-group-epimorphism G g H h phi)
                  (semi-group-isomorphism (quotient-set G
                         (induced-equiv-rel G H phi))
                         (comp-cong-quot-set G g
                             (induced-equiv-rel G H phi))
                         H h (induced-mapping
                                  G (induced-equiv-rel G H phi)
                                  H phi)))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.6c: $\Phi\natural = \varphi$.

```
(theorem 5.6c
   (theorem (all (G (O A)) (all (g (G G G)) (all (H (O A))
            (all (h (H H H)) (all (phi (H G))
            (impl (semi-group-epimorphism G g H h phi)
                  (= phi
                     (comp (induced-mapping G
                                 (induced-equiv-rel G H phi)
                                                     H phi)
                           (canonical-mapping G
                                 (induced-equiv-rel G H phi))
                                                    ))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.7 Satz: Seien $\varphi_1 : F \to H_1$, $\varphi_2 : F \to H_2$ zwei Homomorphismen in die Halbgruppe H_1 bzw. H_2 und ρ_1 bzw. ρ_2 die induzierten Kongruenzen.
1. Gibt es einen Homomorphismus $\Phi : H_1 \to H_2$ mit $\Phi\varphi_1 = \varphi_2$, so ist $\rho_1 \subset \rho_2$.
2. Sei $\rho_1 \subset \rho_2$ und φ_1 surjektiv, so gibt es genau einen Homomorphismus $\Phi : H_1 \to H_2$ mit $\Phi\varphi_1 = \varphi_2$, und dieser Homomorphismus ist eindeutig bestimmt. Ist überdies φ_2 surjektiv, so ist es auch Φ.

*** theorem 5.7a: Let φ_1 and φ_2 be homomorphisms into the semigroups H_1 and H_2 and ρ_1 and ρ_2 the corresponding induced congruences. If there exists a homomorphism $\Phi : H_1 \to H_2$ with $\Phi\varphi_1 = \varphi_2$ then $\rho_1 \subset \rho_2$.

```
(theorem 5.7a
   (theorem (all (G (O A)) (all (g (G G G)) (all (H1 (O A))
            (all (H2 (O A)) (all (h1 (H1 H1 H1))
            (all (h2 (H2 H2 H2)) (all (phi1 (H1 G))
            (all (phi2 (H2 G))
            (impl (and (and
                        (semi-group-homomorphism
                                                G g H1 h1 phi1))
                        (semi-group-homomorphism
                                                G g H2 h2 phi2))
                        (ex (Phi (H2 H1))
                                 (= (comp Phi phi1) phi2)))
                   (subset (induced-equiv-rel G H1 phi1)
                           (induced-equiv-rel G H2 phi2))
                                                 ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.7b: Let φ_1 and φ_2 be homomorphisms into the semigroups H_1 and H_2 and ρ_1 and ρ_2 the corresponding induced congruences. Let further be $\rho_1 \subset \rho_2$ and φ_1 surjective, then there exists a homomorphism $\Phi : H_1 \to H_2$ with $\Phi\varphi_1 = \varphi_2$.

```
(theorem 5.7b
   (theorem (all (G (O A)) (all (g (G G G)) (all (H1 (O A))
            (all (H2 (O A)) (all (h1 (H1 H1 H1))
            (all (h2 (H2 H2 H2)) (all (phi1 (H1 G))
            (all (phi2 (H2 G))
            (impl (and (and (and
                      (semi-group-homomorphism G g H1 h1 phi1)
                      (semi-group-homomorphism G g H2 h2 phi2))
                      (subset (induced-equiv-rel G H1 phi1)
                              (induced-equiv-rel G H2 phi2)))
                      (surjective G H1 phi1))
                   (ex (Phi (H2 H1))
                      (and (semi-group-homomorphism
                                           H1 h1 H2 h2 Phi)
                           (= (comp Phi phi1) phi2))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.7c: Let φ_1 and φ_2 be homomorphisms into the semigroups H_1 and H_2 and ρ_1 and ρ_2 the corresponding induced congruences. Let further be $\rho_1 \subset \rho_2$ and φ_1 surjective, then there exists at most one homomorphism $\Phi : H_1 \longrightarrow H_2$ with $\Phi\varphi_1 = \varphi_2$.

```
(theorem 5.7c
   (theorem (all (G (O A)) (all (g (G G G)) (all (H1 (O A))
            (all (H2 (O A)) (all (h1 (H1 H1 H1))
            (all (h2 (H2 H2 H2)) (all (phi1 (H1 G))
            (all (phi2 (H2 G)) (all (Phi1 (H2 H1))
            (all (Phi2 (H2 H1))
            (impl (and (and (and (and (and (and (and
                    (semi-group-homomorphism G g H1 h1 phi1)
                    (semi-group-homomorphism G g H2 h2 phi2))
                    (subset (induced-equiv-rel G H1 phi1)
                              (induced-equiv-rel G H2 phi2)))
                    (surjective G H1 phi1))
                    (semi-group-homomorphism H1 h1 H2 h2 Phi1))
                    (= (comp Phi1 phi1) phi2))
                    (semi-group-homomorphism H1 h1 H2 h2 Phi2))
                    (= (comp Phi2 phi1) phi2))
                  (= Phi1 Phi2)))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.7d: Let φ_1 and φ_2 be homomorphisms into the semigroups H_1 and H_2 and ρ_1 and ρ_2 the corresponding induced congruences. Let further be $\rho_1 \subset \rho_2$, φ_1 and φ_2 be surjective, then $\Phi : H_1 \longrightarrow H_2$ is also surjective.

```
(theorem 5.7d
   (theorem (all (G (O A)) (all (g (G G G)) (all (H1 (O A))
            (all (H2 (O A)) (all (h1 (H1 H1 H1))
            (all (h2 (H2 H2 H2)) (all (phi1 (H1 G))
            (all (phi2 (H2 G))
            (impl (and (and (and (and
                       (semi-group-homomorphism G g H1 h1 phi1)
                       (semi-group-homomorphism G g H2 h2 phi2))
                       (subset (induced-equiv-rel G H1 phi1)
                               (induced-equiv-rel G H2 phi2)))
                       (surjective G H1 phi1))
                       (surjective G H2 phi2))
                  (ex (Phi (H2 H1))
                      (and (and (semi-group-homomorphism
                                                  H1 h1 H2 h2 Phi)
                                (= (comp Phi phi1) phi2))
                           (surjective H1 H2 Phi)))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.8 Satz: F/ρ_2 ist homomorphes Bild von F/ρ_1 genau dann, wenn $\rho_1 \subset \rho_2$.

*** theorem 5.8: Let φ_1 and φ_2 be homomorphisms into the semigroups H_1 and H_2 and ρ_1 and ρ_2 the corresponding induced congruences. Then: $\exists$ epimorphism $\Phi : G/\rho_1 \to G/\rho_2 \Leftrightarrow \rho_1 \subset \rho_2$.
The formulation of this theorem is unsufficient in the book.

```
(theorem 5.8
   (theorem (all (G (O A)) (all (g (G G G)) (all (H1 (O A))
            (all (H2 (O A)) (all (h1 (H1 H1 H1))
            (all (h2 (H2 H2 H2)) (all (phi1 (H1 G))
            (all (phi2 (H2 G))
            (impl (and
                       (semi-group-homomorphism G g H1 h1 phi1)
                       (semi-group-homomorphism G g H2 h2 phi2))
                 (equiv (ex (Phi
                               ((quotient-set G
                                 (induced-equiv-rel G H2 phi2))
                                (quotient-set G
                                 (induced-equiv-rel G H1 phi1)))
                        (semi-group-epimorphism
                             (quotient-set G
                               (induced-equiv-rel G H1 phi1))
                             (comp-cong-quot-set G f
                               (induced-equiv-rel G H1 phi1))
                             (quotient-set G
                               (induced-equiv-rel G H2 phi2))
                             (comp-cong-quot-set G f
                               (induced-equiv-rel G H2 phi2))
                                                           Phi)
                        (subset (induced-equiv-rel G H1 phi1)
                                (induced-equiv-rel G H2 phi2))
                                                   ))))))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.9: Sei also F eine Gruppe und ρ eine Linkskongruenz. Dann ist $\rho 1 = U$ eine Untergruppe von F. Mithin gilt
$f \rho g \quad \Leftrightarrow \quad f \in gU$.

*** definition of $\rho 1 = U$ as characteristic subgroup.

```
(simple-definition char-sub-group
  (parameters (G (O A))
              (f (G G G))
              (rho (O G G))
  (preconditions (and (group G f)
                      (left-congruence G f rho)))
  (definition (left-domain-of-s G rho (one G f)))
  (theory groups))
```

*** theorem 5.9a: Let G be a group and ρ a left congruence, then $\rho 1 = U$ is a subgroup of G.

```
(theorem 5.9a
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (impl (group G f)
                  (sub-group (char-sub-group G f rho)
                             G f))))))
   (status "conjectured")
   (theory groups))
```

*** theorem 5.9b: Let G be a group, ρ a left congruence, $\rho 1 = U$, then: $(x, y) \in \rho \quad \Leftrightarrow \quad x \in yU$.

```
(theorem 5.9b
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
            (all (x G) (all (y G)
            (impl (group G f)
                  (equiv (rho x y)
                         ((left-complex-product G y
                                   (char-sub-group G f rho) f)
                                                      x))))))))))
   (status "conjectured")
   (theory groups))
```

5.10: Die Untergruppe U erfüllt die Bedingung
$fUf^{-1} = U$ für alle $f \in F$.
Solche Untergruppen heißen *Normalteiler* von F.

*** normal divisor

```
(simple-definition normal-divisor
   (parameters (G (O A))
               (f (G G G))
               (U (O A)))
   (optional parameters (n G)
                        (inv (G G)))
   (preconditions (and (group G f n inv)
                       (sub-group U G f)))
   (definition (all (x G)
               (= U (left-complex-product G x (right-complex-
                       product G U (inv x) f) f))))
   (theory groups))
```

*** theorem 5.10: The subgroup U of 5.9 fulfils the condition $xUx^{-1} = U$ for all x in G.

```
(theorem 5.10
   (theorem (all (G (O A)) (all (f (G G G)) (all (1 G)
            (all (inv G) (all (rho (O G G))
            (impl (group G f 1 inv)
                  (normal-divisor G f
                    (char-sub-group G f rho) 1 inv))))))))
   (status "conjectured")
   (theory groups))
```

5.11: Sei L ein Linksideal in F, so ist
$f \rho_L g \quad :\Leftrightarrow \quad f = g \quad \vee \quad f, g \in L$
eine Linkskongruenz, deren Klassen das Linksideal L und die einelementigen Mengen $\{f\}$ mit $f \in F \setminus L$ sind.
Ähnlich führen Rechtsideale auf Rechtskongruenzen und zweiseitige Ideale auf Kongruenzen.

*** induced left congruence

```
(implicit-definition induced-left-congruence
   (parameters (L (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (and (semi-group G f)
                       (left-ideal L G f)))
   (definition (all (x A) (all (y A)
               (equiv ((induced-left-congruence L G f) x y)
                      (or (= x y)
                          (and (L x) (L y)))))))
   (theory semi-groups))
```

*** induced right congruence

```
(implicit-definition induced-right-congruence
   (parameters (R (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (and (semi-group G f)
                       (right-ideal R G f)))
   (definition (all (x A) (all (y A)
               (equiv ((induced-right-congruence R G f) x y)
                      (or (= x y)
                          (and (R x) (R y)))))))
   (theory semi-groups))
```

*** induced two-sided congruence

```
(implicit-definition induced-two-sided-congruence
   (parameters (T (O A))
               (G (O A))
               (f (G G G)))
   (sort O)
   (preconditions (and (semi-group G f)
                       (two-sided-ideal T G f)))
   (definition (all (x A) (all (y A)
               (equiv ((induced-two-sided-
                              congruence T G f) x y)
                      (or (= x y)
                          (and (T x) (T y)))))))
   (theory semi-groups))
```

*** theorem 5.11a1: the induced left congruence is indeed a left congruence.

```
(theorem 5.11a1
   (theorem (all (G (O A)) (all (f (G G G)) (all (L (O A))
            (impl (and (semi-group G f)
                       (left-ideal L G f))
                  (left-congruence G f
                       (induced-left-congruence L G f)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.11a2: the classes of the induced left congruence are the left ideal L and the one element sets $\{x\}$ with $x \in G \setminus L$.

```
(theorem 5.11a2
   (theorem (all (G (O A)) (all (f (G G G)) (all (L (O A))
            (all (X (O A))
            (impl (and (semi-group G f)
                       (left-ideal L G f))
                  (equiv ((set-of-all-equiv-classes G
                             (induced-left-congruence L G f)) X)
                         (or (= X L)
                             (ex (x G) (and (not (L x))
                                       (and (X x)
                                            (singleton X))
                                                     ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.11b1: the induced right congruence is indeed a right congruence.

```
(theorem 5.11b1
   (theorem (all (G (O A)) (all (f (G G G)) (all (R (O A))
            (impl (and (semi-group G f)
                       (right-ideal R G f))
                  (right-congruence G f
                       (induced-right-congruence R G f)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.11b2: the classes of the induced right congruence are the right ideal R and the one element sets $\{x\}$ with $x \in G \setminus R$.

```
(theorem 5.11b2
   (theorem (all (G (O A)) (all (f (G G G)) (all (R (O A))
            (all (X (O A))
            (impl (and (semi-group G f)
                       (right-ideal R G f))
                  (equiv ((set-of-all-equiv-classes G
                           (induced-right-congruence R G f)) X)
                         (or (= X R)
                             (ex (x G) (and (not (R x))
                                       (and (X x)
                                            (singleton X))
                                                 ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.11c1: the induced two-sided congruence is indeed a two-sided congruence.

```
(theorem 5.11c1
   (theorem (all (G (O A)) (all (f (G G G)) (all (T (O A))
            (impl (and (semi-group G f)
                       (two-sided-ideal T G f))
                  (two-sided-congruence G f
                    (induced-two-sided-congruence
                                         T G f)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.11c2: the classes of the induced two-sided congruence are the twosided ideal J and the one element sets $\{x\}$ with $x \in G \setminus J$.

```
(theorem 5.11c2
   (theorem (all (G (O A)) (all (f (G G G)) (all (T (O A))
            (all (X (O A))
            (impl (and (semi-group G f)
                       (two-sided-ideal T G f))
                  (equiv ((set-of-all-equiv-classes G
                              (induced-two-sided-
                                    congruence T G f)) X)
                         (or (= X T)
                             (ex (x G) (and (not (T x))
                                       (and (X x)
                                            (singleton X))
                                                ))))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.12 Def.: Sei I ein zweiseitiges Ideal in F, so heißt $F/I := F/\rho_I$ versehen mit der Verknüpfung (5.4) die *Reessche Quotientenhalbgruppe* nach dem Ideal I.

*** theorem 5.12: If I is a two-sided ideal of the semi-group G then G/I together with the composition of 5.4 forms a semi-group.

```
(theorem 5.12
   (theorem (all (G (O A)) (all (f (G G G)) (all (J (O A))
            (impl (and (semi-group G f)
                       (two-sided-ideal J G f))
                  (semi-group
                       (quotient-set G
                          (induced-congruence J G f))
                       (comp-cong-quot-set G f
                          (induced-congruence J G f))))))))
   (status "conjectured")
   (theory semi-groups))
```

*** the rees quotient semigroup
As in the definition of the quotient semigroup this problem cannot be easily encoded in Post because of the limitation of the language.

```
(simple-definition rees-quot-semi-group
   (parameters (G (O A))
               (f (G G G))
               (J (O A)))
   (sort (O (O A)))
   (preconditions (and (semi-group G f)
                       (two-sided-ideal J G f)))
   (definition (*pair* (quotient-set G
                          (induced-congruence J G f))
                      (comp-cong-quot-set G f
                            (induced-congruence J G f))))
   (theory semi-groups))
```

5.13 Def.: Sei ρ eine Links- [Rechts-] Kongruenz in F. Die Menge
$N(\rho) := \{g \in F : \forall f \in F : (\rho f) \cdot g \subset \rho(fg)\}$
$[N(\rho) := \{g \in F : \forall f \in F : g \cdot (\rho f) \subset \rho(gf)\}]$
heißt der *Normalisator* von ρ in F.

*** left normalizer

```
(implicit-definition left-normalizer
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G)))
   (preconditions (left-congruence G f rho))
   (sort O)
   (definition (all (x G)
               (equiv ((left-normalizer G f rho) x)
                      (all (y G)
                      (subset (right-complex-product G
                                 (left-domain-of-s G rho y)
                                 x f)
                              (left-domain-of-s G rho (f y x))
                                                              )))))
   (theory semi-groups))
```

*** right normalizer

```
(implicit-definition right-normalizer
   (parameters (G (O A))
               (f (G G G))
               (rho (O G G)))
   (preconditions (right-congruence G f rho))
   (sort O)
   (definition (all (x G)
               (equiv ((right-normalizer G f rho) x)
                      (all (y G)
                      (subset (left-complex-product G x
                                    (left-domain-of-s G rho y) f)
                              (left-domain-of-s G rho (f x y)))
                                                              ))))
   (theory semi-groups))
```

Nun ist zwar für eine Teilmenge $T \subset F$ ρ nicht mehr Äquivalenzrelation auf T im Sinne der ursprünglichen Definition, wohl aber die *Einschränkung* $\rho \cap (T \times T)$ von ρ auf T. Etwas ungenau, aber sprachlich einfacher, können wir dennoch von der Äquivalenzrelation ρ auf T reden.

*** the restriction of a relation

```
(implicit-definition restricted-rel
   (parameters (T (O A))
               (G (O A))
               (rho (O G G)))
   (sort (O G G))
   (preconditions (subset T G))
   (definition (all (x G) (all (y G)
               (equiv ((restricted-rel T G rho) x y)
                      (and (rho x y)
                           (and (T x) (T y)))))))
   (status "conjectured")
   (theory binary-relations))
```

*** theorem 5.13a: For a subset T of G the left congruence ρ isno equivalence relation on T, but the restriction $\rho \cap (T \times T)$ of ρ on T is an equivalence relation.

```
(theorem 5.13a
   (theorem (all (T (O A)) (all (G (O A)) (all (f (G G G))
            (all (rho (O G G))
            (impl (and (subset T G)
                       (left-congruence G f rho))
                  (equiv-rel T (restricted-rel (T G rho)))
                                                   ))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.13b: For a subset T of G is the right congruence ρ no equivalence relation on T, but the restriction $\rho \cap (T \times T)$ of ρ on T is an equivalence relation.

```
(theorem 5.13b
   (theorem (all (T (O A)) (all (G (O A)) (all (f (G G G))
            (all (rho (O G G))
            (impl (and (subset T G)
                       (right-congruence G f rho))
                  (equiv-rel T (restricted-rel T G rho)))
                                                   )))))
   (status "conjectured")
   (theory semi-groups))
```

In diesem Sinne ist nun $N(\rho)$ die größte Teilmenge von F, auf der die Linkskongruenz ρ auch Rechtskongruenz, also Kongruenz schlechthin ist.

*** theorem 5.13c: On the left normalizer of G the left congruence ρ is also a right congruence.

```
(theorem 5.13c
   (theorem (all (G (O A)) (all (f (G G G)) (all (rho (O G G))
        (impl (and (left-congruence G f rho)
                   (left-congruence
                              (left-normalizer G f rho) f rho))
              (right-congruence (left-normalizer G f rho) f
                                (restricted-rel
                                    (left-normalizer G f rho)
                                                  G rho)))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.13d: $N(\rho)$ is the largest subset of G, on which the left congruence ρ is also a right congruence.

```
(theorem 5.13d
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G)) (all (X (O A))
            (impl (and (and (and (subset X G)
                       (left-congruence G f rho))
                       (left-congruence X f
                           (restricted-rel X G rho)))
                       (right-congruence X f
                           (restricted-rel X G rho)))
                  (subset X (left-normalizer G f rho))
                                                   ))))))
   (status "conjectured")
   (theory semi-groups))
```

5.14 Satz: $N(\rho)$ ist Unterhalbgruppe von F.

*** theorem 5.14a: The left normalizer $N(\rho)$ is a sub-semigroup of G.

```
(theorem 5.14a
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G))
            (impl (left-congruence G f rho)
                  (sub-semi-group
                       (left-normalizer G f rho) G f))))))
   (status "conjectured")
   (theory semi-groups))
```

*** theorem 5.13b: The right normalizer $N(\rho)$ is a sub-semigroup of G.

```
(theorem 5.14b
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G))
            (impl (right-congruence G f rho)
                  (sub-semi-group
                       (right-normalizer G f rho) G f))))))
   (status "conjectured")
   (theory semi-groups))
```

5.15: Sei F eine Gruppe und bezeichne U wieder $\rho 1$. Es gilt also hier $N(\rho) = \{f \in F : f^{-1}Uf \subset U\}$.

*** theorem 5.15: Let G be a group and $U = \rho 1$. Then there is $N(\rho) = \{f \in F : f^{-1}Uf \subset U\}$.

```
(theorem 5.15
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G)) (all (1 G) (all (inv (G G))
            (impl (group G f 1 inv)
                  (all (x G)
                  (equiv ((left-normalizer G f rho) x)
                         (subset (left-complex-product
                                        G (inv x)
                                      (right-complex-product G
                                        (char-sub-group
                                            G f rho 1)
                                     x f) f)
                                 (char-sub-group G f rho 1))
                                                   )))))))))
   (status "conjectured")
   (theory semi-groups))
```

5.16: Der Normalisator $N(\rho)$ einer Linkskongruenz auf einer Gruppe F ist wohl zu unterscheiden von dem *Normalisator der Untergruppe* U, der durch
$N(U) := \{f \in F : f^{-1}Uf = U\}$
definiert ist.

*** normalizer of a subgroup

```
(implicit-definition normalizer-of-sub-group
   (parameters (G (O A))
               (f (G G G))
               (U (O A)))
   (optional-parameters (1 G)
                        (inv (G G)))
   (sort (O A))
   (preconditions (and (group G f 1 inv)
                       (sub-group U G f)))
   (definition (all (x G)
               (equiv ((normalizer-of-sub-group G f U) x)
                      (= U
                         (left-complex-product G (inv x)
                           (right-complex-product G U x f)
                           f)))))
   (status "conjectured")
   (theory semi-groups))
```

5.17 Satz: Sei F eine Gruppe, ρ eine Linkskongruenz in F und $U = \rho 1$. Dann gilt
1. $n \in N(U)$ (genau dann), wenn $n, n^{-1} \in N(\rho)$
2. $N(\rho)$ ist eine Vereinigung von Klassen nach ρ,
d.h. es gilt für alle $f \in F$ die Implikation
$fU \cap N(\rho) \neq \emptyset \rightarrow fU \subset N(\rho)$
3. $N(\rho) \setminus N(U)$ ist ein zweiseitiges Ideal in der Halbgruppe $N(\rho)$.

*** theorem 5.17a: Let G be a group, ρ a left congruence in G and $U = \rho 1$.
Then: $n \in N(U) \Leftrightarrow n, n^{-1} \in N(\rho)$.

```
(theorem 5.17a
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G)) (all (1 G)
            (all (inv (G G)) (all (n G) (all (U (O A))
           (impl (and (and (group G f 1 inv)
                           (left-congruence G f rho))
                      (= U (char-sub-group G f rho)))
              (equiv ((normalizer-of-sub-group
                                         G f U 1 inv) n)
                     (and ((left-normalizer G f rho) n)
                          ((left-normalizer G f rho)
                                 (inv n))))))))))))
   (status "conjectured")
   (theory groups))
```

*** theorem 5.17b: Let G be a group, ρ a left congruence in G and $U = \rho 1$. Then $N(\rho)$ is a union of classes after ρ.

```
(theorem 5.17b
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G))
            (all (x G) (all (U (O A))
            (impl (and (and (and (group G f)
                           (left-congruence G f rho))
                           (= U (char-sub-group G f rho)))
                           (not= empty-set
                                (intersection
                                    (left-complex-product
                                                  G x U f)
                                    (left-normalizer G f
                                                   rho))))
                     (subset (left-complex-product G x U f)
                             (left-normalizer G f rho))
                                                    ))))))))
   (status "conjectured")
   (theory groups))
```

*** theorem 5.17c: Let G be a group, ρ a left congruence in G and $U = \rho 1$. Then $N(\rho) \setminus N(U)$ is a twosided ideal in the semigroup $N(\rho)$.

```
(theorem 5.17c
   (theorem (all (G (O A)) (all (f (G G G))
            (all (rho (O G G)) (all (1 G)
            (all (inv (G G)) (all (x G)
            (all (U (O A)) (all (X (O A))
            (impl (and (and (and (group G f 1 inv)
                       (left-congruence G f rho))
                       (= U (char-sub-group G f rho 1)))
                       (= X (set-difference
                                (left-normalizer G f rho)
                                (normalizer-of-sub-group
                                             G f U 1 inv))))
                  (two-sided-ideal X
                                (left-normalizer G f rho) f)
                                                  ))))))))))
   (status "conjectured")
   (theory groups))
```