Reading an Algebra Textbook

Clemens Ballarin

Stephanienstr. 61
76133 Karlsruhe, Germany
http://www21.in.tum.de/~ballarin/

Abstract. We report on a formalisation experiment where excerpts from an algebra textbook are compared to their translation into formal texts of the Isabelle/Isar prover, and where an attempt is made in the formal text to stick as closely as possible with the structure of the informal counterpart. The purpose of the exercise is to gain understanding on how adequately a modern algebra text can be represented using the module facilities of Isabelle. Our initial results are promising.

1 Introduction

We study an informal mathematical text by translating it to a formal document in the Isabelle/Isar language [17]. Our motivation for this exercise is twofold:

- Can the reuse of concepts present in the informal text be expressed with Isabelle's facilities for modular reasoning in an adequate fashion?
- Are there linguistic features in the informal text that identify that modular reasoning takes place?

This study is currently in a preliminary state, and in this work we focus on the first item. In fact, the work is very preliminary: no proofs have been carried out yet in the formal system.[1]

By Isabelle's facilities for modular reasoning we mainly mean local theory specifications locales [4] and [10]. The informal mathematical text is taken from Nathan Jacobson's textbook on basic algebra [11]. The choice of a modern exposition on algebra is intentional, because we are interested in modularity and how it is reflected in the proof text. In this study, we are less interested in aspects related to the underlying logic of the prover, such as how well partiality can be dealt with.

This note is organised as follows: Section 2 is a brief exposition of locales. For more information, the tutorial [3] and an exposition of their semantics [4] are available. For the complete syntax, see the Isabelle reference manual [18]. The detailed study of the algebra text follows in Section 3. The methodology is simple: quotations from the informal text and their translation to Isabelle are set side-by-side and are commented. Observations made in the experiment are discussed in Section 4.

[1] This means that some definitions likely require some revisions for proofs to go through, but we do not expect significant changes.

2 Locales

Locales are an extension of Isabelle's Isar proof language [17] by means for manipulating "knowledge containers" or modules, which were designed for dealing with algebraic structures. The central concept is the *locale*, a theory functor that maps parameters and a specification to derived operations (that is, operations whose definitions involve the parameters) and theorems.

A locale declaration is of the form

$$\textbf{locale}\ n = \ \textbf{fixes}\ \overline{y}\ +\ \textbf{assumes}\ a_1 : A_1 \ldots a_j : A_j$$

where $\overline{y}$ are the parameters and $A_1 \wedge \ldots \wedge A_j$ is the specification. The A_i are versions of the user input where free variables except parameters are universally closed. For example, a locale for groups may be declared like this:

```
locale group =
  fixes G and composition (infixl "·" 70) and unit ("1")
  assumes assoc: "⟦a ∈ G; b ∈ G; c ∈ G⟧ ⟹ (a · b) · c = a · (b · c)"
    and ...
```

Note that types, in particular types of the parameters, may be left implicit. They are inferred automatically.[2] Operations and theorems may be added to n in *context blocks*:

$$\textbf{context}\ n\ \textbf{begin}\ \ldots\ \textbf{end}$$

These commands are available in context blocks:

$$\textbf{definition}\ c\ \textbf{where}\ c \equiv t$$
$$\textbf{theorem}\ b : B$$
$$\textbf{abbreviation}\ c\ \textbf{where}\ c \equiv t$$

The first two are straightforward: **definition** declares a new operation symbol with defining equation $c \equiv t$ and **theorem** introduces a theorem B named b; **abbreviation** also adds a new operation symbol, but in contrast to definitions, the equation is unfolded while parsing and folded while printing terms. This is useful for providing concrete syntax for existing concepts. Declarations in a context block are persistent — that is, they are present when the context is visited again.

Locales are hierarchic and a graph of interdependent locales is maintained by the system. Dependencies may be given when declaring a locale by inserting a *locale expression* before the **fixes** and **assumes** clauses,

$$\textbf{locale}\ n = \ \mathit{expr}\ +\ \ldots,$$

or between existing locales through

$$\textbf{sublocale}\ n \subseteq \mathit{expr}\ \mathit{rewrites}\ \langle \mathit{proof} \rangle.$$

[2] While inferred types may involve type classes, locales do not rely on them.

An expression is a sequence of locale instances followed by an optional **for** clause:

$$I_1 + \ldots + I_k \ \textbf{for}\ \overline{x}$$

Locale instances I_i are qualified and are either positional or by name. The positional instance $q_i : n_i\ \overline{t_i}$ denotes locale n_i with parameters, in the order of declaration, instantiated by the terms $\overline{t_i}$. Likewise the named instance n_i **where** $\overline{x_i} = \overline{t_i}$ denotes n_i with parameters occurring in $\overline{x_i}$ instantiated by the $\overline{t_i}$. In a locale declaration the terms $\overline{t_i}$ are over the $\overline{x}$ from the **for** clause, which are (together with the $\overline{y}$ from the **fixes** clause) parameters of the declared locale n. In a **sublocale** declaration the $\overline{t_i}$ are over the parameters from the target locale n. The qualifier q_i of an instance I_i is optional and, if present, qualifies symbols of derived operations and theorem names of the instance.

The expression in a locale declaration denotes locale instances that are imported to the declared locale. That is, the specifications of the imported instances contribute to the specifications of the declared locale, and their definitions and theorems are available in its body. Here is a locale declaration involving an expression:

```
locale abelian_group =
  group G composition unit
  for G and composition (infixl "·" 70) and unit ("1") +
  assumes comm: "⟦a ∈ G; b ∈ G⟧ ⟹ a · b ∈ G"
```

As a notational convenience parameters from instantiated locales may be implicitly added to the **for** clause by omitting their instantiation. This is convenient when constructing linear locale hierarchies. For example the above declaration may be abbreviated to

```
locale abelian_group = group +
  assumes comm: "⟦a ∈ G; b ∈ G⟧ ⟹ a · b ∈ G"
```

The **sublocale** command enables changing the locale hierarchy. It is implemented via theory interpretation. Changes are only possible if they are implied by the specification of the involved locales and must be supported by proofs. The optional *rewrites* clause is of the form **where** $eq_1 \ldots eq_m$ and lets one change derived operations of the instantiated locales through rewrite rules eq_i. In combination with instantiation they provide signature morphisms [8] on the instantiated locales.

It is important to note that locales are extra-logical. That is, the functors they represent are not encoded in Isabelle's logic. However, each locale is accompanied by a locale predicate n which reflects the specification of n in the logic.

3 Detailed Account

We study Jacobson's presentation of monoids, groups and rings and the congruence relations defined over these structures. For groups and rings congruences can be characterised by working modulo suitable substructures: normal subgroups in

the case of groups, and ideals for rings. In the sequel, excerpts from Jacobson's text [11] and their formal analogs are interleaved. In the latter, no proofs were actually carried out, and ⟨*proof*⟩ merely indicates the places where Isabelle requests proofs. With exception of a few details such as hiding unwanted concrete syntax of Isabelle's standard library, the entire formal text is reproduced.

3.1 Preliminaries

Fundamental to congruences are equivalence relations over sets and the induced partitions of the sets into equivalence classes:

> A relation E on a set S is called an *equivalence relation* if the following conditions hold for any a, b, c, in S:
> 1. aEa (reflexive property).
> 2. $aEb \Rightarrow bEa$ (symmetry).
> 3. aEb and $bEc \Rightarrow aEc$ (transitivity).
>
> If $a \in S$ we let $\bar{a}_E$ (or simply $\bar{a}$) $= \{b \in S | bEa\}$. We call $\bar{a}_E$ the *equivalence class (relative to E or E-equivalence class) determined by a.*
>
> If E is an equivalence relation, the associated partition $\pi = \{\bar{a} | a \in S\}$ is called the *quotient set of S relative to the relation E.* We shall usually denote π as S/E. [11, p 11f]

The translation to a locale is straightforward. Set and equivalence relation are parameters of the `equivalence` locale, class and quotient set derived operations.

```
locale equivalence =
  fixes S and E
  assumes carrier: "E ⊆ S × S"
    and refl: "a ∈ S ⟹ (a, a) ∈ E"
    and sym: "(a, b) ∈ E ⟹ (b, a) ∈ E"
    and trans: "⟦(a, b) ∈ E; (b, c) ∈ E⟧ ⟹ (a, c) ∈ E"
begin
definition "class" where "class = (λa ∈ S. {b ∈ S. (b, a) ∈ E})"
definition "Quot" where "Quot = {class a | a. a ∈ S}"
end
```

3.2 Groups

Jacobson's base for the hierarchy of group structures are monoids:

> **Definition 1.1** *A* monoid *is a triple $(M, p, 1)$ in which M is a non-vacuous set, p is an associative binary composition (or product) in M, and 1 is an element of M such that $p(1, a) = a = p(a, 1)$ for all $a \in M$.* [11, p 28]

It is convenient to split the closedness properties (of p being a composition in M and $1 \in M$) and the algebraic laws into two locales because closure is also part of the definition of submonoid.

```
locale monoid_closed =
  fixes M and composition (infixl "·" 70) and unit ("1")
  assumes composition_closed: "⟦a ∈ M; b ∈ M⟧ ⟹ a · b ∈ M"
    and unit_closed: "1 ∈ M"

locale monoid = monoid_closed +
  assumes assoc: "⟦a ∈ M; b ∈ M; c ∈ M⟧ ⟹ (a · b) · c = a · (b · c)"
    and left_unit: "a ∈ M ⟹ 1 · a = a"
    and right_unit: "a ∈ M ⟹ a · 1 = a"
begin
```

The locales have individual parameters for set and operations while according to Jacobson a monoid is a triple. While it is possible to use a single locale parameter for the entire algebraic structure (preferably using records [13]) we prefer individual parameters, since this will simplify the definition of ring.

In the context of monoids the notions of submonoid, invertible element and inverse are defined:

> If M is a monoid, a subset N of M is called a *submonoid* of M if N contains 1 and N is closed under the product in M, that is, $n_1 n_2 \in N$ for every $n_i \in N$. [11, p 29]

> An element of a monoid M is said to be *invertible* (or a *unit*) if there exists a v in M such that
>
> $$uv = 1 = vu. \tag{3}$$
>
> If v' also satisfies $uv' = 1 = v'u$ then $v' = (vu)v' = v(uv') = v$. Hence v satisfying (3) is unique. We call this *the inverse* of u and write $v = u^{-1}$. [11, p 31]

The translation of these concepts is straightforward, but note the use of the locale predicate `monoid_closed` in the definition of `submonoid`:

```
definition submonoid where "submonoid N K ⟷
  N ⊆ K ∧ monoid_closed N composition unit"
definition inverses
  where "⟦u ∈ M; v ∈ M⟧ ⟹ inverses u v ⟷ u · v = 1 ∧ v · u = 1"
definition invertible
  where "u ∈ M ⟹ invertible u ⟷ (∃v ∈ M. inverses u v)"
definition inverse
  where "u ∈ M ⟹ inverse u = (THE v. v ∈ M ∧ inverses u v)"
end
```

In the group definition found in many textbooks, the inverse is a parameter. For Jacobson the inverse is a derived operation.

> **Definition 1.2** *A group G* (or $(G, p, 1)$) is a monoid all of whose elements are invertible. [11, p 31]

Based on the previously defined predicate `invertible` the locale for groups is declared thus:

```
locale group = monoid G composition unit
  for G and composition (infixl "·" 70) and unit ("1") +
  assumes "a ∈ G ⟹ invertible a"
```

Jacobson defines subgroups in the context of monoids, not just for groups.

> We shall call a submonoid of a monoid M (in particular, of a group) a *subgroup* if, regarded as a monoid, it is a group. [11, p 31]

The translation again involves a locale predicate, namely `group`.

```
context monoid begin
definition subgroup where "subgroup H K ⟷
  submonoid H K ∧ group H composition unit"
end
```

Commutative structures are required later for rings and are introduced now.

> If $ab = ba$ in M then a and b are said to *commute* and if this happens for all a and b in M then M is called a *commutative monoid*. Commutative groups are generally called *abelian groups* after Niels Abel, ... [11, p 40]

The corresponding locales are declared using short notation omitting the **for** clause.

```
locale commutative_monoid = monoid +
  assumes comm: "⟦a ∈ M; b ∈ M⟧ ⟹ a · b ∈ M"
locale abelian_group =
  group + commutative_monoid G composition unit
```

Next Jacobson introduces cosets. These will later be identified as the elements of the factor group, which is a group obtained through a particular equivalence relation.

> Now let G be any group and let H be a subgroup of G. .... If $x \in G$ then it is clear that its $H_L(G)$-orbit is
>
> $$Hx = \{hx | h \in H\}. \tag{23}$$
>
> .... Accordingly, we shall ...call Hx the *right coset* of x relative to H. .... The orbit of x in this case is $xH = \{xh | h \in H\}$ and this is called the *left coset* of x relative to H. [11, p 52f]

Jacobson constructs right cosets as orbits of the transformation group $H_L(G)$ of left translations $x \to hx$ in G for $h \in H$. Since the properties of cosets derived from this construction are not relevant in the subsequent discussion, we use his equation 23 as definition. Likewise for left cosets.

```
context group begin
definition Right_Coset (infixl "|·" 70)
```

```
  where "H |· x = {h · x | h. h ∈ H}"
definition Left_Coset (infixl "·|" 70)
  where "x ·| H = {x · h | h. h ∈ H}"
end
```

Jacobson now introduces congruences in monoids. Composition and unit are lifted to congruence classes, obtaining the quotient monoid.

> **Definition 1.4** *Let $(M, \cdot, 1)$ be a monoid. A* congruence (*or* congruence relation) $\equiv$ *in M is an equivalence relation in M such that for any a, a', b, b' such that $a \equiv a'$ and $b \equiv b'$ one has $ab \equiv a'b'$. (In other words, congruences are equivalence relations which can be multiplied.)*
>
> Let $\equiv$ be a congruence in the monoid M and consider the quotient set $\bar{M} = M/\equiv$ of M relative to $\equiv$. .... Since congruences can be multiplied it is clear in the general case that, if $\bar{a} = \bar{a'}$ and $\bar{b} = \bar{b'}$, then $\overline{ab} = \overline{a'b'}$. Hence
>
> $$(\bar{a}, \bar{b}) \to \overline{ab}$$
>
> is a well-defined map of $\bar{M} \times \bar{M}$ into $\bar{M}$; that is, this is a binary composition on $\bar{M}$. We denote this again as $\cdot$, and we shall now show that $(\bar{M}, \cdot, \bar{1})$ is a monoid. .... The monoid $(\bar{M}, \cdot, \bar{1})$ is called the *quotient monoid of M relative to the congruence* $\equiv$. [11, p 54f]

In the translation to Isabelle we call the conguence relation `E`. The quotient set is `Quot`, lifted composition is `class_composition` and for matters of symmetry lifted unit `class_unit`.

```
locale monoid_congruence = monoid + equivalence where S = M +
  assumes cong:
    "⟦(a, a') ∈ E; (b, b') ∈ E⟧ ⟹ (a · b, a' · b') ∈ E"
begin
definition "class_composition = (λX ∈ Quot. λY ∈ Quot.
  THE Z. ∃x ∈ X. ∃y ∈ Y. Z = class (x · y))"
definition "class_unit = class 1"
end
```

That the quotient set is again a monoid is expressed concisely with a sublocale declaration. We declare a new locale equivalent to `monoid_congruence` so that the current state of the latter remains available.

```
locale quotient_monoid = monoid_congruence
sublocale quotient_monoid ⊆
  quotient!: monoid Quot class_composition class_unit ⟨proof⟩
```

Next Jacobson considers congruences on groups.

> We can say a good deal more if $M = G$ is a group and $\equiv$ is a congruence on G. In the first place, in this case the quotient monoid $(\bar{G}, \cdot, \bar{1})$ is a group since $\bar{a}\overline{a^{-1}} = \bar{1} = \overline{a^{-1}}\bar{a}$. Hence every $\bar{a}$ is invertible and its inverse is $\overline{a^{-1}}$. [11, p 55]

The corresponding constructions in Isabelle are analogous to those for monoids. Notably the sublocale declaration now involves a rewrite clause for mapping the inverse operation of the quotient group to `class_inverse`.

```
locale group_congruence = group + monoid_congruence where M = G
begin
definition "class_inverse =
  (λX ∈ Quot. THE Y. ∃x ∈ X. Y = class (inverse x))"
end
locale quotient_group = group_congruence
sublocale quotient_group ⊆
  quotient!: group Quot class_composition class_unit
  where "quotient.inverse = class_inverse" ⟨proof⟩
```

Now follows the main result for congruences on groups: the factor group induced by a normal subgroup. First comes the definition of normal, then the main theorem.

Definition 1.5 *A subgroup K of a group G is said to be* normal *... if*

$$g^{-1}kg \in K$$

for every $g \in G$ and $k \in K$.

[11, p 55]

```
context group begin
definition normal where "normal H K ⟷
  subgroup H K ∧ (∀k ∈ K. ∀h ∈ H. inverse k · h · k ∈ H)"
end
```

Theorem 1.6 *Let G be a group and $\equiv$ a congruence on G. Then the congruence class $K = \bar{1}$ of the unit is a normal subgroup of G and for any $g \in G$, $\bar{g} = Kg = gK$, the right or left coset of g relative to K. Conversely let K be any normal subgroup of G, then $\equiv$ defined by:*

$$a \equiv b \pmod{K} \quad \textit{if} \quad a^{-1}b \in K$$

is a congruence relation in G whose associated congruence classes are the left (or right) cosets gK.

[11, p 55f]

This theorem consists of two parts. The first is about arbitrary group congruences.

```
context group_congruence begin
theorem "normal class_unit G" ⟨proof⟩
theorem "g ∈ G ⟹ class g = class_unit |· g" ⟨proof⟩
theorem "g ∈ G ⟹ class g = g ·| class_unit" ⟨proof⟩
end
```

In the second part, a congruence relation is constructed from a normal subgroup K, and the classes are identified as cosets. We call the corresponding locale `factor_group` since the factor group will be defined in this context as well.

```
locale factor_group = group +
  fixes K assumes normal: "normal K G"
begin
definition "Cong = {(x, y). inverse x · y ∈ K}"
theorem "monoid_congruence G op · 1 Cong" ⟨proof⟩
theorem "g ∈ G ⟹ equivalence.class G Cong g = K |· x" ⟨proof⟩
theorem "g ∈ G ⟹ equivalence.class G Cong g = x ·| K" ⟨proof⟩
end
```

Finally Jacobson identifies the factor group.

> We shall now write G/K for $\bar{G} = G/K \pmod{K}$ and call this the *factor group* (or *quotient group*) of G relative to the normal subgroup K. By definition, the product in G/K is
>
> $$(gK)(hK) = ghK, \tag{26}$$
>
> $K = 1K$ is the unit, and the inverse of gK is $g^{-1}K$. [11, p 56]

This is again expressed with a sublocale declaration.

```
sublocale factor_group ⊆ quotient_group G composition unit Cong
  where "class_composition = (λX ∈ Quot. λY ∈ Quot.
      THE Z. ∃g ∈ X. ∃h ∈ Y. Z = g · h ·| K)"
    and "class_unit = K"
    and "class_inverse = (λX ∈ Quot.
      THE Y. ∃g ∈ X. Z = inverse g ·| K)"
  ⟨proof⟩
```

The last result on groups presented here will be needed when studying ideals, which give rise to congruences on rings, such as normal subgroups do for groups:

> It is clear from the definition that any subgroup of an abelian group is normal. [11, p 57]

```
context abelian_group begin
lemma "subgroup H K ⟹ normal H K" ⟨proof⟩
end
```

We are now ready to turn to rings.

3.3 Rings

These are built upon groups, and for congruences on rings much of the constructions for groups is reused.

Definition 2.1 *A* ring *is a structure consisting of a non-vacuous set R together with two binary compositions $+$, $\cdot$ in R and two distinguished elements $0, 1 \in R$ such that*
1. $(R, +, 0)$ is an abelian group.
2. $(R, \cdot, 1)$ is a monoid.
3. The distributive laws D $\quad a(b+c) = ab + ac \quad (b+c)a = ba + ca$ hold for all $a, b, c \in R$.

[11, p 86]

This definition translates directly to locales. Since the additive inverse of a will be donoted by $-a$ and $a + (-b)$ by $a - b$ this syntax is introduced here as well.

```
locale ring =
  additive: abelian_group R addition zero +
  multiplicative: monoid R multiplication one
  for R and addition (infixl "+" 65) and zero ("0")
    and multiplication (infixl "·" 70) and one ("1") +
  assumes D:
    "⟦a ∈ R; b ∈ R; c ∈ R⟧ ⟹ a · (b + c) = a · b + a · c"
    "⟦a ∈ R; b ∈ R; c ∈ R⟧ ⟹ (b + c) · a = b · a + c · a"
begin
abbreviation unary_minus ("- _" [81] 80)
  where "unary_minus ≡ additive.inverse"
abbreviation minus (infixl "-" 65) where "a - b ≡ a + (-b)"
abbreviation inverse where "inverse ≡ multiplicative.inverse"
```

The translation of the notions of subring and congruence into locales is equally simple.

A subset S of a ring R is a *subring* if S is a subgroup of the additive group and also a submonoid of the multiplicative monoid of R. [11, p 87]

```
definition subring where "subring S T ⟷
  additive.subgroup S T ∧ multiplicative.submonoid S T"
end
```

We define a congruence $\equiv$ in a ring to be a relation in R which is a congruence for the additive group $(R, +, 0)$ and the multiplicative monoid $(R, \cdot, 1)$. [11, p 101]

```
locale ring_congruence = ring +
  additive: group_congruence R addition zero E +
  multiplicative: monoid_congruence R multiplication one E for E
```

Next Jacobson gives notions for the quotient set and its operations.

Hence $\equiv$ is an equivalence relation such that $a \equiv a'$ and $b \equiv b'$ imply $a + b \equiv a' + b'$ and $ab \equiv a'b'$. Let $\bar{a}$ denote the congruence class of $a \in R$ and let $\bar{R}$ be the quotient set. As we have seen in section 1.5, we have binary compositions $+$ and $\cdot$ in $\bar{R}$ defined by $\bar{a} + \bar{b} = \overline{a+b}$, $\bar{a} \cdot \bar{b} = \overline{ab}$. [11, p 101]

In the translation the quotient set remains `Quot`. Notation for the operations is as follows:

```
context ring_congruence begin
abbreviation "class_addition ≡ additive.class_composition"
abbreviation "class_zero ≡ additive.class_unit"
abbreviation
  "class_multiplication ≡ multiplicative.class_composition"
abbreviation "class_one ≡ multiplicative.class_unit"
end
```

As with monoids and groups the quotient set and operations give rise to a ring.

> These define the group $(\bar{R}, +, 0)$ and the monoid $(\bar{R}, \cdot, 1)$. We also have
>
> $$\bar{a}(\bar{b} + \bar{c}) = \bar{a}(\overline{b + c}) = \overline{a(b + c)} = \overline{ab + ac} = \overline{ab} + \overline{ac} = \bar{a}\bar{b} + \bar{a}\bar{c}.$$
>
> Similarly, $(\bar{b} + \bar{c})\bar{a} = \bar{b}\bar{a} + \bar{c}\bar{a}$. Hence $(\bar{R}, +, \cdot, \bar{0}, \bar{1})$ is a ring which we shall call a *quotient* (or *difference*) *ring* of R. [11, p 101]

```
locale quotient_ring = ring_congruence
sublocale quotient_ring ⊆
  quotient!: ring Quot class_addition class_zero
    class_multiplication class_one ⟨proof⟩
```

Coset and ideal are the analog for left coset and normal subgroup.

> We recall that the congruences in $(R, +, 0)$ are obtained from the subgroups I (necessarily normal since $(R, +)$ is commutative) by defining $a \equiv b$ if $a - b \in I$. Then the congruence class $\bar{a}$ is the coset $a + I$. .... We now give the following
>
> **Definition 2.2** *If R is a ring, an ideal I of R is a subgroup of the additive group such that for any $a \in R$ and $b \in I$, ab and $ba \in I$.*
>
> [11, p 101]

```
context ring begin
abbreviation Coset (infixl "+|" 65) where "Coset ≡ additive.Left_Coset"
definition ideal where "ideal I R' ⟷
  additive.subgroup I R' ∧ (∀a ∈ R'. ∀b ∈ I. a · b ∈ I ∧ b · a ∈ I)"
end
```

Finally we arrive at the quotient ring with respect to an ideal, which is the ring analog to factor group.

> Our results show that congruences in a ring R are obtained from ideals I of R by defining $a \equiv a'$ if $a - a' \in I$. The corresponding quotient ring will be denoted as R/I and will be called *quotient ring of R with respect*

to the ideal I. The elements of R/I are the cosets $a + I$ and the addition and multiplication in R/I are defined by

$$\begin{aligned}(a + I) + (b + I) &= (a + b) + I \\ (a + I)(b + I) &= ab + I\end{aligned} \tag{17}$$

Also I is the 0 and $1 + I$ the unit of R/I. [11, p 101]

```
locale quotient_ring_wrt_ideal = ring +
  fixes I assumes ideal: "ideal I R"
begin
definition "Cong = {(a, b). a - b ∈ I}"
end
sublocale quotient_ring_wrt_ideal ⊆
  quotient_ring R addition zero multiplication one Cong
  where "class_addition = (λX ∈ Quot. λY ∈ Quot.
      THE Z. ∃a ∈ X. ∃b ∈ Y. Z = a + b +| I)"
    and "class_zero = I"
    and "class_multiplication = (λX ∈ Quot. λY ∈ Quot.
      THE Z. ∃a ∈ X. ∃b ∈ b. Z = a · b +| I)"
    and "class_one = 1 +| I"
  ⟨proof⟩
```

This completes the translation.

4 Discussion

Jacobson's exposition of basic algebra is highly polished. Identifying the sections that lead to factor group and quotient monoid with respect to an ideal respectively required scanning a fairly large body of mathematical text that stretches over about 100 pages, and in which the foundations for much more group and ring theory are layed out than what is considered here.

4.1 Reuse in the Original Text

As an experienced author Jacobson takes care to introduce notions in a manner that facilitates reuse and avoids repetition. Monoid and group are the foundations for many concepts, and group is defined as an extension of monoid. Likewise congruence on a monoid is based on equivalence and is extended to congruence on a group. The quotient monoid induced by a congruence is identified and similarly the quotient group. The factor group is obtained from the quotient group of the congruence implied by a normal subgroup.

The concepts around rings are derived from the related concepts for the additive group and multiplicative monoid. This concerns the definition of ring itself and the congruence on a ring. As for groups the quotient ring induced by a congruence is identified. The definition of the congruence relation $\{(a, b)|a -$

$b \in I\}$ from the ideal I is seemingly a duplication of the related definition $\{(a, b)|a^{-1}b \in K\}$ for groups, but in fact the underlying concepts of normal subgroup and ideal differ.

4.2 Adequacy of the Translation

The main motivation for translating this corpus of mathematics to a formal proof document in Isabelle is to understand whether the reuses employed by Jacobson could be reflected with locales in an adequate fashion. Of course, we are not the first to formalise these particular pieces of group and ring theory. Incidentally, already Kammüller [12] used them as examples in his initial investigations of locales.[3] In principle, adequacy concerns definitions, theorem statements and proofs. In the current work, though, only definitions and theorem statements are considered.

Definitions of algebraic structures (with associated derived operations and theorems) are translated to locale declarations. Inheritance is achieved through imported locale expressions. Other definitions such as those of subgroups, normal subgrups or ideals simply become definitions in appropriate contexts. Theorems that relate algebraic structures to each other are expressed with sublocale declarations. The quotient structures are notable examples. The morphisms available with locales are sufficiently expressive for declaring dependencies to the contained structures in `quotient_monoid`, `quotient_group` and `quotient_ring`. Likewise for the locales `factor_group` and `quotient_ring_wrt_ideal`. All in all the formal text contains no repetitions that would not also be present in the informal text.[4]

4.3 Further Observations

Occasionally, the choice in the informal text as to how a certain concept be introduced is surprising. As one example, subgroups are considered substructures of monoids not groups [11, p 31]. Clearly, utility of a definition is given priority over uniformity. Another example, which is not part of the study in Section 3, concerns homomorphisms on monoids. Jacobson gives this definition:

> **Definition 1.6** *If M and M' are monoids, then a map η of M into M' is called a* homomorphism *if*
>
> $$\eta(ab) = \eta(a)\eta(b), \quad \eta(1) = 1', \quad a, b \in M.$$
>
> If M' is a group the second condition is superfluous. For, if the first holds, we have $\eta(1) = \eta(1^2) = \eta(1)^2$ and multiplying by $\eta(1)^{-1}$ we obtain $1' = \eta(1)$. [11, p 58f]

[3] But he did not have an analog to the sublocale command available.

[4] The formal text is almost as succinct as the informal text, but while we assume that repetitions will be also avoided for proofs, we do not expect the succinctness to carry over.

The second part is easily overlooked, as, so it seems, by authors involved in the MathScheme project, who in the context of a vector space homomorphism h conjecture that $h(0) = 0'$ and similar properties are "obvious" axioms frequently omitted by textbooks [5, Section 4.4], while in fact it is implied by the target structure being an additive group. With locales the definition can be translated thus:

```
locale monoid_map = map η M M' +
  source: monoid M composition unit +
  target: monoid M' composition' unit'
  for η and M and composition (infix "·" 70) and unit ("1")
    and M' and composition' (infix "·''" 70) and unit' ("1''")

locale monoid_homomorphism = monoid_map +
  assumes "⟦a ∈ M; b ∈ M⟧ ⟹ η a ·' η b = η (a · b)"
    and "η 1 = 1'"

locale monoid_homomorphism' = monoid_map +
  target: group M' composition' unit' +
  assumes "⟦a ∈ M; b ∈ M⟧ ⟹ η a ·' η b = η (a · b)"
```

The sublocale command then enables relating these locales to each other:

```
sublocale monoid_homomorphism' ⊆ monoid_homomorphism ⟨proof⟩
```

That it, it is shown that the left hand side implies the right.

5 Conclusion

Translating informal mathematics from a textbook to a formal document processable by a mechanical reasoning system turned out to be an insightful exercise. Even without producing proofs it required studying the text with scrutiny, forcing attention to both detail and high-level connections.

We found that we could express the relations between algebraic structures in Jacobson's text well, both in definitions and theorems. This is the case even where constructions are not uniform — for example, subgroups being defined for monoids. Linguistic features that identify modularity do not stand out in the text, except that algebraic structures are denoted by tuples. The amount of mathematics that was formalised here is very small, but sizable formalisations have been based on locales, for example a proof of the Basic Perturbation Lemma in homological algebra [1].

Of course, proof assistants other than Isabelle provide means for modular reasoning as well and have been used for formalising abstract algebra. The most remarkable is perhaps the Mizar system and its library, abstracts of which have been published since 1990 in *Formalized Mathematics*, an online journal which to date has published twenty volumes of mechanically checked mathematics of various areas [7]. The system's means of abstraction have never been related to mainstream techniques such as functors but some literature is available [14,15].

For Coq there are two efforts towards modular reasoning: its module system, which is modelled around ML functors [16], and work of a group around Gonthier, who recently have completed a mechanical proof of Feit and Thompson's theorem on the solvability of groups of odd order [6] based Gonthier's contributed library Ssreflect [9]. The publication of this work appears to be imminent.

In spirit our exercise follows Avigad's reflections on understanding proofs [2], in particular Section 12.7, which is focussed on algebraic reasoning. While Avigad suggests that proof languages of mechanical systems can help better understand the notion of proof in mathematics, we use the informal text as a benchmark for a (declarative) scripting language that enables a prover to produce formal derivations

References

1. J. Aransay, C. Ballarin, and J. Rubio. A mechanized proof of the basic perturbation lemma. *Journal of Automated Reasoning*, 40(4):271–292, 2008.
2. J. Avigad. Understanding proofs. In P. Mancosu, editor, *The philosophy of mathematical practice*, chapter 12, pages 302–316. Oxford University Press, 2008.
3. C. Ballarin. Tutorial to locales and locale interpretation. In L. Lambán, A. Romero, and J. Rubio, editors, *Contribuciones Científicas en honor de Mirian Andrés Gómez*. Servicio de Publicaciones de la Universidad de La Rioja, Logroño, Spain, 2010. Also part of the Isabelle user documentation.
4. C. Ballarin. Locales: A module system for mathematical theories. *Journal of Automated Reasoning*, 2013. Online version; to appear in print.
5. J. Carette, W. M. Farmer, F. Jeremic, V. Maccio, R. O'Connor, and Q. M. Tran. The MathScheme library: Some preliminary experiments. Manuscript arXiv:1106.1862v1, 2011.
6. W. Feit and J. G. Thompson. Solvability of groups of odd order. *Pacific Journal of Mathematics*, 13(3):775–1029, 1963.
7. Formalized mathematics. Internet journal, http://fm.mizar.org/, 1990.
8. J. A. Goguen and R. M. Burstall. Institutions: abstract model theory for specification and programming. *J. ACM*, 39(1):95–146, 1992.
9. G. Gonthier and S. Le Roux. An Ssreflect tutorial. Technical Report RT-0367, INRIA, 2009.
10. F. Haftmann and M. Wenzel. Local theory specifications in Isabelle/Isar. In S. Berardi, F. Damiani, and U. de'Liguoro, editors, *Types for Proofs and Programs, TYPES 2008, Torino, Italy*, LNCS 5497, pages 153–168. Springer, 2009.
11. N. Jacobson. *Basic Algebra*, volume I. Freeman, 2nd edition, 1985.
12. F. Kammüller. *Modular Reasoning in Isabelle*. PhD thesis, University of Cambridge, Computer Laboratory, 1999. Also Technical Report No. 470.
13. W. Naraschewski and M. Wenzel. Object-oriented verification based on record subtyping in higher-order logic. In *Theorem Proving in Higher Order Logics*, LNCS 1479, pages 349–366. Springer, 1998.
14. P. Rudnicki and A. Trybulec. Mathematical knowledge management in MIZAR. In B. Buchberger and O. Caprotti, editors, *Mathematical Knowledge Management*, 2001.

15. C. Schwarzweller. Mizar attributes: A technique to encode mathematical knowledge into type systems. *Studies in Logic, Grammar and Rhetoric*, 10(23):387–400, 2007.
16. E. Soubiran. *Modular development of theories and name-space management for the Coq proof assistant*. PhD thesis, École Polytechnique, 2012.
17. M. Wenzel. Isabelle/Isar—a generic framework for human-readable proof documents. *Studies in Logic, Grammar and Rhetoric*, 10(23):277–298, 2007. Festschrift in Honour of Andrzej Trybulec.
18. M. Wenzel. *The Isabelle/Isar Reference Manual*, 2013. Revision for Isabelle 2013.