Gröbner Bases — Theory Refinement in the Mizar System

Christoph Schwarzweller

Department of Computer Science, University of Gdańsk, Gdańsk, Poland
schwarzw@math.univ.gda.pl

Abstract. We argue that for building mathematical knowledge repositories a broad development of theories is of major importance. Organizing mathematical knowledge in theories is an obvious approach to cope with the immense number of topics, definitions, theorems, and proofs in a general repository that is not restricted to a special field. However, concrete mathematical objects are often reinterpreted as special instances of a general theory, in this way reusing and refining existing developments. We believe that in order to become widely accepted mathematical knowledge management systems have to adopt this flexibility and to provide collections of well-developed theories.
As an example we describe the Mizar development of the theory of Gröbner bases, a theory which is built upon the theory of polynomials, ring (ideal) theory, and the theory of rewriting systems. Here, polynomials are considered both as ring elements and elements of rewriting systems. Both theories (and polynomials) already have been formalized in Mizar and are therefore refined and reused. Our work also includes a number of theorems that, to our knowledge, have been proved mechanically for the first time.

1 Introduction

One major goal of mathematical knowledge management is to design and construct large repositories containing a wide range of different topics, such as algebra, analysis, topology and many more. To be as broad as possible seems reasonable in order to explore the use of such repositories for distributing mathematics over the internet and extracting introductory courses, among others. On the other hand, to be attractive for professional mathematicians also, more advanced mathematics must be taken into account. As has been pointed out at the last MKM-meetings by Andrzej Trybulec "We should try to reach the research frontier".

Advanced, contemporary mathematics, however, cannot be brought onto the computer by simply choosing one theory and formalizing it "to its end". More advanced mathematics usually uses a number of theories to develop its results. Different theories are reused or combined to get new ones. Moreover, modern mathematics lives from the fact that one and the same object can be considered as a special instance of different theories. For example the integers can be considered as a group (generated by 1), as an Euclidean ring, as an ordered domain

or even as (the ring of) coefficients for polynomials rings. In each case the instantiation, or refinement as we shall also call it, of the general theory with the integers allows for both reusing results of the general theory and deducing new results for the particular case.

We believe that mathematical repositories should reflect this way of "working" with mathematical theories. Continuing the work of [GS04] where combination of theories has been investigated, we focus in this paper on theory refinement in the Mizar system [Miz05,RT01]. We consider the theory of Gröbner bases [Buc98] as an example. Gröbner bases are a method to decide among other things the ideal membership problem in polynomial rings: Via computing normal forms of polynomials with respect to a given ideal — a reduction in the sense of rewriting systems — ideal membership can be decided by syntactic equality, if the polynomials generating the ideal form a Gröbner base. We thus have polynomials as basic objects, usually defined as lists of elements from a coefficient ring or as functions from terms into a coefficient ring. Note that the definition of polynomials already uses a theory, the theory of rings. In the theory of Gröbner bases, however, polynomials are also used as special elements for different, more general theories:

1. Polynomials are considered as elements of a ring, that is addition and multiplication of polynomials coincide with ring addition and multiplication.
2. Polynomials are considered as elements of ideals, that is, though almost trivial, polynomials coincide with elements of sets while still obeying their addition and multiplication.
3. Polynomials are considered as elements of a relation, the reduction relation, that is polynomials coincide with the elements of relations.

Not taking into account the second item from above, we thus get the theory structure illustrated in figure 1. Of course one can define polynomials and all the concepts necessary for Gröbner bases from scratch without even employing theories for rings and reduction systems (see for example [The01]), but in repositories for mathematical knowledge management we should — if possible — build new theories by reusing and refining older ones.

The plan of the paper is as follows. After an introduction to the Mizar language we briefly recall polynomials, rings, ideals, and rewriting systems by reviewing their Mizar formalization as done in [RT99,BRS00,Ban95]. Section 4 and 5 describe the development of Gröbner bases based on these theories. In the last two sections we discuss the Mizar approach for refining and reusing theories and compare it to other approaches in the literature. Conclusions for the design of mathematical knowledge repositories are also drawn.

2 The Mizar System

Mizar's [RT01,Miz05] logical basis is classical first order logic extended with so-called schemes. Schemes allow for free second order variables, in this way

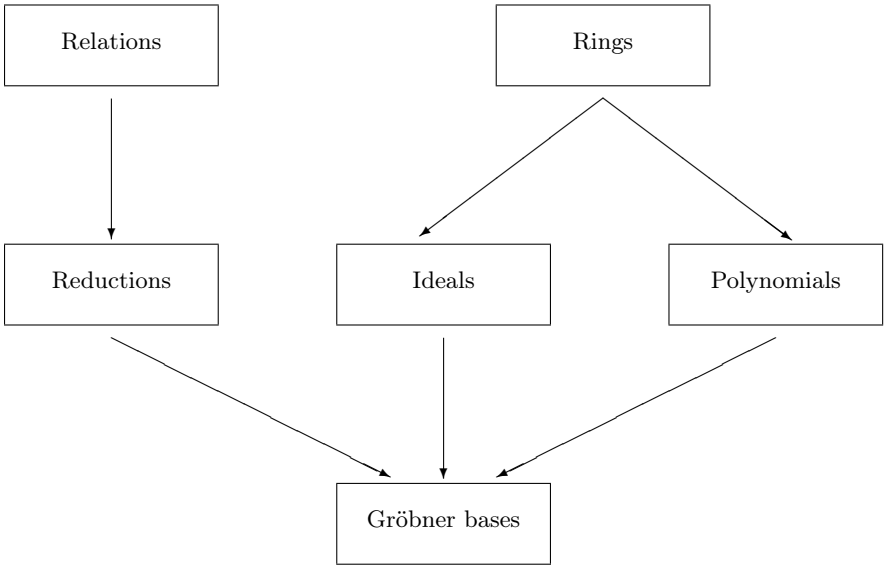

Fig. 1. Theory structure for the development of Gröbner bases

enabling, for example, the definition of induction schemes. The current development of the Mizar Mathematical Library (MML) is based on Tarski-Grothendieck set theory — a variant of Zermelo Fraenkel set theory using Tarski's axiom on arbitrarily large, strongly inaccessible cardinals [Tar39] which can be used to prove the axiom of choice —, though in principle the Mizar language allows for other axiom systems also. Mizar proofs are written in natural deduction style similar to the calculus of [Jaś34]. The rules of the calculus are connected with corresponding (English) natural language phrases so that the Mizar language is close to the one used in mathematical textbooks. The Mizar proof checker verifies the individual proof steps using the notion of obvious inferences [Dav81] to shorten the rather long proofs of pure natural deduction.

Mizar objects are typed, the types forming a hierarchy with the fundamental type `set` [Ban03]. New types are constructed using type constructors called modes. Modes can be decorated with adjectives — given by so-called attribute definitions — in this way extending the type hierarchy: For example, given the mode `Ring` and an attribute `commutative` a new mode `commutative Ring` can be constructed, which obeys all the properties given by the mode `Ring` plus the ones stated by the attribute `commutative`. Furthermore, a variable of type `commutative Ring` then is also of type `Ring`, which implies that all notions defined for `Ring` are available for `commutative Ring`. In addition all theorems proved for type `Ring` are applicable for objects of type `commutative Ring`; indeed the Mizar checker itself infers subtype relations in order to check whether theorems are applicable for a given type.

3 Polynomials, Rings, Ideals, and Rewriting Systems

In this section we briefly review the theories used to define Gröbner bases — polynomials, rings and ideals, and reduction systems. The main purpose is to present the basics of their Mizar formalization needed later to develop the theory of Gröbner bases.

3.1 Mizar Formalization of Rings and Ideals

In Mizar rings, or more generally algebraic domains, are defined as attributed structures, see [RST01]. That is, based on a structure mode giving carriers and operations of the domain properties of these operations, e.g. commutativity of addition or multiplication, are introduced by Mizar attributes. For rings the underlying structure mode is called `doubleLoopStr`:

```
struct (LoopStr, multLoopStr_0) doubleLoopStr
  (# carrier     -> set,
     add, mult   -> BinOp of the carrier,
     unity, Zero -> Element of the carrier #);
```

and the mode `Ring` is nothing else than this structure mode decorated with attributes describing the ring axioms. Note that `doubleLoopStr` is a descendant of two other structure modes `LoopStr` and `multLoopStr_0`, and thus a subtype of these. Now given a subset `F` of a structure mode `L` one can easily define an attribute describing that `F` is closed with respect to addition, left- and right-multiplication, for example

```
definition
let L be non empty LoopStr, F be Subset of L;
attr F is add-closed means
  for x,y being Element of L st x in F & y in F holds x+y in F;
end;
```

Combination of these properties then gives the definition of ideals. Note that using the already defined attributes it is trivial to additionally define left and right ideals. Also ideals generated by a subset `F` can be easily defined as a functor from subsets of the domain into ideals.

```
definition
let L be non empty doubleLoopStr;
mode Ideal of L is
  add-closed left-ideal right-ideal (non empty Subset of L);
end;

definition
let L be non empty doubleLoopStr, F be Subset of L;
assume F is non empty;
func F-Ideal -> Ideal of L means
  F c= it & for I being Ideal of L st F c= I holds it c= I;
end;
```

Note that both definitions actually do not use the mode `Ring`, but only the underlying structure mode `doubleLoopStr`: The existence of ideals and generated ideals does not depend on algebraic properties of the ring (just take the whole domain). Hence such objects should be defined without using these. Nevertheless, due to Mizar's type system, the so-defined notions of (and theorems proved for) ideals are available for rings in Mizar, because the mode `Ring` is based on and thus a subtype of `doubleLoopStr`.

3.2 Mizar Formalization of Polynomials

Polynomials are defined as functions from terms (called `bags` in Mizar) into coefficients, see [RST01,RT99]. Thus a polynomial is a subytype of `Series of n,L`, where `n` gives the number of indeterminates used to build terms and `L` the structure mode describing the coefficients. The attribute `finite-Support` ensures that the `Support` of a polynomial, that is the set of terms with a non zero coefficient, is finite.

```
definition
let n be Ordinal, L be non empty ZeroStr;
mode Polynomial of n,L is finite-Support Series of n,L;
end;
```

In the theory of Gröbner bases terms are assumed to be ordered. In Mizar we can use the mode `Order of X`, where `X` is an arbitrary set to do so. This will enable us to develop Gröbner bases for arbitrary orderings on terms and is actually another example for reusing theories in Mizar: The set of all terms for a given set of variables — remember that `n` gives the number of variables — is denoted by `Bags n`. Thus we get term orders, that is orders on terms, by just defining

```
definition
let n be set;
mode TermOrder of n is Order of Bags n;
end;
```

A term order is admissible, if the empty term — in Mizar denoted by `Empty Bag n` — is the smallest one and the order respects multiplication of terms, that is if we have for all terms t, t_1, t_2 both `Empty Bag n` $\leq t$ and $t_1 \leq t_2$ implies $t_1 \cdot t \leq t_2 \cdot t$. In Mizar this can be straightforwardly formalized as an attribute `admissible` for the mode `TermOrder`. Thus the Mizar type `admissible TermOrder` describes arbitrary admissible term orders. Note, that an admissible term order is well-founded.

Given an order `T` on terms, the head term of a polynomial `p` is the biggest term in `Support p` with respect to `T`; head coefficients and head monomials are defined analogously. In order to develop Gröbner bases for arbitrary term orders, we defined functors `HT(p,T)`, `HC(p,T)` and `HM(p,T)` taking the (admissible) term order `T` as an additional argument. Note that in order to define head terms the order `T` must be total (called `connected` in Mizar), but not admissible.

3.3 Mizar Formalization of Rewriting Systems

Rewriting systems are an approach to handle structures defined via equivalence relations. The idea is to decide a given equivalence by computing (unique) normal forms for the objects of concern: The objects are reduced until no more rewrite rules are applicable. Then, if the set of rules is "suitable", deciding the equivalence relaton is no more than syntactical comparison. Rewriting systems have various applications in such different fields as specification and verification, algebraic computation or pure theorem proving, see [DJ90]. In the following we recall the basic definitions of general rewriting systems as defined in [Ban95]. Given a relation `R` a reduction sequence is a sequence over `R` in which each two neighboured elements are in `R`. Thus the theory of rewriting systems actually is an extension of the theory of relations.

```
definition
let R be Relation;
mode RedSequence of R -> FinSequence means
  len it > 0 &
  for i being Nat st i in dom it & i+1 in dom it
  holds [it.i, it.(i+1)] in R;
end;
```

Then we have that `a` can be reduced to `b` (`R reduces a,b`), if there exists a reduction sequence of `R` with `a` being the first and `b` the last element in the sequence. Based on these definitions it is straightforward to introduce other basic concepts of rewriting systems, such as confluence, local confluence or the Church-Rosser property, for example

```
definition
let R be Relation;
attr R is locally-confluent means
  for a,b,c being set st [a,b] in R & [a,c] in R
  holds b,c are_convergent_wrt R;
end;
```

where `are_convergent_wrt` means that there exits an element `d` $\in$ `R` such that both `b` and `c` can be reduced to `d`. Termination properties such as `strongly-` and `weakly-normalizing` are also introduced in [Ban95]. Finally, a complete rewriting system is a strongly-terminating confluent one, that is a rewriting system in which in particular for every element a unique normal form exists.

4 Reinterpreting Polynomials

In this section we describe how the general theories of rings and rewriting systems are refined with polynomials. This allows to use notations and theorems of these theories for the special case of polynomials as well as further properties of polynomials themselves when later developing the theory Gröbner bases. Note that no special care has to be taken for ideals of polynomials; these are given automatically because ideals have been defined for general rings.

4.1 Polynomials as Rings

To get the theory of rings available for polynomials (the domain and) the operations of polynomials have to be interpreted as the ring (domain and) operations. This is done by defining a functor `Polynom-Ring(n,L)` into the underlying structure mode of rings (see [RT99]), called `doubleLoopStr` in Mizar. Here, `n` gives the number of indeterminates and `L` the structure mode describing the coefficient domain; note that the number of indeterminates need not be finite. In the definition the components of the structure, that is the domain and the operations of the ring, are simply identified with polynomials and the corresponding operations on polynomials.

```
definition
let n be Ordinal,
    L be right_zeroed add-associative right_complementable
         unital distributive non trivial (non empty doubleLoopStr);
func Polynom-Ring(n,L) -> strict non empty doubleLoopStr means
 (for x being set holds
   x in the carrier of it iff x is Polynomial of n, L) &
 (for x,y being Element of it, p,q being Polynomial of n, L
   st x = p & y = q holds x+y = p+q) &
 (for x,y being Element of it, p,q being Polynomial of n, L
   st x = p & y = q holds x*y = p*'q) &
 0.it = 0_(n,L) & 1_ it = 1_(n,L);
end;
```

Now, to apply theorems proved for general rings we need not only that `Polynom-Ring(n,L)` is a `doubeLoopStr`, but also that the attributes establishing the type `Ring` hold. It turns out that for different attributes to hold different properties of the coefficient domain are necessary. Therefore each attribute is proved in a cluster registration stating exactly the properties of the coefficient ring necessary to prove it (see [RT99]), for example

```
registration
let n be Ordinal,
    L be Abelian right_zeroed add-associative right_complementable
         unital distributive non trivial (non empty doubleLoopStr);
cluster Polynom-Ring(n,L) -> Abelian;
end;
```

The effect is that properties of `Polynom-Ring(n,L)` are automatically bound to properties of the coefficient domain `L`: If `L` obeys the properties stated in the registration, the Mizar checker itself infers that `Polynom-Ring(n,L)` has the concluding property, hence is a subtype of `Ring`. This supports reusing theorems of the general ring theory for the special case of polynomials.

4.2 Polynomial Reduction

Polynomial reduction establishes a generalization of polynomial division for the univariate case: Each reduction step describes a single step in the division process. Thus a non-zero polynomial `f` reduces to a polynomial `g` using polynomial

p by eleminating term (bag) t if there exists a term s such that s · HT(p,T) = t and g = (f - f.t/HC(p,T)) * s *' p, which in Mizar can be easily defined as a predicate f reduces_to g,p,b,T. Note that the reduction depends on the term order T used to define head terms and head monomials.

Now, to introduce polynomial reduction as a special case of general rewriting we have to define the reduction relation, that is the relation R which contains all pairs (p_1, p_2) such that p_1 reduces (in one step) to p_2 with respect to a given set of polynomials P. This is done with a functor PolyRedRel(P,T) returning an object of type Relation of. Note that the ring of polynomials which is available according to section 4.1 is used to describe the domain of the relation.

```
definition
let n be Ordinal, T be connected TermOrder of n,
    L be Field, P be Subset of Polynom-Ring(n,L);
func PolyRedRel(P,T) ->
  Relation of (the carrier of Polynom-Ring(n,L)) \ {0_(n,L)},
              the carrier of Polynom-Ring(n,L) means
for p,q being Polynomial of n,L holds [p,q] in it iff p reduces_to q,P,T;
end;
```

Now, PolyRedRel(P,T) being of type Relation — the type Relation of widens to Relation — allows reuse of the whole theory, that is both notations and theorems developed for rewriting systems, for polynomial reductions. For example, to show that polynomial reduction is terminating we just use the attribute strongly-terminating defined for arbitrary reduction systems and prove in a cluster registration that PolyRedRel(P,T) fulfils it for an arbitrary set P of polynomials and an arbitrary term order T:

```
registration
let n be Nat, T be connected admissible TermOrder of n,
    L be Field, P be Subset of Polynom-Ring(n,L);
cluster PolyRedRel(P,T) -> strongly-normalizing;
end;
```

Also, showing that polynomial reduction with respect to a set of polynomials P describes the congruence given by the ideal generated by P — a necessary precondition to decide ideal membership with reduction techniques — needs no further preparations: The reflexive symmetric transitive closure of the reduction relation is given by the predicate are_convertible_wrt from rewriting theory, whereas generated ideals and their congruences — the functor P-Ideal and the predicate are_congruent_mod — are reused from general ideal theory. We thus get the following

```
theorem
for n being Nat, T being admissible connected TermOrder of n,
    L being Field, P being non empty Subset of Polynom-Ring(n,L),
    f,g being Element of Polynom-Ring(n,L)
holds f,g are_congruent_mod P-Ideal
      iff f,g are_convertible_wrt PolyRedRel(P,T);
```

5 Mizar Formalization of Gröbner Bases

We start with a brief introduction to Gröbner bases, see also [BW93,CLO'S96]. Let $K[X_1, \dots X_n]$ be the ring of polynomials over a field K with n indeterminates. For $P \subseteq K[X_1, \dots X_n]$ the ideal generated by P — the minimal ideal including P — is given by $<P> = \{\sum_{i=0}^{n} f_i \cdot p_i \mid f_i \in K[X_1, \dots X_n],\ p_i \in P\}$. The basic problem that can be algorithmically solved using Gröbner bases is the following: Given $f \in K[X_1, \dots X_n]$ and $P \subseteq K[X_1, \dots X_n]$, does $f \in <P>$ hold? Denoting the reduction for polynomials introduced in section 4.3 by $\longrightarrow_P$ it is easy to show that $f \xrightarrow{*}_P 0$ implies $f \in <P>$. The other direction, however, does not hold in general and can actually serve as a definition for Gröbner bases. In our formalization we use the equivalent definition, that $G \subseteq K[X_1, \dots X_n]$ is a Gröbner base if and only if $\longrightarrow_G$ is locally confluent. Note again, that $\longrightarrow_G$ is terminating.

To check whether a given (finite) set $P \subseteq K[X_1, \dots X_n]$ is a Gröbner base it is sufficient to consider the (finite set of) s-polynomials generated by P, that is

$$\mathrm{spoly}(p_1, p_2) \ = \ \mathrm{HC}(p_2) \cdot \frac{t}{\mathrm{HT}(p_1)} \cdot p_1 \ - \ \mathrm{HC}(p_1) \cdot \frac{t}{\mathrm{HT}(p_2)} \cdot p_2$$

where $t = lcm(\mathrm{HT}(p_1), \mathrm{HT}(p_2))$ for all $p_1, p_2 \in P$: G is a Gröbner base if we have $\mathrm{spoly}(p_1, p_2) \xrightarrow{*}_G 0$ for all $p_1, p_2 \in P$, which in the view of general rewriting can be interpreted as checking critical pairs. This gives rise to a completion algorithm: If $\mathrm{spoly}(p_1, p_2)$ not reduces to 0, its normal form is added to P — note that $p_1, p_2 \in P$ implies $\mathrm{spoly}(p_1, p_2) \in <P>$, so that the generated ideal $<P>$ is not changed — and s-polynomials are recursively computed. This is the basic version of Buchberger's Algorithm transforming a set $P \subseteq K[X_1, \dots X_n]$ into a set $G \subseteq K[X_1, \dots X_n]$ such that $<P> = <G>$ and $\longrightarrow_G$ is locally confluent. Further investigations and improvements of the algorithm can be found in the literature (see for example [Buc79]).

In the following we present the main results of our formalization in Mizar so far. Besides the definition of Gröbner bases and the usual characterization using s-polynomials, we also considered other characterizations and the existence of both ordinary and reduced Gröbner bases.

5.1 Definition and Characterizations

A Gröbner base for a given ideal `I` is a set `G` of polynomials such that the induced reduction relation `PolyRedRel(G,T)` is locally confluent (hence a complete rewriting system) and the ideal generated by `G` equals `I`. Note again, that the term order `T` is a parameter of the reduction relation.

```
definition
let n be Ordinal, T be connected TermOrder of n,
    L be Field, G,I be Subset of Polynom-Ring(n,L);
pred G is_Groebner_basis_of I,T means
  G-Ideal = I & PolyRedRel(G,T) is locally-confluent;
end;
```

We proved a number of further characterizations of Gröbner bases from [BW93], so for example, that `G` is a Gröbner base if each polynomial in `G-Ideal` is top-reducible with respect to `G`, if each polynomial in `G-Ideal` is reducible to the zero polynomial `0_(n,L)` or if each head term of a polynomial in `G-Ideal` is divided by a head term of a polynomial in `G`. The main property of Gröbner bases `G` — ideal membership of a polynomial `p` is decidable by reducing `p` with respect to `G` — can be formulated as follows.

```
theorem
for n being Nat, T being connected admissible TermOrder of n,
    L being Field, p being Polynomial of n,L,
    G being non empty Subset of Polynom-Ring(n,L)
st G is_Groebner_basis_wrt T
holds p in G-Ideal iff PolyRedRel(G,T) reduces p,0_(n,L);
```

A completely different characterization of Gröbner bases, that is often used to prove more involved theorems, relies on so-called standard representations of polynomials [BW93]. A standard representation of a polynomial `p` with respect to a set of polynomials `P` is a linear combination $\mathtt{p} = \sum_{i=1}^{k} m_i p_i$ where the m_i are arbitrary monomials, the p_i are from the set `P` and the head terms of the $m_i p_i$ are bounded by `HT(p,T)`, or more general by a given term `t`. The concept of linear combinations again can be reused from ring theory [BRS00]:

```
definition
let L be non empty multLoopStr, S be non empty Subset of L;
mode LeftLinearCombination of S -> FinSequence of the carrier of L means
  for i being set st i in dom it
  ex u being Element of L, s being Element of S st it/.i = u * s;
end;
```

A standard representation of a polynomial `f` is then straightforwardly defined as a `LeftLinearCombination of P` with the two additional conditions from above. Now one can show that a set `G` is a Gröbner base if and only if there exists a standard representation for each polynomial in the ideal generated by `G`. Defining a predicate `f has_a_Standard_Representation_of G,t,T` with the obvious meaning we thus get

```
theorem
for n being Nat, T being connected admissible TermOrder of n,
    L being Field, G being non empty Subset of Polynom-Ring(n,L)
holds G is_Groebner_basis_wrt T
      iff for p being Polynomial of n,L st p in G-Ideal
          holds p has_a_Standard_Representation_of G,HT(f,T),T;
```

5.2 Construction of Gröbner Bases

The key point in the construction of Gröbner bases is the observation that there exists a finite test to check whether a set of polynomials `G` is locally confluent, hence a Gröbner base. Critical situations that have to be checked are

given by s-polynomials describing the "difference" between two polynomials p1 and p2. The Mizar definition is as follows. Note again that p1,p2 ∈ G implies S-Poly(p1,p2,T) ∈ G-Ideal.

```
definition
let n be Ordinal, T be connected TermOrder of n,
    L be Field, p1,p2 be Polynomial of n,L;
func S-Poly(p1,p2,T) -> Polynomial of n,L equals
  HC(p2,T) * (lcm(HT(p1,T),HT(p2,T))/HT(p1,T)) *' p1 -
  HC(p1,T) * (lcm(HT(p1,T),HT(p2,T))/HT(p2,T)) *' p2;
end;
```

Now, if for a given set G of polynomials we have that PolyRedRel(G,T) reduces S-Poly(g1,g2,T) to 0_(n,L) for all p1,p2 ∈ G, then G is a Gröbner base. Note that if G is finite there exist only finitely many s-polynomials. Using the transition lemma, basically stating that if a polynomial p1-p2 is reducible to 0_(n,L) with respect to G, then there exists a polynomial q such that PolyRedRel(G,T) reduces both p1 and p2 to q, we proved the following

```
theorem
for n being Nat, T being admissible connected TermOrder of n,
    L being Field, G being Subset of Polynom-Ring(n,L)
holds (for p1,p2 being Polynomial of n,L st p1 in G & p2 in G
       holds PolyRedRel(G,T) reduces S-Poly(p1,p2,T),0_(n,L))
implies G is_Groebner_basis_wrt T;
```

This theorem gives rise to a completion algorithm to compute Gröbner bases (see [Buc98]). Note, that the proof of the theorem's opposite direction is almost trivial using the characterizations from the section 5.1.

For the construction of Gröbner bases, however, not all s-polynomials need to be considered, hence detecting such s-polynomials saves a number of reductions in the construction process. In the literature theorems characterizing such situations can be found (see e.g. [Buc79]). We formalized a first theorem into this direction stating that s-polynomials of polynomials p1 and p2 with lcm(HT(p1,T),HT(p2,T)) = HT(p1,T) · HT(p2,T), in other words the head terms of p1 and p2 have no variables in common, need not be considered, they always reduce to the zero polynomial:

```
theorem
for n being Ordinal, T being connected admissible TermOrder of n,
    L being Field, p1,p2 being Polynomial of n,L
st HT(p1,T),HT(p2,T) are_disjoint
holds PolyRedRel({p1,p2},T) reduces S-Poly(p1,p2,T),0_(n,L);
```

5.3 Existence of Gröbner bases

Finally we consider the existence and uniqueness of Gröbner bases. It is a theoretically interesting fact that a finite Gröbner base exists for any given ideal

I; or from a rewriting point of view that there exists a (finite) completion for every set G of polynomials. Using the characterization of section 5.1 — that G is a Gröbner base if each head term of a polynomial in G-Ideal is divided by a head term of a polynomial in G — and Dickson's lemma from [LR02] it is easy to prove the following

```
theorem
for n being Nat, T being connected admissible TermOrder of n,
    L being Field, I being Ideal of Polynom-Ring(n,L)
ex G being finite Subset of Polynom-Ring(n,L)
st G is_Groebner_basis_of I,T;
```

which actually is another formulation (and another proof) of the Hilbert basis theorem. Note that the theorem states even more, namely that a Gröbner base for a given ideal exists for any total admissible term order T.

We also considered reduced Gröbner bases. In general a Gröbner base is of course not uniquely determined by the ideal I, even if we choose a fixed term order T. However, introducing the concept of reduced Gröbner bases, the situation looks different. A set G of polynomials is called reduced, if every p ∈ G is monic, that is HC(p,T) = 1 for all p ∈ G, and every p ∈ G is irreducible with respect to G\{p}. Note that only the second condition can be reused from rewriting theory, the other being a property of polynomials. Using a predicate is_reduced_wrt we proved the following theorems showing existence and uniqueness of reduced Gröbner bases.

```
theorem
for n being Nat, T being connected admissible TermOrder of n,
    L being Field, I being Ideal of Polynom-Ring(n,L) st I <> {0_(n,L)}
ex G being finite Subset of Polynom-Ring(n,L)
  st G is_Groebner_basis_of I,T & G is_reduced_wrt T;

theorem
for n being Nat, T being connected admissible TermOrder of n,
    L being Field, I being Ideal of Polynom-Ring(n,L),
    G1,G2 being non empty Subset of Polynom-Ring(n,L)
st G1 is_Groebner_basis_of I,T & G1 is_reduced_wrt T &
   G2 is_Groebner_basis_of I,T & G2 is_reduced_wrt T
holds G1 = G2;
```

6 Mathematical Knowledge Repositories

6.1 Mizar Mathematical Library

The Mizar Mathematical Library [Miz05] is a long term project that aims at developing both a comprehensive library of mathematical knowledge and a formal language for doing so. At the time of writing the library consists of 904 articles stating about 40000 theorems and 8000 definitions. Also because of the

huge number of covered areas Mizar is well-suited for our experiments concerning building up new developments on existing ones. In the following we discuss some issues of our formalization which we consider of general interest for the development of mathematical repositories.

As a first point, we want to stress that adopting notations is a crucial issue when building mathematical repositories. By "adopting" we mean not only reusing notations in a more specialized situation, but also slightly changing and extending these. We illustrate this with reduced sets of polynomials already mentioned in section 5.3. Irreducibility of sets stems from rewriting and can be defined as follows.

```
definition
let R be Relation, A be set;
pred A is_irreducible_wrt R means
  for a being Element of A holds a is_a_normal_form_wrt R;
end;
```

Reduced sets of polynomials, though based on this notion, are somewhat different: Each polynomial must be irreducible with respect to all other polynomials. Furthermore only monic polynomials are considered here. Hence, the predicate concerning reduction has not only to be refined but also to be extended. In Mizar this can be straightforwardly done as follows.

```
definition
let n be Ordinal, T be connected TermOrder of n,
    L be Field, P be Subset of Polynom-Ring(n,L);
pred P is_reduced_wrt T means
  for p being Polynomial of n,L st p in P
  holds p is_monic_wrt T & {p} is_irreducible_wrt PolyRedRel(P\{p},T);
end;
```

Note, that the reduction relation `PolyRedRel` is used with the argument `P\{p}` rather than `P`. We believe, that it is this flexibility that we need to built up large repositories covering not only few theories.

Equivalence proofs are often cyclic, that is actually given by a number of implications. Of course this can be easily mirrored in Mizar (or other repositories) by stating each implication as a theorem. However, using such equivalences then becomes rather tedious, because to get an equivalent formulation more than one theorem is necessary. In [BW93], for example, theorem 5.35 gives 10 equivalent characterizations of Gröbner bases, so that using these equivalences requires up to 9 theorems in Mizar. Here, it might be helpful to extend the language of mathematical repositories to also include "equivalence theorems".

A last point we want to mention concerns the general development of repositories. Most projects are concerned with the formalization of a particular theorem to illustrate the usability of a certain approach. Therefore, for obvious reasons, often parts or theorems actually belonging to the theory considered are ignored just because they are not really necessary to prove the goal. We believe that the

development of a general mathematical repository as necessary for mathematical knowledge management has to go another way: Often it turns out that the parts left out would enable a better development beyond the theorem originally chosen. So we should seek for completeness in the sense that when formalizing a theory we should bear alternative characterizations in mind. An example here are standard representations. Of course one can define Gröbner bases and their construction by s-polynomials without using standard representations. To prove more involved results on s-polynomials, however, one finds that in the literature often standard representations are used, which implies that the definition of standard representations cannot be left out.

6.2 Other Formalizations

Gröbner bases and polynomials have been defined in other systems; we first mention [The01], where Buchberger's algorithm is formalized using the Coq proof assistant. From this development an implementation of the algorithm in Ocaml has been extracted. In [MPAR04] a Common Lisp implementation of Buchberger's algorithm is presented that has been verified in Acl2. This is part of a larger project that aims at the computational formalization of polynomial algorithms in the spirit of combining computer algebra and theorem proving. Harrison [Har01] presents a Gröbner base algorithm for complex polynomials in HOL and uses it as a semi-decision procedure for polynomial equations in his work on quantifier elimination. Focusing on the algorithm, however, the notations of both rings and rewriting are not introduced, but defined from scratch for polynomials only.

C-CoRN [CGW04] also includes polynomials as a result of the "Fundamental Theroem of Algebra"-Project. Here a real number structure is used is used to develop polynomials in Coq so that an instantiation with a construction of the real numbers results in a full constructive proof. Polynomials have been defined in other repositories such as for example IMPS [FGT93] and Theorema [Buc01]. However, none of these approaches has been used to develop Gröbner bases so far. It would be interesting to do so and to compare these developments with our experiences in Mizar.

7 Conclusions and Further Work

We believe that mathematical repositories serves at least two goals. Firstly, of course, repositories form the basis for other Mathematical Knowledge Management activities by providing the knowledge to deal with. Secondly, it seems to us that mathematical repositories are also the key for attracting mathematicians and other users: The more knowledge we include in our repositories, the more likely will be the acceptance of both mathematical repositories and its attached software. Therefore developing mathematical repositories should

- be broadly based, that is a large number of different fields of mathematics has to be covered.

- be highly reusable and refinable to impress possible users how easily one can adopt existing developments, in particular basic theories.
- aim at describing (basic) theories as completely as possible — and not only at developing the proof a special theorem — in order to increase the number of possible users.

We also believe that such development techniques will lead to the formalization of contemporary mathematics easier just because a broad basic repository supports — and is necessary for — more involved mathematics.

In this paper we have presented a case study in Mizar to illustrate what such a broad theory development may look like. The Mizar type mechanism, especially the possibility to extend types with adjectives to describe additional properties, elegantly supports the refinement and reuse of existing developments and theories: Adjectives allow not only to refine theories as a whole. In addition theorems itself can be formulated using only properties, that is adjectives, necessary to prove them and can therefore be reused in every theory fulfilling these adjectives. Because this kind of reasoning is present in nearly all areas of mathematics, we claim that such a flexible type system is of major importance for the development of mathematical repositories.

The work presented in this paper can be continued in two ways. Firstly, the theory of Gröbner bases in Mizar should be further developed: Theorems concerning avoiding s-polynomials should be formalized, also to explore the use of "non-standard characterizations", here by standard representations. Also, in the spirit of section 6.1, the characterization of Gröbner bases by division with remainder and of course generalizations of the topic such as for example syzygies are of further interest. Secondly, it would be interesting to transform the formalized material into an introductionary course on Gröbner bases. The main point here would be to take into account both the underlying ring and rewrite theories and the proofs as they have been written in the Mizar language. We think that this would not only give insights in how to use Mizar for generating teaching material, but also — due to the number of theories involved — how to structure courses with larger number of prequisites.

References

[Ban95] G. Bancerek, Reduction Relations; Formalized Mathematics, 1995, available in JFM from [Miz05].

[Ban03] G. Bancerek, On the Structure of Mizar Types; in: H. Geuvers and F. Kamareddine (eds.), Proc. of MLC 2003, ENTCS 85(7), 2003.

[BW93] T. Becker and V. Weispfenning, Gröbner Bases — A Computational Approach to Commutative Algebra; Springer Verlag, 1993.

[BRS00] J. Backer, P. Rudnicki, and C. Schwarzweller, Ring Ideals; Formalized Mathematics, 2000, available in JFM from [Miz05].

[Buc79] B. Buchberger, A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Bases; in: Proceddings of Eurosam 79, Lecture Notes in Computer Science 72, pp. 3-21, 1979.

[Buc98] B. Buchberger, Introduction to Gröbner bases; in: B. Buchberger and F. Winkler (eds.), Gröbner Bases and Applications, pp. 3-31, Camebridge University Press, 1998.

[Buc01] B. Buchberger, Mathematical Knowledge Management in Theorema; in: B. Buchberger, O. Caprotti (eds.), Proceedings of the First International Workshop on Mathematical Knowledge Management, Linz, Austria, 2001.

[CGW04] L. Cruz-Filipe, H. Geuvers, and F. Wiedijk, C-CoRN, the Constructive Coq Repository at Nijmegen; `http://www.cs.kun.nl/~freek/notes/`.

[CLO'S96] D. Cox, J. Little, and D. O'Shea, Ideals, Varieties and Algorithms; Springer Verlag, New York, 2nd edition, 1996.

[Dav81] M. Davies, Obvious Logical Inferences; in: Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 530-531, 1981.

[DJ90] N. Dershowitz and J.P. Jounnaud, Rewrite Systems; in: J. van Leeuven (ed.), Formal Models and Semantics — Handbook of Theoretical Computer Science, vol. B, Elsevier, 1990.

[FGT92] W. Farmer, J. Guttman, and F. Thayer, Little Theories; in: D. Kapur (ed.), Automated Deduction – CADE-11, LNCS 607, pp. 567–581, 1992.

[FGT93] W. Farmer, J. Guttman, and F. Thayer, IMPS – An Interactive Mathematical Proof System; Journal of Automated Reasoning 11, pp. 213–248, 1993.

[GS04] A. Grabowski and C. Schwarzweller, Rough Concept Analysis — Theory Development in the Mizar System; in: G. Bancerek, A. Asperti, and A. Trybulec (eds.), Proceeding of the Third International Conference on Mathematical Knowledge Management, Lecture Notes in Computer Science 3119, pp. 130-145, 2004.

[Har01] J. Harrison, Complex Quantifier Elimination in HOL; in: supplementary Proceedings of the 14th International Conference on Theorem Proving in Higher Order Logic, pp. 159-174, 2001.

[Jaś34] S. Jaśkowski, On the Rules of Suppositon in Formal Logic; in: Studia Logica, vol. 1, 1934.

[LR02] G. Lee and Piotr Rudnicki, Dickson's Lemma; Formalized Mathematics, 2002, available in JFM from [Miz05].

[Miz05] The Mizar Home Page, `http://mizar.org`.

[MPAR04] J. Medina-Bulo, F. Paloma-Lozano, J. Alonzo-Jiménez, and J.-L. Ruiz-Reina, Verified Computer Algebra in Acl2: Gröbner bases; in: B. Buchberger and J. Campbell (eds.), 7th Conference on Artificial Intelligence and Symbolic Computation (AISC04), Lecture Notes in Computer Science 3249, pp. 171-184.

[RST01] P. Rudnicki, C. Schwarzweller, and A. Trybulec, Commutative Algebra in the Mizar System; in: Journal of Symbolic Computation vol. 32(1/2), pp. 143-169, 2001.

[RT99] P. Rudnicki and A. Trybulec, Multivariate polynomials with arbitrary number of variables; Formalized Mathematics, 1999, available in JFM from [Miz05].

[RT01] P. Rudnicki and A. Trybulec, Mathematical Knowledge Management in Mizar; in: B. Buchberger, O. Caprotti (eds.), Proc. of MKM 2001, Linz, Austria, 2001.

[Tar39] A. Tarski, On Well-Ordered Subsets of Any Set; in: Fundamenta Mathematicae, vol. 32, pp. 176-183, 1939.

[The01] L. Théry, A Machine-Checked Implementation of Buchberger's Algorithm; in: Journal of Automated Reasoning 26, pp. 107-137, 2001.