Completion and Invariant Theory in Symbolic Computation and Artificial Intelligence

Eugen E. Ardeleanu*

Research Institute for Symbolic Computation
Johannes Kepler University, Linz, A-4040 Austria
eardelea@risc.uni-linz.ac.at

Abstract. An outline for the study of invariant theoretic (as structural) and completion (as syntactical) concepts in symbolic computation and artificial intelligence is presented on a level of abstraction which permits a unifying viewpoint on problems in symbolic computation and artificial intelligence. We refer to applications in computational polynomial ideal theory and in general problem-solving in the sense of AI research.

1 Introduction

In this article we discuss two methodological principles commonly used in mathematics and computer science.

The first principle we refer to, integrates *structural methods* mainly used to provide for an *external characterization* of relational (algebraic) structures.

The second envisioned principle embodies *syntactical methods* usually in force when dealing with an *internal characterization* of equational theories.

Correspondingly, we will concentrate on two instances of these principles:

1. the *search for invariants* method, and
2. the *completion* method.

It is our opinion that both areas of symbolic computation and artificial intelligence could benefit from a combined structural-syntactical approach to some of their problems.

In Section 2 we succinctly present some methodological considerations on the role the two principles play in mathematics and computer science and on their interrelation. In Section 3 we concentrate on the completion method and its most pertinent application to the construction of standard (Gröbner) bases by Buchberger's completion algorithm in polynomial ideal theory, which, in turn, represents an important segment of the new emerging border area (between mathematics and computer science) of symbolic computation. In Section 4 we consider general problem-solving (from AI viewpoint) and stress the parallelism with symbolic computation in applying the two methods. An example is also included. Section 5 concludes the article.

* Supported by the Austrian Ministry for Science and Research.

2 Methodology

2.1 The Methodology of Structural Theories: Invariants, Representations, Completeness

The relation between *Invariants, Representation Theorems*, and *Completeness Theorems* appears frequently in algebra and logic.

Let **C** be an arbitrary category of algebraic structures (i.e objects and morphisms obeying the usual axioms). Consider in **C** a fixed universal construct $\mathcal{U}$, i.e an object and a set of distinguished morphisms (e.g sums, products), assuming, of course, it exists. Then, one is interested in the following problems formulated on a general level:

- Determine the structure of the objects in **C** using $\mathcal{U}$.
- Determine the objects with the simplest structure which are preserved under (a certain class of) morphisms, i.e the invariants.
- Determine the complete set of invariants with which one is capable of characterizing any object in **C**.

Example 1. The theory of modules (with the well known instantiations, e.g the theory of finitely generated abelian groups, the theory of linear transformations for finite dimensional vector spaces), cf. [1].

The theory of quadratic forms, cf. [2].

The algebraic theory of dynamical systems, cf. [3].

The representation theorem for boolean functions, cf. [4].

The concept of indexed systems, e.g logical fiberings, cf. [5], and indexed categories and related topics of increasing interest for the foundations of computer science, logic and mathematics.

The typical gender of results one would expect by following this method is illustrated by the following theorem of Birkhoff in the context of universal algebra, cf. [6]:

> Every universal algebra is a subdirect product of subdirectly irreducible algebras (the *structural* characterization of universal algebras).

2.2 The Methodology of Syntactical Theories

It is often the case in computer science that the *syntactical* approach is emphasized cf. [7], mainly because of immediate possibilities for applications and implementations, e.g equational theories (varieties in universal algebras defined by a set of equations) in algebraic specification of programming languages, rewriting systems (with instantiations as grammars, Thue systems, etc), unification (with implementations as Prolog).

The main problems in syntactical theories are not to determine the structure of the objects, but to develop (constructive) methods for "simplifying" them.

The process is known as *reduction to normal forms* and the *termination* and *uniqueness* of the result of reduction are the crucial points.

To prove termination the common technique consists of defining a total well ordering on the set of objects compatible with the reduction relation between the "more" and "less" complex objects, cf. [8].

To prove uniqueness comes down to prove that the reduction relation has (some variant of) the Church-Rosser property, which, roughly, ensures that the reduction to normal forms does not dependent on the different ways it can be performed, i.e normal forms are unique, cf. [8].

There have been also attempts to introduce a categorical viewpoint in dealing with syntactical theories; the objects are the well-formed expressions in the theory and the morphisms are the reduction rules. In this way some fundamental properties of syntactical theories correspond to the existence of universal categorical constructs, cf. [9].

Finally we refer again to an illustrative result of Birkhoff, cf. [6].

> A class $\mathbf{C}$ of universal algebras is a variety (or equational class) if and only if it has the following closure properties (the *internal* characterization of varieties):
> If $A \in \mathbf{C}$, then every subalgebra of A is in $\mathbf{C}$;
> If $A \in \mathbf{C}$, then every homomorphic image of A is in $\mathbf{C}$;
> If $\{A_\alpha : \alpha \in I\} \subseteq \mathbf{C}$, then the product $\Pi A_\alpha \in \mathbf{C}$.

3 The Completion Principle

The general method of designing algorithms by completion (for a survey see [10]) is of ancient vintage in algebra. It is used nowadays in symbolic computation as a basic tool. The most famous instance is Buchberger's algorithm for "completion" to standard (Gröbner) bases in polynomial ideal theory. We will first briefly describe the problem and the idea of the method in this context (for the exact definitions and proofs see [11]).

Definition 1 *The ideal membership problem:*
Given a polynomial, say $f \in \mathbb{R}[x_1, ..., x_n]$ and a finite set of polynomials $S = \{f_1, ..., f_m\} \subset \mathbb{R}[x_1, ..., x_m]$, decide whether f belongs to the ideal generated by S, $f \in I[S]$.

The natural idea to solve this problem is to start to divide f by polynomials f_i, $f = p_i f_i + r$ (we say that f *reduces* to r by f_i, in symbols $f \to_{f_i} r$) and continue like in Euclid's algorithm (in the case of univariate polynomials it *is* the Euclidean algorithm), with r as f. We can view this as a reduction process and we have to ensure termination and uniqueness. Termination is ensured by introducing a total ordering $<$ on $\mathbf{R}[x_1, ..., x_n]$, (e.g a graded lexicographical ordering) compatible with the reduction relation. Thus, if f reduces to r, then $r < f$. It turns out that this ordering is noetherian, by Hilbert's basis theorem.

Definition 2 *We say that f reduces to f' with respect to S, $f \rightarrow_S f'$ if and only if there exist $f_i, ..., f_j \in S$ such that*

$$f \rightarrow_{f_i} \dots \rightarrow_{f_j} f'.$$

It is clear that, if $f \rightarrow_S 0$, then $f \in I[S]$. However, the converse is in general not true. For the converse to hold, we have to "complete" the set S to a so-called *standard (Gröbner) base*, first introduced by Buchberger, cf. [11]. If no polynomial in this set can be reduced via some other polynomial in the set, we have a *reduced* standard base. Motivated by the following Theorem 1, we will take the liberty of introducing some new terminology: we will call standard (Gröbner) bases *complete generating sets* and reduced standard (Gröbner) bases, *complete bases.*

There are many (syntactical) characterizations for the complete sets, cf. [12]; among others, the one stating that S is a complete set if and only if every polynomial f reduces to a unique normal form (i.e a polynomial f' which cannot be further reduced) with respect to S.

Definition 3 *A set of polynomials S is a complete set if one of the following equivalent conditions is fulfilled:*
The relation $\rightarrow_S$ has the Church-Rosser property.
For all $f_i, f_j \in S$, $spoly(f_i, f_j) \rightarrow_S 0$.

(Here $spoly(f_i, f_j)$ is the *substraction polynomial* of f_i and f_j defined as the difference between the polynomials obtained by reducing the least common multiple of the leading power products of the two polynomials with respect to f_i and f_j.) The equivalence of the conditions in Definition 3. is known as the Buchberger's theorem. This result makes it possible to design an algorithm which accepts as input S' a finite set of polynomials and produces as output a complete set S such that $I[S] = I[S']$, cf. [11]. Hence the ideal membership problem is decidable.

We can now restate the ideal membership problem (cf. Definition 1) as:

Definition 4 *The ideal membership problem is decidable by reduction with respect to S if and only if it can be formulated in the following way:*

$$f \in I[S] \quad \textit{if and only if} \quad f \rightarrow_S 0.$$

We note that if $f \rightarrow_S f'$ then the coset $\overline{f}_{I[S]}$ of f modulo $I[S]$ is an *invariant* with respect to the reduction relation $\rightarrow_S$, i.e

$$\overline{f}_{I[S]} = \overline{f'}_{I[S]}.$$

The following theorem states a necessary and sufficient condition for the solvability of the ideal membership problem.

Theorem 1 *The set S is a complete set for $I[S]$ if and only if the ideal membership problem is decidable by reduction with respect to S.*

Proof. Cf. [11].

4 Problem-Solving

We consider "problem-solving" in the sense used in artificial intelligence, as the discipline concerned with the development of general applicable intelligent strategies which implemented on computers can (help to) provide solutions by imitating human reasoning.

For defining what we mean by a problem we employ a similar to "state-operators" description, cf. [14]. Given a relational structure $\mathbf{A}$, we first define a *state space* having as elements *states*. A state is defined by a finite set of formulas in $\mathbf{A}$ (e.g., in the example at the end of this section a state is an enumeration of formulas $\bigwedge_{ij} \phi_{ij}$ where $\phi_{ij} = 0$ if and only if $cover(i,j) = true$); sets of formulas defining distinct states differ only in the values of the constants appearing in the formulas. A *state construction problem* expressed in $\mathbf{A}$ consists of a couple $StConP = (StSp, ConR)$ where $StSp$ represents the state space, and $ConR$ the *set of construction rules*. The states in $StSp$ and the construction rules in $ConR$ are characterized by formulas involving only constants, relations and functions from $\mathbf{A}$. Let $s, s' \in StSp$ and $cr \in ConR$. If rule cr applied to state s transforms it into state s' we say that s' is *reachable* from s. In order to establish an analogy with the reduction technique in symbolic computation described in Section 3, we will adopt the hypothesis that every rule is reversible in the sense that, if s' can be reached from s by cr then s can be reached from s' by applying the corresponding reversed rule r. We say that s' *reduces* to s via r, in symbols $s' \to_r s$, and call r a *reduction rule*. We will use $RedR$ to denote the set of reduction rules. A state s' can be reached from a state s if there exist a finite set of construction rules $\{cr_i\} \subseteq ConR$ which successively applied to s transforms it into s'. If this is the case, we will say that s' is in the *reachable space* of s, $s' \in ReSp[s]$.

In this context we will not be concerned with finding a solution (i.e a way to transform s in s' by specifying the construction rules $\{cr_i\}$ and the order in which these are applied, cf. [13]) but merely to state necessary and sufficient conditions for the existence of such a solution. Thus, we assume that a necessary and sufficient condition for state s' to be reached from state s is that s' can be reduced to s by successively applying reduction rules from $RedR$.

> "*Solving problems by heuristically guided, trial-and-error search in a space of possible solutions is a dominant theme in artificial-intelligence research.*" (cf. [14])

We propose a new approach to the subject by considering the study of necessary and sufficient conditions for a problem to be solvable, i.e:

> *Finding necessary and sufficient conditions for the solvability of the problems by heuristically guided, trial-and-error search for invariants corresponding to the reduction rules R.*

One could name such an attempt: *Invariant Driven Reasoning.*

Definition 1' *The state reachability problem consists of a triple* $StReP = (InSt, R, FinSt)$, *where* $InSt \subseteq StSp$ *is the set of initial states,* $FinSt \subseteq StSp$ *is the set of final states, and* $R \subseteq RedR$ *is the set of admissible reduction rules.*

(For simplicity reasons we will consider the case in which $InSt = \{is\}$ and $FinS = \{fs\}$ are singletons and $R = ConR \cup RedR$; which corresponds to considering the transitive symmetric closure of the reduction (or construction relation), cf. [12].)

The problem can now be stated as: Decide if the final state fs can be reduced to the initial state is by successively applying reduction rules from R, i.e $fs \in ReSp[is]$.

We will introduce now the basic concept for the proposed approach. A formula ϕ expressed in **A** which holds in state s' is said to be an *invariant formula* for the set of reduction rules R if for all $s \in StSp$ and $r \in R$ such that $s' \rightarrow_r s$ the formula ϕ holds also in the state s, i.e $\phi(s) = \phi(s')$, where $\phi(s)$ denotes the *value* of the invariant formula ϕ in state s.

Let $\Phi = \{\phi_i\}$ be a set of invariants for problem $StReP$.

Definition 2' *A set of invariants* Φ *is compatible with the problem* $StReP = (is, R, fs)$ *if the values of the invariants in the initial state are respectively equal to the values of the invariants in the final state*

$$\phi_i(is) = \phi_i(fs) \quad \textit{for all} \quad i.$$

Definition 3' *The set* Φ *is a complete set of invariants (or an invariant system) for the problem* $StReP$ *if every invariant formula of the problem* $StReP$ *can be expressed as a formula involving only invariants from* Φ.

Conceivably, the invariant system of a problem is not uniquely defined. Let Φ be an invariant system for the problem $StReP$.

Definition 4' *The state reachability problem* $StReP$ *is decidable by reduction with respect to* Φ *if and only if it can be formulated in the following way:*

> *The state reachability problem has a solution if and only if* Φ *is a compatible invariant system for the problem.*

The analog of Theorem 1 is the following statement.

Theorem 1' *The set* Φ *is a complete set of invariants for the problem* $StReP$ *if and only if the state reachability problem* $StReP$ *is decidable by reduction.*

Proof. Suppose Φ is complete with respect to R and compatible with respect to $StReP = (is, R, fc)$, and that $StReP$ has no solution. Let $s \in StSp$ be an arbitrary state and consider the formula $\phi(s)$: $\exists R'$ $(R' \subseteq R$ $finite; fc \rightarrow_{R'} s)$. Then, if $\phi(s)$ holds then $\phi(s')$ also holds for all states s' and rules r such that $s \rightarrow_r s'$, and if $\phi(s)$ does not hold then also $\phi(s')$ does not hold (recall that we have considered $\rightarrow_R$ as the symmetric transitive closure of the reduction relation).

This shows that ϕ is an invariant formula for R. But now $\phi(fs)$ holds (trivially) while $\phi(is)$ does not, since *StReP* has no solution. This is a contradiction with the fact that Φ is a complete, compatible invariant system.

Suppose now that Φ is complete and *StReP* is solvable. Then clearly Φ is compatible.

For the reverse implication of the theorem suppose *StReP* is decidable by reduction with respect to R and Φ is not complete. Then we can construct an invariant function ϕ_0 such that $\phi_0(fc) \neq \phi_0(ic)$ (because Φ is not complete) and this would contradict the assumption that *StReP* is solvable.

Example 2. Let $\mathbf{A} = \mathbb{N}$, the set of natural numbers. Consider a rectangular board with dimensions $a, b \in \mathbb{N}$, completely divided into squares of dimension 1. Suppose we have at our disposal at least $\lfloor a \cdot b/2 \rfloor$ domino tiles of dimensions 2×1. We can take away squares from the board without changing the position of the remaining squares obtaining in this way different configurations. The problem is:

> Given a certain configuration, decide whether it can be completely covered with non-overlapping domino tiles such that only the squares present in the configuration are covered.

We will first restate the problem as a state construction problem *StConP*. For this, let *StSp* be the state space corresponding to a given configuration, having as elements (states) different coverings of the given configuration (board) with domino tiles, and *ConR* the set of construction rules containing just one rule *cr*: *put a domino tile on two adjacent not covered squares in the configuration* (adjacent means here that they have a common edge).

The state construction problem *StConP* can now be formulated as:

> Given a board configuration and an initial state (assume, for simplicity, it is the one in which no tile has been placed, thus it actually coincides with the board configuration) and a final state (assume it is the one in which the given configuration has been completely covered with tiles) decide whether, using the construction rule *cr*, the initial state can be transformed into the final one, and, if so, show how can this be performed.

The reduction rule corresponding to *cr* is the rule *r*: *remove a tile from two adjacent squares in the configuration.*

The corresponding state reachability problem *StReP* can now be formulated as:

> Given a board configuration, an initial and a final state (as in *StConP*) decide whether the final state can be reduced to the initial one.

Consider a coordinate system $\mathbb{N} \times \mathbb{N}$ with the origin in the lower left corner of the board. Then, each square is represented by a pair of natural numbers $(x, y) \in \mathbb{N} \times \mathbb{N}$, $0 \leq x \leq a$, $0 \leq y \leq b$. For each state $s \in StSp$, i.e partial covering with tiles of the initial configuration, let us define the following predicates:

$cover(x, y) = true$ if and only if the square (x, y) is covered by a tile in state s; $cover$ characterizes the state;

$sing(x, y) = true$ if and only if the square (x, y) is in a *singular position* in state s, i.e all adjacent squares are covered with tiles;

$even(x, y) = true$ if and only if $x + y \equiv 0 \pmod 2$.

Let us denote by $N_{cover}, N_{sing}, N_{even}$ the number of positions (x, y) for which the corresponding predicates take the value $true$ (in state s). These all are functions defined on $StSp$ with values in $\mathbb{N}$.

For the initial state is, let us define the predicate:

$in - config(x, y) = true$ if and only if the square at position (x, y) is in the board configuration; thus $in - config$ characterizes the configuration of the board and, since we have considered by hypothesis that the board configuration coincides with the initial state, it also characterizes the initial state. Clearly, $N_{in-config}$ represents then the number of positions (squares) in the initial state (board configuration). The solvability of $StReP$ is equivalent to the conditions: $N_{cover}(is) = 0$ and $N_{cover}(fs) = N_{in-config}$.

Let us now put into evidence some of the invariants. It is easy to see that $N_d = N_{even} - N_{\neg even}$ is an invariant formula for the reduction rule r because a tile can be removed only from two adjacent squares. For the same reason N_{sing} is also an invariant. Clearly these two functions are independent, in the sense that they can take any value (in their codomains) independently of one another. It is also clear that N_0 in $N_{cov} \equiv N_0 \pmod 2$ is an invariant, but, since $N_{cover} = N_{even} + N_{\neg even} = 2N_{even} - N_d \equiv N_d \pmod 2$, it is not an independent (invariant) function. Now, since $N_{cover}(is) = 0 \equiv 0 \pmod 2$ it follows $N_0 = 0$, hence $N_{in-config} \equiv 0 \pmod 2$ is a necessary condition. Further, since $N_d(fs) = 0$ and $N_{sing}(fs) = 0$ it follows that, for the set of invariants $\{N_d, N_{sing}\}$ to be compatible, $N_d(is) = N_{sing}(is) = 0$. However the set $\{N_d, N_{sing}\}$ is not a complete invariant set. We were not able to find such a set for the problem.

We should also point out here that, in general we cannot reasonably hope for the existence of an algorithm which, performing on $StReP$s as input, will give an invariant system as output (even in the simple problem considered in the example, the relational structure we had to consider for expressing the invariants, contains as subsystem the number-theoretic formal system where incompleteness theorems hold, cf. [4]). This situation contrasts with the situation we had when we considered Buchberger's completion algorithm for polynomials. However, it is not difficult construct simple $StReP$s where a complete set of invariants can be exhibited.

5 Conclusions and Further Research

We have presented the basic idea of a new approach to problem solving from a certain unifying general perspective as invariant driven reasoning. While there is no way to systematically search for invariants, (i.e computer programs that "produce" such invariant formulas) we can envision programs that could test for invariant properties. In general, no completion algorithm exist, but since we know examples (in symbolic computation) were such algorithms do exist, it seems worthy to undertake such a research also in other particular fields.

An attempt for describing Robinson's resolution procedure as an instance (model) of the completion principle was made in [10]. It would be of interest to restate the resolution method using the notions introduced in this article.

Another research direction could be, given a relational structure **A** in which we express a problem $StReP$, to embed **A** into a richer algebraic structure where the invariants are solutions (fixed points) of certain equations (operators). Such a program was realized by Engeler [15], by constructing a combinatory model for programmable functions defined on a relational structure.

We are aware of the fact that our sketch for a new approach to problem-solving does not (yet) stand a comparison with the more developed techniques as resolution, rule-based strategies, etc., and (still) lacks convincing examples. However, viewed as a natural extrapolation of already proven successful methods in symbolic computation, there is a reasonable hope that analogies could perform well.

References

[1] Lang, S., (1984). *Algebra.* Addison-Wesley.

[2] Rota G.-C., Sturmfels B., (1988). Introduction to Invariant Theory in Superalgebras. *Invariant Theory and Tableaux*, ed., Stanton, D., Springer.

[3] Kalman, R., Arbib, M., Falb, P., (1965). *Topics in Mathematical System Theory*, McGraw-Hill.

[4] Kleene, S., (1967). *Introduction to Metamathematics.* North-Holland.

[5] Pfalzgraf, J., (1991). Logical Fiberings and Polycontextural Systems. In: *Fundamentals of Artificial Intelligence Research*, eds., Jorrand, Ph., Kelemen, J., LNCS **535**, Springer.

[6] Jacobson, N., (1980). *Basic Algebra II.* Freeman.

[7] Jouannaud, J-P., (1990). Syntactic Theories. In: *Mathematical Foundations of Computer Science*, ed., Rovan, B., LNCS **452**, Springer.

[8] Dershowitz, N., Jouannaud, J.-P. (1990). Rewrite Systems. In: *Handbook of Theoretical Computer Science. Vol B: Formal Methods and Semantics*, ed., van Leeuwen, J., North-Holland.

[9] Goguen, J., (1986). What Is Unification? In: *Resolution of Equations in Algebraic Structures. Vol. 1: Algebraic Techniques*, eds., Nivat, M., Ait-Kaci, H., Academic Press.

[10] Buchberger, B., (1987). History and Basic Features of the Critical-pair/completion Procedure. *J. Symbolic Computation*, **3**.

[11] Buchberger, B., (1976). A Theoretical Basis for the Reduction of Polynomials to Canonical Form. *ACM SIGSAM Bull.*, **10**.

[12] Winkler, F., (1984). The Church-Rosser Property in Computer Algebra and Special Theorem Proving. *PhD Thesis*, University. of Linz.

[13] Huet, G., (1987). Deduction and Computation. In: *Logic of Programming and Calculi of Discrete Design*, ed., Broy, M., Springer.

[14] Nilsson, N., (1971). *Problem Solving Methods in Artificial Intelligence.* McGraw-Hill.

[15] Engeler, E., (1990). Combinatory Differential Fields, *Theoretical Computer Science* **72**.