Logic Colloquium '73, Bristol, 1
July 1973, (eds H. E. Rose et al.,)
©North Holland, 1975, 231–251

On the Solvability of Algorithmic Problems

The classical construction problems of algebra and geometry have the following form: Given are a relational structure $\mathcal{A}$ and a problem predicate $\rho(x_1, \ldots, x_n; y_1, \ldots, y_m)$. Does there exist an algorithm which uses a limited supply of basic capabilities and which solves ρ, in the sense that it constructs, for any given elements $a_1, \ldots, a_n \in A$, a set of elements $b_1, \ldots, b_m \in A$ with the property that $\rho[a_1, \ldots, a_n; b_1, \ldots, b_m]$ holds in $\mathcal{A}$? The methods of classical Galois theory allow to discuss the solvability of the classical problems.

In this paper we develop the basic concepts of a generalized Galois theory in order to make the paradigm of Galois available for the discussion of the solvability of algorithmic problems which have the general form of the construction problems mentioned above. The first part of this paper presents this development as an abstract theory in the framework of the theory of models of a modest extension of first-order logic. In the second part we discuss some relations to computing. We conclude with some historical remarks.

1 The amalgamation and permutation properties

Let $\mathcal{A}$ be a relational structure. Let $L_{\omega_1\omega}(\mathcal{A})$ be the well-known infinitary language associated to $\mathcal{A}$ (strictly speaking: to the type of $\mathcal{A}$) and let L be any fragment of $L_{\omega_1\omega}(\mathcal{A})$ which has the following properties:

(i) L contains the atomic formulas and is closed under the finite Boolean connectives $\wedge, \neg$;

(ii) L has a complete (infinitary) proof system.

Lower case greek letters denote formulas; sets of formulas are denoted by capital greek letters. By the notation $\varphi(x, y, z), \Gamma(u, v), \Delta(x, Y)$ we indicate formulas, respectively sets of formulas, whose free variables are chosen from the sets $\{x, y, z\}, \{u, v\}, \{x\} \cup Y$, respectively.

Let L_0 be the set of *basic* (i.e., negated or unnegated atomic) formulas of $L_{\omega_1\omega}(\mathcal{A})$. (Thus $L_0 \subseteq L$.) Let X be a (countable) set of additional free variables. By $L(X)$ we understand the language obtained by admitting these new variables; $L_0(X)$ is the set of basic formulas of $L(X)$. By a *diagram* $\Delta(X)$ we understand any set of formulas of $L_0(X)$ such that for each $\alpha(x_1, \ldots, x_n)$ with $x_1, \ldots, x_n \in X$ exactly one of the formulas $\alpha(x_1, \ldots, x_n), \neg\alpha(x_1, \ldots, x_n)$ belongs to $\Delta(X)$.

If $\varphi(x,y) \in L$ and $a, b \in A$, we indicate by $\mathcal{A} \models \varphi[a,b]$ that $\varphi(x,y)$ holds in $\mathcal{A}$ for the assignment $\langle a,b\rangle$. If φ is closed, $\mathcal{A} \models \varphi$ indicates that $\mathcal{A}$ satisfies φ. The *provability* relation is indicated by $\vdash$: The expression $\Gamma \vdash \varphi$ states that φ is provable from Γ; by $\Gamma_1 \vdash \Gamma_2$ is expressed that $\Gamma_1 \vdash \gamma$ for at least one $\gamma \in \Gamma_2$. $\Gamma(x_1,\ldots) \vdash \varphi(x_1,\ldots)$ expresses that $\varphi(x_1,\ldots)$ is provable from $\Gamma(x_1,\ldots)$ with free variables $x_1, x_2, \ldots$ treated as parameters. Γ is *L-complete* if $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$ for all $\varphi \in L$. $\Gamma(x_1,\ldots)$ is L-complete if $\Gamma(x_1,\ldots) \vdash \varphi(x_1,\ldots)$ or $\Gamma(x_1,\ldots) \vdash \neg\varphi(x_1,\ldots)$ for all $\varphi \in L$ (with free variables among $x_1,\ldots$). Γ is *consistent* if it has a model. By a problem predicate, or simply a *problem*, we understand a formula

$$\rho(x_1,\ldots,x_n,y_1,\ldots,y_m)$$

of L. For notational convenience only we restrict ourselves to the case $m = n = 1$.

Let $Y = \{y_1, y_2, \ldots\}$ be a countable (finite or denumerably infinite) set of free variables. Let $S(Y)$ be the group of all permutations of Y. For any $t \in S(Y)$ and $\varphi(x, y_1, \ldots) \in L$ define

$$\varphi^t(x, y_{i_1}, y_{i_2}, \ldots) \equiv_{def} \varphi(x, t(y_{i_1}), t(y_{i_2}), \ldots)$$

By extension, if $\Phi(x,Y) \subseteq L$ and $t \in S(Y)$, we let

$$\Phi^t(x,Y) = \{\varphi^t(x, y_{i_1}, \ldots) : \quad \varphi(x, y_{i_1}, \ldots) \in \Phi(x,Y)\}$$

Let $\Delta(x)$ be a diagram.

Definition 1 *A problem $\rho(x,y)$ is* well posed *with respect to Γ and $\Delta(x)$ if there exists Y and a diagram $\Delta(x,Y)$ satisfying the following conditions:*

(a) $\Gamma \cup \Delta(x,Y)$ *is consistent;* $\Gamma \cup \triangle(x,Y) \vdash \rho(x,y_i)$ *for all* $y_i \in Y$,

(b) $(y_i \neq y_j) \in \Delta(x,Y)$ *for all* $y_i, y_j \in Y, i \neq j$,

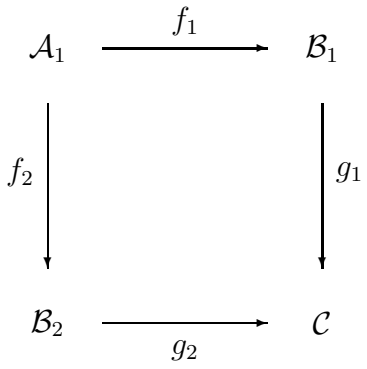
(c) $\Gamma \cup \Delta(x,Y) \cup \{\rho(x,y)\} \cup \{\rho(x,y_i) : y_i \in Y\} \vdash \{y = y_i : y_i \in Y\}$.

Definition 2 Γ *has the* permutation property *with respect to ρ if for any $\Delta(x,Y), \Delta'(x,Y)$ which satisfy (a), (b) and (c) above and for which $\Delta(x,Y) \cap L_0(x) = \Delta'(x,Y) \cap L_0(x)$, there exists $s \in S(Y)$ such that*

$$\Delta^s(x,Y) = \Delta'(x,Y).$$

The permutation property will play an important role in the sequel. We therefore formulate a more familiar concept from universal algebra which implies it.

Definition 3 Γ *has the* amalgamation property *if for any models* $\mathcal{A}, \mathcal{B}_1$ *and* $\mathcal{B}_2$ *of* Γ *and any injections* $f_1 : \mathcal{A} \to \mathcal{B}_1$ *and* $f_2 : \mathcal{A} \to \mathcal{B}_2$*, there exists a model* $\mathcal{C}$ *of* Γ *and injections* $g_1 : \mathcal{B}_1 \to \mathcal{C}$ *and* $g_2 : \mathcal{B}_2 \to \mathcal{C}$ *such that the following diagram commutes:*



Theorem 4 *If* Γ *is universal and has the amalgamation property, then it has the permutation property with respect to all well-posed problems.*

Proof. Let $\mathcal{B}_1$ be a model of $\Gamma \cup \Delta(x, Y)$ and $\mathcal{B}_2$ a model of $\Gamma \cup \Delta'(x, Y)$. Let T_x be the set of terms of $L(x)$ which have no other free variables except x. Let $\mathcal{A}_1, \mathcal{A}_2$ be the restrictions of $\mathcal{B}_1, \mathcal{B}_2$ to the elements denoted by terms in T_x. The structures $\mathcal{A}_1, \mathcal{A}_2$ are models of Γ since Γ is universal. Furthermore $\mathcal{A}_1 \cong \mathcal{A}_2$ since

$$\Delta(x, Y) \cap L_0(x) = \Delta(x) = \Delta'(x, Y) \cap L_0(x)$$

Let therefore $\mathcal{A} = \mathcal{A}_1 = \mathcal{A}_2$, and denote by f_i the canonical injections $f_i : \mathcal{A} \to \mathcal{B}_i, i = 1, 2$. By the amalgamation property there exists a model $\mathcal{C}$ of $\Gamma \cup \Delta(x)$ and injections $g_i : \mathcal{B}_1 \to \mathcal{C}$ such that the above diagram commutes. Let $a', b'_1, b'_2, \ldots$ be the denotations of $x, y_1, y_2, \ldots$ in $\mathcal{B}_1$, and a'', b''_1, b''_2, the denotations of these variables in $\mathcal{B}_2$; let a be the denotation of x in $\mathcal{C}$ and let $c_1, c_2, \ldots$ be an enumeration of all solutions $\rho[a, c]$ in $\mathcal{C}$.

By the canonical injections: $g_1(a') = g_2(a'')$ and, $\rho(x, y)$ being well posed,

$$\{c_1, c_2, \ldots\} = \{g_1(b'_1), g_1(b'_2), \ldots\} = \{g_2(b''_1), g_2(b''_2), \ldots\}$$

Let t be such that for all j

$$g_1(b'_{t(j)}) = g_2(b''_j)$$

and define $s \in S(Y)$ by

$$s(y_j) = y_{t(j)}, \quad j = 1, 2, \ldots$$

Then

$$\mathcal{B}_1 \models \alpha^s[a', b'_{i_1}, \ldots b'_{i_m}] \quad \text{iff} \quad \mathcal{B}_2 \models \alpha[a'', b''_{i_1}, \ldots, b''_{i_m}]$$

for all atomic α, as is easy to verify. Hence (by isomorphisms) for all terms τ'_i over a', b'_i and corresponding terms τ''_i over a'', b''_i :

$$\mathcal{B}_1 \models \alpha^s[\tau'_1, \ldots, \tau'_k] \quad \text{iff} \quad \mathcal{B}_2 \models \alpha[\tau''_1, \ldots, \tau''_k]$$

It follows that $\Delta'(x, Y) = \Delta^s(x, Y)$.

Definition 5 *A problem $\rho(x, y)$ is of* degree n *in Γ if*

$$\Gamma \cup \{\rho(x, y_i)\}_{i=0}^n \vdash \{y_i = y_j\}_{0 \leq i < j \leq n}.$$

Since problems of finite degree are a fortiori well posed, we have:

Corollary 6 *If Γ is universal, has the amalgamation property and $\rho(x, y)$ is of finite degree, then Γ has the permutation property with respect to ρ.*

2 Galois theory

Let $L \subseteq L_{\omega_1\omega}(\mathcal{A})$ and $L' \subseteq L$ satisfy the conditions of Section 1. Let $\Gamma \subseteq L$ be consistent, $\rho(x, y) \in L$ a well-posed problem with respect to Γ and let $\Delta(x, Y) \subseteq L_0$ be a diagram satisfying condition (a), (b) and (c) of Definition 1.

Our first task is to define the group of ρ. The idea is to take all permutations of solutions of $\rho(x, y)$ which leave the truth-values of formulas of L' unchanged. More exactly:

Definition 7

$$G' = \{t \in S(Y) : \Gamma \cup \Delta(x, Y) \vdash \varphi \equiv \varphi^t \textit{ for all } \varphi \in L'(x, Y)\}$$

$$G = \{t \in S(Y) : \Gamma \cup \Delta(x, Y) \vdash \varphi \equiv \varphi^t \textit{ for all } \varphi \in L_0(x, Y)\}$$

The sets G, G' turn out to be groups and depend only on $\Delta(x) = \Delta(x, Y) \cap L_0(x)$, as is shown by the following two lemmas.

Lemma 8 *G, G' are groups.*

Proof. We show closure under composition and inverses. Suppose that

$$\begin{array}{l} \Gamma \cup \Delta(x, Y) \vdash \varphi_1 \equiv \varphi_1^{t_1} \\ \Gamma \cup \Delta(x, Y) \vdash \varphi_2 \equiv \varphi_2^{t_2} \quad \text{for all } \varphi_1, \varphi_2 \in L' \text{ (resp. } L_0) \end{array}$$

From the second line, taking $\varphi_2 = \varphi_1^{t_1}$, we get

$$\Gamma \cup \Delta(x, Y) \vdash \varphi_1^{t_1} \equiv \varphi_1^{t_1 t_2}$$

and, combining with the first line,

$$\Gamma \cup \Delta(x, Y) \vdash \varphi_1 \equiv \varphi_1^{t_1 t_2}$$

For inverses, if we assume

$$\Gamma \cup \Delta(x, Y) \vdash \varphi_1 \equiv \varphi_1^t \text{ for all } \varphi_1, \in L \text{ (resp. } L_0)$$

we obtain, taking $\varphi_1 = \varphi^{t^{-1}}$

$$\Gamma \cup \Delta(x, Y) \vdash \varphi^{t^{-1}} \equiv \varphi^{t^{-1}t}$$

hence $\Gamma \cup \Delta(x, Y) \vdash \varphi \equiv \varphi^{t^{-1}}$.

Let $\Delta_1 = \Delta(x, Y), \Delta_2 = \Delta'(x, Y)$ satisfy conditions (a), (b) and (c) of Definition 1. Let $\Delta(x) = \Delta_1 \cap L_0(x) = \Delta_2 \cap L_0(x)$, and let G_1, G_1', G_2, G_2' be the groups obtained in Definition 5 using Δ_1, respectively Δ_2. *Assume from now on that Γ has the permutation property with respect to ρ.*

Lemma 9 *$G_1 \cong G_2$ and $G_1' \cong G_2'$.*

Proof. Suppose $t \in G_1'$, i.e. $\Gamma \cup \Delta_1 \vdash \varphi \equiv \varphi^t$ for all $\varphi \in L'$. Let $s \in S(Y)$. Then $\Gamma \cup \Delta_1^s \vdash \varphi^s \equiv \varphi^{ts}$ for all $\varphi \in L'$ and all $t \in G_1'$, in particular for $\varphi' = \varphi^{s^{-1}}$. Thus $\Gamma \cup \Delta_1^s \vdash \varphi^{s^{-1}s} \equiv \varphi^{s^{-1}ts}$ for all $\varphi \in L'$ and $t \in G_1'$. Hence $\Gamma \cup \Delta_1^s \vdash \varphi \equiv \varphi^{t'}$ for all $\varphi \in L'$ and $t' \in s^{-1}G_1's$. Since Γ has the permutation property there is $s \in S(Y)$ such that $\Delta_2 = \Delta_1^s$. Hence $\Gamma \cup \Delta_2 \vdash \varphi \equiv \varphi^{t'}$ for all $t' \in s^{-1}G_1's$. Thus $G_1' \subseteq G_2'$ by an inner automorphism of $S(Y)$; symmetrically, $G_2' \subseteq G_1'$ by its inverse; hence $G_1' \cong G_2'$. The proof of $G_1 \cong G_2$ is the same.

Lemmas 1 and 2 justify talking of G, G' as groups of the problem ρ. To indicate their dependence on $\Delta(x)$ only, we introduce the notation

$$G_{\Delta(x)}(\rho), \quad G'_{\Delta(x)}(\rho)$$

for these groups. - The remainder of this section is devoted to the task of justifying our claim that $G_{\Delta(x)}(\rho)$ are reasonable generalizations of the concept of a Galois group.

Let $\Gamma \subseteq L, \Delta(x, Y)$ be given, and let $\rho(x, y)$ be well posed with respect to Γ, i.e., assume conditions (a), (b) and (c) of Definition 1 for these $\Gamma, \Delta(x, Y)$.

Theorem 10

(1) $G_{\Delta(x)}(\rho) = \{s \in S(Y) : \Delta(x, Y) = \Delta^s(x, Y)\}$.

(2) If $\Gamma \cup \Delta(x, Y)$ is L'-complete, then $G'_{\Delta(x)}(\rho) = G_{\Delta(x)}(\rho)$.

Proof. It suffices to show that for L'-complete $\Gamma \cup \Delta(x, Y)$

$$G'_{\Delta(x)}(\rho) = \{s \in S(Y) : \Delta(x, Y) = \Delta^s(x, Y)\}$$

because $\Gamma \cup \Delta(x, Y)$ is clearly L_0-complete, $\Delta(x, Y)$ being a diagram.

Suppose $s \in S(Y)$ is such that $\Delta(x, Y) = \Delta^s(x, Y)$, and assume that for some $\varphi \in L'$ we have

$$\Gamma \cup \Delta(x, Y) \nvdash \varphi \equiv \varphi^s$$

Then, by completeness, either

$$\Gamma \cup \Delta(x, Y) \vdash \varphi \text{ and } \Gamma \cup \Delta(x, Y) \vdash \neg\varphi^s$$

or

$$\Gamma \cup \Delta(x, Y) \vdash \varphi^s \text{ and } \Gamma \cup \Delta(x, Y) \vdash \neg\varphi$$

In the first case, we would have $\Gamma \cup \Delta^s(x, Y) \vdash \varphi^s$, hence $\Gamma \cup \Delta(x, Y) \vdash \varphi^s$, a contradiction; similarly in the second case.

Conversely, suppose $s \in G'_{\Delta(x)}(\rho)$. Since

$$\Gamma \cup \Delta(x, Y) \vdash \varphi \text{ for all } \varphi \in \Delta(x, Y)$$

and

$$\Gamma \cup \Delta(x, Y) \vdash \varphi \supset \varphi^s \text{ for all } \varphi \in \Delta(x, Y)$$

we have

$$\Gamma \cup \Delta(x, Y) \vdash \varphi^s \text{ for all } \varphi \in \Delta(x, Y)$$

Since $\Delta(x, Y)$ is a diagram, we conclude $\Delta(x, Y) = \Delta^s(x, Y)$.

Part (1) of Theorem 2 allows us to connect our definition of the group of a problem with a more familiar one. Namely, assume that Γ is a universal theory and let

$$\mathcal{A}(x), \mathcal{A}(x, Y)$$

be the minimal models of $\Gamma \cup \Delta(x), \Gamma \cup \Delta(x, Y)$, respectively. $\mathcal{A}(x)$ is a substructure of $\mathcal{A}(x, Y)$ and any automorphism of $\mathcal{A}(x, Y)$ leaving $\mathcal{A}(x)$ pointwise fixed induces a permutation $s \in S(Y)$ such that $\Delta^s(x, Y) = \Delta(x, Y)$. Conversely, every such permutation induces an automorphism of $\mathcal{A}(x, Y)$ over $\mathcal{A}(x)$. Hence:

Corollary 11 *If Γ is universal, i.e., consists only of universal sentences, then $G_{\Delta(x)}(\rho)$ is the group of automorphisms of $\mathcal{A}(x, Y)$ over $\mathcal{A}(x)$.*

For later applications it is important to know for which $\varphi \in L_{\omega_1\omega}(\mathcal{A})$ the basic property of $G_{\Delta(x)}(\rho)$ holds, i.e., for which φ we have

$$\Gamma \cup \Delta(x, Y) \vdash \varphi \equiv \varphi^s \text{ for all } s \in G_{\Delta(x)}(\rho)$$

By Theorem 2, part (2) we know that this is true for all $\varphi \in L'$ whenever $\Gamma \cup \Delta(x, Y)$ is L'-complete. The following theorem shows that for some, still rather expressive L' this condition is not needed. These L' include in particular the language that we shall need for the expression of algorithmic properties in Section 3.

Let L_1 be the closure of L_0 under finite Boolean operations and denumerable disjunctions, i.e., let L_1 be the smallest subset of $L_{\omega_1\omega}(\mathcal{A})$ such that

$$\begin{array}{lll} L_0 \subseteq L_1; & & \\ \text{if } \varphi, \psi \in L_1, & \text{then } \varphi \wedge \psi \in L_1 & \text{and } \neg\varphi \in L_1; \\ \text{if } \varphi_i \in L_1, i = 1, 2, \ldots, & \text{then } \bigvee_{i=1}^{\infty} \varphi_i \in L_1. & \end{array}$$

Theorem 12

$$G^{L_1}_{\Delta(x)}(\rho) = \{s \in S(Y) : \Gamma \cup \Delta(x, Y) \vdash \varphi \equiv \varphi^s, \ \textit{all } \varphi \in L_1\} = G_{\Delta(x)}(\rho)$$

Proof. We show $\Gamma \cup \Delta(x, Y) \vdash \varphi \equiv \varphi^s$ for all $\varphi \in L_1$ by induction on the structure of φ. By symmetry, it is sufficient to show

$$\Gamma \cup \Delta(x, Y) \vdash \varphi \supset \varphi^s \text{ for all } \varphi \in L_1$$

For $\varphi = \varphi_1 \wedge \varphi_2$ we assume $\Gamma \cup \Delta(x, Y) \vdash \varphi_1 \supset \varphi_1^s$ and $\Gamma \cup \Delta(x, Y) \vdash \varphi_2 \supset \varphi_2^s$. Hence $\Gamma \cup \Delta(x, Y) \vdash (\varphi_1 \wedge \varphi_2) \supset \varphi_1^s$ and $\Gamma \cup \Delta(x, Y) \vdash (\varphi_1 \wedge \varphi_2) \supset \varphi_2^s$; thus

$$\Gamma \cup \Delta(x, Y) \vdash (\varphi_1 \wedge \varphi_2) \supset \varphi_1^s \wedge \varphi_2^s$$
$$\Gamma \cup \Delta(x, Y) \vdash (\varphi_1 \wedge \varphi_2) \supset (\varphi_1 \wedge \varphi_2)^s$$

For $\varphi = \neg\varphi_1$ we assume $\Gamma \cup \Delta(x, Y) \vdash \varphi_1 \supset \varphi_1^s$. By contraposition, $\Gamma \cup \Delta(x, Y) \vdash \neg\varphi_1^{s^{-1}} \supset \neg\varphi_1$, hence $\Gamma \cup \Delta^s(x, Y) \vdash \neg\varphi_1 \supset \neg\varphi_1^s$, and because

$$\Delta(x, Y) = \Delta^s(x, Y), \quad \Gamma \cup \Delta(x, Y) \vdash \neg\varphi_1 \supset (\neg\varphi_1)^s$$

For $\varphi = \bigvee_i \varphi_i$ we assume $\Gamma \cup \Delta(x, Y) \vdash \varphi_i \supset \varphi_i^s$ for all i. It follows that $\Gamma \cup \Delta(x, Y) \vdash \varphi_i \supset \bigvee_i \varphi_i^s$ for all i, and therefore $\Gamma \cup \Delta(x, Y) \vdash \bigvee_i \varphi_i \supset \bigvee_i \varphi_i^s$, which is the same as $\Gamma \cup \Delta(x, Y) \vdash \bigvee_i \varphi_i \supset (\bigvee_i \varphi_i)^s$.

How should we go about actually determining the group $G_{\Delta(x)}(\rho)$? For problems $\rho(x, y)$ of finite degree classical Galois theory suggests that one can find a single equation $\Theta(x, y_1, \ldots, y_n)$ the so-called resolvent, and determine the set of permutations of $y_1, \ldots, y_n$ which leave the value of $\Theta(x, y_1, \ldots, y_n)$ unchanged, with the change that instead of being a single equation, $\Theta(x, y_1, \ldots, y_n)$ will in general be a finite conjunction of basic formulas.

Theorem 13 *Let Γ be a set of sentences, and let $\rho(x, y)$ be a problem of degree n in Γ. Then there exists a finite conjunction $\Theta(x, y_1, \ldots, y_n)$ of basic formulas in L_0 such that*

$$\begin{aligned} G_{\Delta(x)}(\rho) \;=\; & \{s \in S(y_1,\ldots,y_n) : \\ & \Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \Theta(x,y_1,\ldots,y_n) \equiv \Theta^s(x,y_1,\ldots,y_n)\} \end{aligned}$$

Proof. Let $S(y_1,\ldots,y_n) - G_{\Delta(x)}(\rho) = \{t_1,t_2,\ldots,t_m\}$ and assume that a conjunction $\Theta(x,y_1,\ldots,y_n)$ of basic formulas has been determined such that

$$\begin{aligned} &\Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \Theta_i \\ &\Gamma \cup \Delta(x,y_1,\ldots,y_n) \nvdash \Theta_i \equiv \Theta_i^{t_j} \quad \text{for all } j \leq i \end{aligned}$$

Consider t_{i+1}. If $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \nvdash \Theta_i \equiv \Theta_i^{t_j}$, we let $\Theta_{i+1} = \Theta_i$. Otherwise let $\varphi_{i+1}(x,y_1,\ldots,y_n) \in L_0$ be chosen such that $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \nvdash \varphi_{i+1} \equiv \varphi_{i+1}^{t_{i+1}}$, say $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \varphi_{i+1}$ and $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \neg\varphi_{i+1}$. Then $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \Theta_i \wedge \varphi_{i+1}$ but not $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \Theta_i^{t_{i+1}}$, that is, $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \nvdash (\Theta_i \wedge \varphi_{i+1})^{t_{i+1}}$. Hence, $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \nvdash (\Theta_i \wedge \varphi_{i+1}) \equiv (\Theta_i \wedge \varphi_{i+1})^{t_{i+1}}$. We may therefore take $\Theta_{i+1} = \Theta_i \wedge \varphi_{i+1}$ in this case. Finally, we let $\Theta = \Theta_m$ and observe by construction $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \nvdash \Theta \equiv \Theta^t$ for all $t \notin G_{\Delta(x)}(\rho)$, but $\Gamma \cup \Delta(x,y_1,\ldots,y_n) \vdash \Theta \equiv \Theta^t$ for all $t \in G_{\Delta(x)}(\rho)$.

3 Algorithmic languages[1]

In the present section we sketch a logical framework in which those problems which we shall call algorithmic can conveniently be treated in a formal fashion. We start by enumerating a number of problem types which belong to this area and have received a good deal of attention from computer scientists.

By a *data structure* we simply understand a relational structure $\mathcal{A}$ of finite type, consisting of a non-empty underlying set A, some finitary relation R_i and operations f_j.

$$\begin{aligned} \mathcal{A} &= \langle A; \ldots, R_i, \ldots; \ldots, f_j, \ldots \rangle \\ R_i &\subseteq A^{n_i}, \quad f_j : A^{m_j} \to A \end{aligned}$$

With respect to such a data structure we imagine a given set of *elementary capabilities* which consist in admitting the executability of some atomic acts of decision and operations. Typically, a list of capabilities is given by enumerating types of instructions:

[1]The material of Sections 3 and 4 was presented at the 1. Fachtagung der Ges. f. Informatik, Bonn, July 1973 and appears in their proceedings LN Computer Science, Springer, 1973, 2-15. It is reprinted here with some slight changes with the permission of the publishers.

$$\mathcal{B} = \{\ldots, x_i := f_j(x_{k_1}, \ldots, x_{k_{n_j}}), \ldots, R_j(x_{k_1}, \ldots, x_{k_{m_j}}), \ldots\}$$

where f_j, R_j are the operations and relations of $\mathcal{A}$, $x_1, x_2, \ldots$ are variables ranging over A.

With the aid of these capabilities we construct programs in the manner of ALGOL (*flowchart programs* use only *go to* and *if then else; recursive programs* use procedure calls).

Let π be a program in the variables $x_1, \ldots, x_n$ with respect to some elementary capabilities $\mathcal{B}$ for some relational structure $\mathcal{A}$. The *termination problem* for π is: 'does $\pi(x_1, \ldots, x_n)$ halt for all assignments of initial values $a_1, \ldots, a_n \in A$ to $x_1, \ldots, x_n$?' We formulate this problem as

$$\mathcal{A} \models \forall x \ \mathrm{Term}_\pi(x) \tag{1}$$

where $\mathrm{Term}_\pi(x)$ is short for $\mathrm{Term}_\pi(x_1, \ldots, x_n)$ and denotes the termination predicate (n-ary) which is true for $\langle a_1, \ldots, a_n\rangle$ iff π halts on input $a_1, \ldots, a_n$. (So far we have not specified a language in which Term_π is a formula.)

The problem of *partial correctness* of a program $\pi(x_1, \ldots, x_n)$ is: "given that the values assigned to the variables $x_1, \ldots, x_n$ at input time satisfy the predicate $\varphi(x_1, \ldots, x_n)$ and assuming that π terminates on that input, do the values assigned to the variables $x_1, \ldots, x_n$ at output time satisfy the predicate $\psi(x_1, \ldots, x_n)$?" Let $\mathrm{Trans}_\pi(x, y)$ be short for $\mathrm{Trans}_\pi(x_1, \ldots, x_n; y_1, \ldots y_n)$ and denote the transduction predicate of π which is true for $\langle a_1, \ldots, a_n; b_1, \ldots, b_n\rangle$ exactly when π, with input $a_1, \ldots, a_n$ terminates with output $b_1, \ldots, b_n$. The problem of partial correctness is formulated in these terms by

$$\mathcal{A} \models \forall x, y(\varphi(x) \wedge \ \mathrm{Trans}_\pi(x, y). \supset \psi(y)) \tag{2}$$

The problem of *equivalence* of two programs $\pi_1(x_1, \ldots, x_n)$ and $\pi_2(x_1, \ldots, x_n)$ is formulated as

$$\mathcal{A} \models \forall x, y(\mathrm{Trans}_{\pi_1}(x, y) \equiv \ \mathrm{Trans}_{\pi_2}(x, y)) \tag{3}$$

A further important algorithmic problem is the problem of *algorithmic solvability*: "Given a problem predicate $\rho(x_1, \ldots, x_n; y_1, \ldots y_n)$ does there exist a program $\pi(x_1, \ldots, x_n)$ which solves ρ in the sense that whenever π halts on input $a_1, \ldots, a_n$ with output $b_1, \ldots, b_n$ then $\langle a_1, \ldots, a_n, b_1, \ldots, b_n\rangle$ satisfies ρ?" The classical construction problems

of algebra and geometry clearly have this form. We formulate the problem of algorithmic solvability by:

$$\exists \pi ? \mathcal{A} \models \forall x, y(\mathrm{Trans}_\pi(x, y) \supset \rho(x, y)) \tag{4}$$

Closely related to the problem of algorithmic solvability is that of *conditioning*: 'Given a problem predicate ρ and a proposed algorithmic solution π, find a condition φ on the input which guarantees that π is a solution for inputs satisfying φ!' We may formulate this by

$$\exists \varphi ? \mathcal{A} \models \forall x, y(\varphi(x) \wedge (\mathrm{Trans}_\pi(x, y). \supset \rho(x, y)) \tag{5}$$

With this example we close for now the enumeration of problems that we would call algorithmic. It is apparent that the list contains patterns for problems that justify considerable mathematical activity. For us, this activity takes on the form of establishing, and working within, an appropriate formalized framework; a logical framework as it were. What are the primary requirements for such a frame? Clearly, it must be able to talk about properties of structure $\mathcal{A}$, at least to the extent of formulating the predicates $\varphi(x), \psi(y)$ in partial correctness (2), problem predicates $\rho(x, y)$ in (4) and (5), and it must hold out fair hope of finding the condition predicate $\varphi(x)$ in the conditioning problem (5). On the other hand, it must be able to connect these predicates up with the assertions concerning the meaning of a program π; in particular, it should be able to express the predicates Term_π and Trans_π. In addition to these semantical requirements on the frame there is the goal of adding a theoretically manageable, perhaps even a computer-implementable, deductive system to it.

REMARK 1. For practical purposes it may be advantageous to concentrate on one type of algorithmic problem and create a formal system for that alone. This has been done by Hoare [9] for the problem of partial correctness. No need arose to write Term_π explicitly in the system, and it was judged (following the lead of Floyd [6]) that for properties of the structure $\mathcal{A}$ it was convenient to choose the language of first-order predicate logic. Hoare's 'logic' has been used for various tasks, in particular as a basis for automatic program verification and program generation (Luckham and Manna), and for the axiomatic description of a programming language (Hoare and Wirth [10]). (A noteworthy switch in emphasis is present in the last paper: Instead of attempting a logically complete system for a fixed semantic, the system of rules and axioms is deliberately left rudimentary in order to accommodate varying implementations.)

REMARK 2. Actually, the formal languages in which programs are formulated are themselves good enough formalisms for which it is easy to formulate proof procedures. Such a system (working with program statements exclusively) has been proposed by Engeler [5]. Some questions of correctness can easily be formulated as the termination

property. Indeed, for programs in numerical analysis this approach has considerable intuitive appeal, as suggested in Engeler [4].

REMARK 3. A third, very promising approach that goes all the way from a theoretical logical basis (Scott [22]) to implementation (Milner [14]) uses the concepts of λ-calculus.

The task of choosing an appropriate language in which to formulate algorithmic problems is simplified by the observation that $\mathrm{Trans}_\pi(x, y)$ can easily be expressed as $\mathrm{Term}_{\pi'}(x, y)$ for a new program π' (which arises from π by first storing away the values for y, then executing π and finally checking the actual outcome against the stored y values). A second observation is of empirical nature: We have not encountered 'honest' algorithmic problems in which the formulas φ, ψ, ρ in (1)-(5) were not finite Boolean combinations of formulas of the form Term_π. These REMARKs have led to the development of our theory which is based on a language whose expressive power encompasses not much more than (universally quantified) finite Boolean combinations of termination formulas. This attitude, first formulated in Engeler [3], has been at the basis of the work of a group of Polish logicians under the direction of Professor Rasiowa who modified and developed the present author's approach (Salwicki [20, 21], Mirkowska [15], Kreczmar [13], Grabowski [7]).

Let $\mathcal{B} = \{\ldots, x_i := f^r(x_j, x_k), \ldots, R_s(x_i, x_j), \ldots\}$ be an enumeration of the elementary capabilities with which we are concerned. To $\mathcal{B}$ we associate an alphabet

$$alph(\mathcal{B}) = \{\ldots, {}_i f^r_{jk}, \ldots, {}_s R^+_{jk}, {}_s R^-_{ij}, \ldots\}$$

Words over this alphabet are used to represent possible traces through a given program over π; if an operation $x_i := f^r(x_j, x_k)$ is encountered along a path through π this is indicated by appending the symbol ${}_i f^r_{jk}$ to the trace; if a decision box $R_s(x_i, x_j)$ is left through the positive branch, we indicate this by appending ${}_s R^+_{ij}$, etc. In this purely syntactical fashion we associate to a program π a set of words over $alph(\mathcal{B})$, which is called the language of π, in short,

$$lang(\pi) = \{w \in alph(\mathcal{B})^* : w \in \text{ paths through } \pi\}$$

It is well known that $lang(\pi)$ is a regular set if π is a flowchart program; if π is a recursive program, then $lang(\pi)$ is a context-free language. Accordingly, $lang(\pi)$ can be denoted by a regular expression $\sigma(\pi)$ in the first case, by a Gruska [8] context-free expression $\sigma(\pi)$ in the second case; σ is called signature of π

$$lang(\pi) =| \sigma(\pi) |$$

Let π be any program, $\sigma(\pi)$ a signature of π and $w \in |\ \sigma(\pi)\ |$; assume that the variables occurring in π are among $x_1, \ldots, x_n$. We first construct a quantifier-free first-order formula

$$\varphi_w(x_1, \ldots, x_n)$$

which is true in $\mathcal{A}$ for an input $a_1, \ldots, a_n$ exactly if π would follow the path w through π on that input. Letting λ denote the empty word and $\cdot$ the concatenation of words we define recursively

$$\begin{aligned}
\varphi_\lambda(x_1, \ldots, x_n) &:= (x_1 = x_1) \\
\varphi_i f^r_{jk} \cdot w(x_1, \ldots, x_n) &:= \mathrm{Sub}^{x_i}_{f^r(x_j, x_k)}(\varphi_w(x_1, \ldots, x_n)) \\
\varphi_s R^+_{ij} \cdot w(x_1, \ldots, x_n) &:= R_s(x_i, x_j) \wedge \varphi_w(x_1, \ldots, x_n) \\
\varphi_s R^-_{ij} \cdot w(x_1, \ldots, x_n) &:= \neg R_s(x_i, x_j) \wedge \varphi_w(x_1, \ldots, x_n)
\end{aligned}$$

With this abbreviation we now have a formula for $\mathrm{Term}_\pi(x_1, \ldots, x_n)$ in the language $L_{\omega_1\omega}$, namely

$$\mathrm{Term}_\pi(x_1, \ldots, x_n) := \bigvee_{w \in |\sigma(\pi)|} \varphi_w(x_1, \ldots, x_n)$$

Clearly, $\mathcal{A} \models \mathrm{Term}_\pi[a_1, \ldots, a_n]$ in the sense of the semantics of $L_{\omega_1\omega}$ iff $\pi(x_1, \ldots, x_n)$ halts in $\mathcal{A}$ on input $a_1, \ldots, a_n$.

In view of our earlier discussion we therefore define our *algorithmic* language (in distinction to the given *programming* language) as:

$$\begin{aligned}
algL(\mathcal{B}) \quad = \quad & \{\text{set of all finite Boolean combinations of formulas} \\
& \mathrm{Term}_\pi(x), \pi \text{ a flowchart program over } \mathcal{B}\}.
\end{aligned}$$

Let us now consider a given relational structure $\mathcal{A}$ and ask what are the properties of $\mathcal{A}$ relevant to various algorithmic problems. Looking back on their formulation, we observe that all we need to know about $\mathcal{A}$ is its *algorithmic theory*:

$$algT(\mathcal{A}; \mathcal{B}) = \{\varphi \in algL(\mathcal{B}) : \mathcal{A} \models \forall x \varphi(x)\}$$

Since $L_{\omega_1\omega}$ has a complete (infinitary) proof system, we have one for $algL(\mathcal{B})$. It is therefore possible to ask for the axiomatizability of $algT(\mathcal{A}; \mathcal{B})$ for various $\mathcal{A}$ and $\mathcal{B}$

which are of interest. For we can then ask $algT(\mathcal{A};\mathcal{B}) \vdash \psi$ instead of $\mathcal{A} \models \psi$. (Of course, if in the definition of $algL(\mathcal{B})$ we had chosen recursive programs instead of flowcharts, we would *in general* get a different theory $algT(\mathcal{A};\mathcal{B})$; for the example below, however, the choice is immaterial.)

For any set Γ of algorithmic formulas, let $\Gamma^{\vdash}$ denote its deductive closure under the rules of proof of $L_{\omega_1\omega}$ in $algL(\mathcal{B})$, i.e.,

$$\Gamma^{\vdash} = \{\psi \in algL(\mathcal{B}) : \forall x\psi \text{ can be proven from the set of universally quantified formulas of } \Gamma\}$$

To *axiomatize* $algL(\mathcal{A};\mathcal{B})$, therefore, means finding a recursive set Γ such that $\Gamma^{\vdash} = algT(\mathcal{A};\mathcal{B})$. We now give a few examples.

4 The classical constructibility theories

We present, by way of illustration, some examples of axiomatizations of $algT(\mathcal{A};\mathcal{B})$ for structures $\mathcal{A}$ and elementary capabilities $\mathcal{B}$ familiar from classical constructibility theories in mathematics. The purpose is to give a precise formulation to these theories, which were largely unformalized and to a degree imprecisely conceived (by modern standards). Earlier formulations, e.g. Tarski [24], Suppes-Moler [23] differ from ours by not making formal the notion of a construction not of a construction problem and, as it turns out, give only a parital axiomatization if these notions are made precise by programs and algorithmic problems in our sense.

In all the cases considered below, the amalgamation property of $algT(\mathcal{A};\mathcal{B})$ follows by classical results. Therefore, the generalized Galois theory developed in Section 2 applies to these theories. (Of course, these are not the only theories of interest in computing which have the amalgamation property. Another prime example is Boolean algebra. The possibility of developing a Galois theory for Boolean algebras was first noticed, to our knowledge, by Hotz [11], who used it to discuss switching circuits.)

The *ordered field of reals*

$$\mathrm{R}_{\leq} = \langle R; \leq, +, \cdot, -, ^{-1}, 0, 1\rangle$$

can be considered (hypothetically) as furnished with the elementary capabilities

$$\mathrm{B}_{\leq} = \{x_i = x_j, x_i \leq x_j; x_i := x_j + x_k, x_i := x_j \cdot x_k, x_i := -x_j,$$

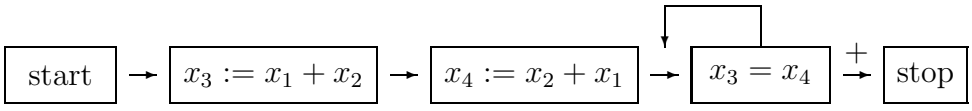
$$x_i := x_j^{-1}(x_j \neq 0), x_i := 0, x_i := 1, \quad i, j, k = 1, 2, \ldots\},$$

abbreviated,

$$\mathrm{B}_{\leq} = \{=, \leq; +, \cdot, -, ^{-1}, 0, 1\}$$

Proposition 14 *algT*$(\mathcal{R}_{\leq}\mathcal{B}_{\leq}) = \{$ *Archimedean ordered field* $\}^{\vdash}$.

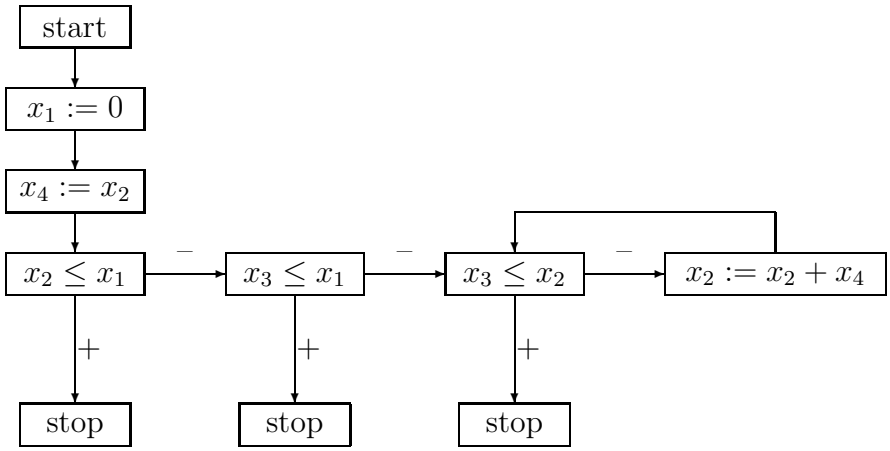
Proof. Observe that the property of being an archimedean ordered field may be formulated by a set of algorithmic formulas in *algL*$(\mathcal{B}_{\leq})$. This is immediate for the axioms of an ordered field. The commutative law of addition, for example, is the termination formula of the following program:



The Archimedean property

$$a > 0 \wedge b > 0 \cdot \supset \bigvee_{n=1}^{\infty} \underbrace{a + a + \cdots + a}_{n} \geq b$$

may be formulated as the termination formula of the following program:



It remains to show that for every (universally quantified) algorithmic formula φ we have

$$\mathcal{R} \models \varphi \text{ iff } \{\text{Archimedean ordered field}\} \vdash \varphi$$

Clearly, if φ is provable in $L_{\omega_1\omega}$ from the axioms of Archimedean ordered fields then φ holds true in all such fields, in particular for the reals. Conversely, suppose that $\mathcal{R} \models \varphi$ and that $\mathcal{F}$ is any Archimedean ordered field. Then, by algebra, $\mathcal{F}$ is a subfield of $\mathcal{R}$ and, φ being a universal formula, $\mathcal{F} \models \varphi$. Hence $\mathcal{F} \models \varphi$ for all Archimedean ordered fields. By completeness of $\vdash$ for $L_{\omega_1\omega}$ it follows that $\{$Archimedean ordered fields$\} \vdash \varphi$.

Slightly less trivial is the case of

$$\mathcal{R}_= = \langle R; +, \cdot, -, ^{-1}, 0, 1\rangle,$$
$$\mathcal{B}_= = \{=; +, \cdot, -, ^{-1}, 0, 1\}$$

arising from our first example by dropping the relation $\leq$ and the elementary capability corresponding to it.

Proposition 15 *$algT(\mathcal{R}_=; \mathcal{B}_=) = \{$ formally real field $\}^\vdash$.*

Proof. The axioms of a field are algorithmic, as noted above. The additional condition (that -1 not be a sum of squares) can just as obviously be formulated as a set of algorithmic formulas of $algL(\mathcal{B}_=)$. Since the reals are formally real, it remains to prove that any algorithmic formula φ which is true for the reals is true for every formally real field. Suppose otherwise, i.e., assume $\varphi(x_1, \ldots, x_n) \in algT(\mathcal{R}_=, \mathcal{B}_=), \mathcal{F}$ formally real, $a_1, \ldots, a_n \in F$ and $\mathcal{F} \models \neg\varphi[a_1, \ldots, a_n]$. Let $\mathcal{F}'$ be the real closure of $\mathcal{F}$; we continue to have $\mathcal{F}' \models \neg\varphi[a_1, \ldots, a_n]$. We have to show that there are $a'_1, \ldots, a'_n \in R$ such that $\mathcal{R} \models \neg\varphi[a'_1, \ldots, a'_n]$. Note the form of $\neg\varphi$:

$$\neg\varphi = \bigvee_{i=1}^{k} \mathrm{Term}_{\pi_i} \wedge \neg\mathrm{Term}_{\pi'_i}$$

Hence

$$\mathcal{F}' \models (\mathrm{Term}_{\pi_i} \wedge \neg\mathrm{Term}_{\pi'_i})[a_1, \ldots, a_n] \text{ for some } i,$$

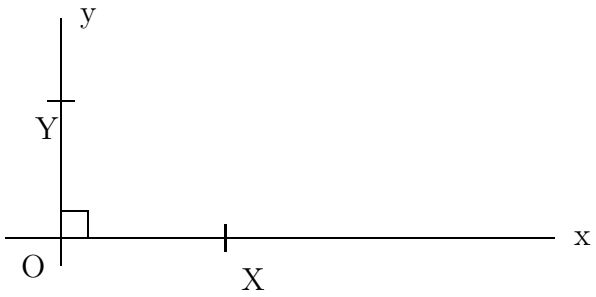
$$\mathcal{F}' \models \Big(\bigvee_{w \in |\sigma(\pi_i)|} \varphi_w \wedge \bigwedge_{w \in |\sigma(\pi'_i)|} \neg\varphi_w \Big)[a_1, \ldots, a_n].$$

each φ_w being a finite set of polynomial equations and inequalities (with integral coefficients). It follows that there is an infinite set P of such equalities and inequalities satisfied by $a_1, \ldots, a_n$ in $\mathcal{F}'$ (and which, if it is satisfied, implies $\neg\varphi[a_1, \ldots, a_n]$). Now, by algebra, if a set P has a solution in any real closed field $\mathcal{F}'$, then it has one in $\mathcal{R}$, which was to be shown.

We now turn to the theory of geometrical constructions. Let

$$\mathcal{G} = \langle \mathcal{P}, \mathcal{L}, ; Inz, Zw, \equiv_s, \equiv_w, //, 0, X, Y, x, y \rangle$$

be the two-dimensional geometry over the reals regarded as a relational structure with two sorts ($\mathcal{P}$: points, $\mathcal{L}$: lines), the relations Inz (of incidence), Zw (betweenness), $\equiv_s, \equiv_w$ (segment and angle congruence as relations on points), // (parallelity), and a fixed 'coordinate system' indicated by



The elementary capabilities with respect to G are formulated with the aid of variables $P, Q, R, \ldots$ for points, $l, g, h \ldots$ for lines. $\mathcal{B}_{aff}$, the *affine constructions* ('ruler alone') are:

$P := O, \quad P := X, \quad P := Y, \quad l := x, \quad l := y;$

$Q := P(g, h)$ point of intersection of g, h (if $g \neq h$);

$l := L(P, Q)$ connecting line of P, Q if P, Q distinct;

$l := L(P, g)$ line parallel to g through P;

$P = Q?, \quad g = h?, \quad g//h?, \quad Inz(P, g)?$

$\mathcal{B}_E$ the *Eichmass constructions* (of Hilbert's Foundations of Geometry), comprise the following additional capabilities:

$Zw(P, Q, R) \quad ? \quad P := E(A, B, C, D),$ the Eichmass construction of finding P such that $Zw(C, D, P) \wedge AB \equiv_s DP \quad$ if $C \neq D$.

Both these construction theories suffer from the adhocness of the fixed coordinate system. More in the spirit of classical constructions would be elementary capabilities which allow arbitrary selections of points and lines. Consider therefore $\mathcal{B}_S$, *selection constructions*, given by the capabilities:

$Q := P(g,h), \quad l := L(P,Q), \quad P := E(A,B,C,D);$

$P = Q?, \quad g = R?, \quad g//h?, \quad Inz(P,g)?, \quad Zw(P,Q,R)?;$

$g := S_0(h)$ selects an arbitrary line,

$P := S_1(g)$ selects an arbitrary point on g,

$P := S_2(g,A)$ selects an arbitrary point on g different from A,

$P := S_3(A,B)$ selects an arbitrary point between A and B,

$P := S_4(g)$ selects an arbitrary point outside g.

The task of axiomatizing $algT(\mathcal{G},\mathcal{B}_{aff})$ is greatly simplified by the availability of analytic geometry for $\mathcal{G}$. These allow us to associate an algorithmic formula φ^* of $algL(\mathcal{B}_=)$ to every $\varphi \in algL(\mathcal{B}_{aff})$ (its translation into terminology of analytic geometry) in a straightforward (purely syntactical) way and such that $\mathcal{G} \models \varphi$ iff $\mathcal{R}_= \models \varphi^*$.

Proposition 16 *$algT(\mathcal{G},\mathcal{B}_{aff}) = \Gamma^{\vdash}$ for every $\Gamma \subseteq algT(\mathcal{G},\mathcal{B}_{aff})$ for which $(\Gamma^*)^{\vdash} \supseteq$ {formally real fields}.*

Proof. $\Gamma^{\vdash} \subseteq algT(G,B_{aff})$ since $algT$ is closed under deduction. Conversely, suppose $\varphi \in algT(\mathcal{G},\mathcal{B}_{aff})$, then $\varphi^* \in algT(\mathcal{R},\mathcal{B}_=)$ by construction of *, hence $\varphi^* \in$ {formally real fields}$^{\vdash}$ and thus $\varphi^* \in (\Gamma^*)^{\vdash}$. Since the syntactic translation * respects proof, we have $\varphi \in \Gamma^{\vdash}$.

To obtain an actual list of axioms Γ, we would now pay close attention to one or another of the standard coordinatization techniques for $\mathcal{G}$ and note down those properties of $\mathcal{G}$ which insure the right algebraic characterization of the field constituted by, say, the points on the line x.

A similar argument allows us to handle the axiomatization of Eichmass construction geometry. Let

$$\mathcal{B}_{eucl} = \{=, \leq; +, \cdot, -, ^{-1}, 0, 1, ^{+}\sqrt{(x^2+y^2)}\}$$

be the list of capabilities which extend $\mathcal{B}_{\leq}$ by the operation of extracting the positive square root of the sum of two squares. We have again a syntactic translation * of

$algL(\mathcal{B}_E)$ into $algL(\mathcal{B}_{eucl})$ with the above properties. Let us call a field *Euclidean*, if it contains $\sqrt{(a^2+b^2)}$ for any a, b of the field; this property is clearly in $algL(\mathcal{B}_{eucl})$.

Proposition 17 *$algT(\mathcal{G}, \mathcal{B}_E) = \Gamma^{\vdash}$ for every $\Gamma \subseteq algT(\mathcal{G}, \mathcal{B}_E)$ for which $(\Gamma^*)^{\vdash} \supseteq \{$archimedean ordered euclidean fields$\}$.*

Proof. Same argument as for the previous proposition.

We now turn to geometries with the selection operations. In the presentation of $\mathcal{G}$ as a two-sorted relational structure, we have (for $\mathcal{B}_E$ and $\mathcal{B}_S$) added some features which are not strictly 'geometric' in nature. For $\mathcal{B}_E$ we added a coordinate system

$$O, X, Y, x, y$$

and (corresponding capabilities); for $\mathcal{B}_S$ we added the selection operations

$$S_0, S_1, S_2, S_3, S_4,$$

which also suffer from a certain degree of arbitrariness.

Lemma 18 *If π is a program over $\mathcal{B}_S$ and π is invariant under reinterpretation of S_0, S_1, S_2, S_3, S_4, then there exists π' over $\mathcal{B}_E$ such that π and π' compute the same partial function (in any fixed $\mathcal{C}$).*

Proof. Because of the invariance under the choice of S_0 and S_4 it is sufficient to provide subroutines over $\mathcal{B}_E$ for the elementary capabilities involving the S_i. This is easily done.

Lemma 19 *If π is a program over $\mathcal{B}_E$ and π is invariant under reinterpretation of O, X, Y, x, y then there exists π' over $\mathcal{B}_S$ such that π and π' compute the same partial function (in any fixed $\mathcal{C}$).*

Proof. Again by providing subroutines.

The coordinate-free and selection-free algorithmic language of Eichmass constructions is defined by

$$constrL = algL(\mathcal{B}_E) \cap algL(\mathcal{B}_S)$$

The obvious task is to axiomatize the set of $\varphi \in constrL$ which are true for $\mathcal{G}$, call this $constrT$.

Proposition 20 $constrT = algT(\mathcal{C}, \mathcal{B}_E) \cap constrL$.

Proof. It follows from the above lemmas that

$$constrL \cap \bigcap_{reinterpret.} algT(\mathcal{C}, \mathcal{B}_E) = constrL \cap \bigcap_{reinterpret.} algT(\mathcal{C}, \mathcal{B}_s)$$

Since reinterpretations of O, X, Y, x, y simply introduce a similarity transformation, we have

$$constrL \cap \bigcap_{reinterpret.} algT(\mathcal{C}, \mathcal{B}_E) = constrL \cap algT(\mathcal{C}, \mathcal{B}_E)$$

It is clear that similar methods can be used to discuss various other construction geometries that have been studied in great detail in the past. We hope that some mathematicians may be tempted to do this and thereby give the theory of geometrical constructions a long missing formal precision.

5 Some historical remarks

One of the most striking experiences that this author had in his involvement with Galois theory concerns the extent to which the way the theory is presented has been changed ever since Artin [1]. Indeed, we did in fact reinvent the definition of a Galois group as the group of permutations of variable symbols that do not change the provability of expressions and were very happy for this characterization (closer to actual manipulative aspects and therefore more promising from a computing point of view). It was then pointed out to us that even as late as O. Perron's algebra [18], this was at many universities the standard way to present Galois theory. Of course, ours is still a true generalization and one in which the formal aspects are better profiled.

Not surprisingly, therefore, other attempts to generalize Galois theory in a universal algebra setting started from the concept of an algebraic element. For this concept, there have been a number of formulations, mainly by Robinson [19], Jónsson [12], Morley [16] and also by Park [17]. These were recently very nicely brought into relation by Bacsich [2]. The merit to have observed the central importance of the amalgamation property in Galois theory clearly goes to Jónsson; the concept itself is due to Fraïssé.

Bibliography

[1] E. Artin. *Galois Theory*, Notre Dame Mathematical Lectures 2 (1942).

[2] P. D. Bacsich. Defining algebraic elements, *Journal of Symbolic Logic* 38 (1973).

[3*] E. Engeler. *Formal Languages: Automata and Structures.* Markham, Chicago (1968), 81 pp.

[4*] E. Engeler. Proof theory and the accuracy of computations. In: *Symposium on Automatic Demonstration* (eds M. Laudet et al.), LN Math. 125, Springer (1970), 62-71.

[5*] E. Engeler. Structure and meaning of elementary programs. In: *Symposium on Semantics of algorithmic Languages* (ed. E. Engeler), LN Math. 188, Springer (1971), 89-101.

[6] R. W. Floyd. Assigning meaning to programs. In: *Mathematical Aspects of Computer Science* (ed J. T. Schwarz) Proceedings of Symposia in Applied Mathematics 19 (Am. Math. Soc. Providence, R.I.) (1967), 19-32.

[7] M. Grabowski. The set of all tautologies of the zero-order algorithmic logic is decidable, *Bulletin de l'Academie Polonaise des Sciences. Serie des Sciences Mathématiques Astronomiques et Physiques* 20 (1972), 575-582.

[8] J. Gruska. A characterization of context-free languages, *Journal of Computer and Systems Sciences* 5 (1971), 353-364.

[9] C. A. R. Hoare. An axiomatic basis for computer programming, *Communications of the Association for Computing Machinery* 12 (1969), 576-583.

[10] C. A. R. Hoare and N. Wirth. An axiomatic definition of the programming language Pascal, *Acta Informatica* 2 (1973), 335-355.

[11] G. Hotz. Zur Reduktionstheorie der Boole'schen algebra. In: *Kolloquium über Schaltkreis und Schaltwerktheorie* (eds Unger and Peschel), Birkhäuser, Basel (1962).

[12] B. Jónsson. Algebraic extensions of relational systems, *Mathematica Scandinavica* 11 (1962), 179-205.

[13] A. Kreczmar. The set of all tautologies of algorithmic logic is hyperarithmetical, *Bulletin de l'Academie Polonaise des Sciences. Serie des Sciences Mathématiques Astronomiques et Physiques* 21 (1971), 781-783.

[14] R. Milner. Implementation and applications of Scott's logic for computable functions. In: *Proceedings of the ACM Conference on Proving Assertions about Programs*, New Mexico (1972), 1-6.

[15] G. Mirkowska. On formalized systems of algorithmic logic, *Bulletin de l'Academie Polonaise des Sciences. Serie des Sciences Mathématiques Astronomiques et Physiques* 21 (1971), 421-428.

[16] M. Morley. Categoricity in power, *Transactions of the American Mathematical Society* 114 (1965), 514-538.

[17] D. Park. Set theoretic constructions in model theory, Ph.D. Thesis MIT (1964).

[18] O. Perron. *Algebra*, 2 Vols. Göschen Verlag, (1927).

[19] A. Robinson. *On the Metamathematics of algebra*, North-Holland, Amsterdam (1951).

[20] A. Salwicki. Formalized algorithmic languages, *Bulletin de l'Academie Polonaise des Sciences. Serie des Sciences Mathematiques Astronomiques et Physiques* 18 (1970), 227-232.

[21] A. Salwicki. On the equivalence of FS-expressions and programs, *Bulletin de l'Academie Polonaise des Sciences. Serie des Sciences Mathematiques Astronomiques et Physiques* 18 (1970), 275-278.

[22] D. S. Scott and C. Strachey. Toward a mathematical semantics for computer languages. In: *Proceedings of the Symposium on Computers and Automata*, New York, Microwave Research Institute Series 21(1971), 19-46.

[23] P. Suppes and N. Moler. Quantifier-free axioms for constructive plane geometry, *Computer Mathematics* 20 (1968), 143-152.

[24] A. Tarski. What is elementary geometry? In: *The Axiomatic Method* (eds L. Henkin, P. Suppes and A. Tarski), North-Holland, Amsterdam (1959), 16-29.