Algorithmic logic and its applications in the theory of programs II*

GRAŻYNA MIRKOWSKA

University of Warsaw

Received November 5, 1975

AMS Categories: 68A05, 02C93

6. Diagrams of formulas

In this section we shall consider another axiomatization of the set of tautologies of the algorithmic language $\mathscr{L}$. We shall follow Gentzen's ideas. At first we recall some auxiliary notions.

DEFINITION 1. Let Γ_1, Γ_2 denote finite sequences (the empty sequence is admitted) of formulas in $\mathscr{L}$. Every expression of the form $\Gamma_1 \to \Gamma_2$ will be called a *sequent*.

A sequent S of the form $a_1, \ldots, a_n \to \beta_1, \ldots, \beta_m$ is called *indecomposable* if and only if every formula a_i, β_j $(i = 1, \ldots, n,\ j = 1, \ldots, m)$ is a propositional variable or is of the form $\varrho(\tau_1, \ldots, \tau_n)$ where $\tau_i \in T$.

DEFINITION 2. A sequent S is said to be an *axiom* if and only if there exist indices i and j $(1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m)$ such that a_i and β_j are identical, or if $\mathbf{1} \in \{\beta_1, \ldots, \beta_m\}$ or $\mathbf{0} \in \{a_1, \ldots, a_n\}$.

DEFINITION 3. By a *scheme of inference* we shall understand a pair $\{S, S_0\}$ of sequents, a triple $\{S, S_0; S_1\}$ or an enumerable sequence of sequents $\{S, S_0; S_1; S_2; \ldots\}$. Such a scheme will be written in the form

$$\frac{S}{S_0} \quad \text{or} \quad \frac{S}{S_0;\ S_1} \quad \text{or} \quad \frac{S}{\{S_i\}_{i \in \mathcal{N}}}.$$

The sequent S is called the *conclusion*; the sequent S_0 in the first case, S_0, S_1 in the second, and $S_0, S_1, \ldots$ in the third case the *premises*.

---

* The paper is a continuation of *Algorithmic logic and its applications in the theory of programs I*, Fundamenta Informaticae, this volume, pp. 1–17. All unexplained notions and denotations can be found there.

In the sequel we shall consider three groups of schemes:

Group I

1A $$\frac{\Gamma_1, s_1 \ldots s_k a_0, \Gamma_2 \to \Gamma_3}{s_1 \ldots s\ _{-1}\overline{s_k a_0}, \Gamma_1, \Gamma_2 \to \Gamma_3}$$

1B $$\frac{\Gamma_1 \to \Gamma_2, s_1 \ldots s_k a_0, \Gamma_3}{\Gamma_1 \to s_1 \ldots s_{k-1}\overline{s_k a_0}, \Gamma_2, \Gamma_3}$$

2A $$\frac{\Gamma_1, s\varrho(\tau_1, \ldots, \tau_n), \Gamma_2 \to \Gamma_3}{s\chi\big(\varrho(\tau_1, \ldots, \tau_n)\big), \Gamma_1, \Gamma_2 \to \Gamma_3}$$

2B $$\frac{\Gamma_1 \to \Gamma_2, s\varrho(\tau_1, \ldots, \tau_n), \Gamma_3}{\Gamma_1 \to s\chi\big(\varrho(\tau_1, \ldots, \tau_n)\big), \Gamma_2, \Gamma_3}$$

3A $$\frac{\Gamma_1, s \neg \alpha, \Gamma_2 \to \Gamma_3}{\Gamma_1, \Gamma_2 \to s\alpha, \Gamma_3}$$

3B $$\frac{\Gamma_1 \to \Gamma_2, s \neg \alpha, \Gamma_3}{\Gamma_1, s\alpha \to \Gamma_2, \Gamma_3}$$

4A $$\frac{\Gamma_1, s(\alpha \cap \beta), \Gamma_2 \to \Gamma_3}{s\alpha, s\beta, \Gamma_1, \Gamma_2 \to \Gamma_3}$$

5B $$\frac{\Gamma_1 \to \Gamma_2, s(\alpha \cup \beta), \Gamma_3}{\Gamma_1 \to s\alpha, s\beta, \Gamma_2, \Gamma_3}$$

6B $$\frac{\Gamma_1 \to \Gamma_2, s(\alpha \Rightarrow \beta), \Gamma_3}{s\alpha, \Gamma_1 \to s\beta, \Gamma_2, \Gamma_3}$$

7A $$\frac{\Gamma_1, s\bigcap K\alpha, \Gamma_2 \to \Gamma_3}{s\bigcap K(K\alpha), \Gamma_1, s\alpha, \Gamma_2 \to \Gamma_3}$$

8B $$\frac{\Gamma_1 \to \Gamma_2, s\bigcup K\alpha, \Gamma_3}{\Gamma_1 \to s\bigcup K(K\alpha), \Gamma_2, s\alpha, \Gamma_3}$$

9A $$\frac{\Gamma_1, s(\circ[KM]\alpha), \Gamma_2 \to \Gamma_3}{s\big(K(M\alpha)\big), \Gamma_1, \Gamma_2 \to \Gamma_3}$$

9B $$\frac{\Gamma_1 \to \Gamma_2, s(\circ[KM]\alpha), \Gamma_3}{\Gamma_1 \to s\big(K(M\alpha)\big), \Gamma_2, \Gamma_3}$$

10A

$$\frac{\Gamma_1, s(\veebar[\delta KM]\alpha), \Gamma_2 \to \Gamma_3}{s\big((\delta \cap K\alpha) \cup (\neg\delta \cap M\alpha)\big), \Gamma_1, \Gamma_2 \to \Gamma_3}$$

10B

$$\frac{\Gamma_1 \to \Gamma_2, s(\veebar[\delta KM]\alpha), \Gamma_3}{\Gamma_1 \to s\big((\delta \cap K\alpha) \cup (\neg\delta \cap M\alpha)\big), \Gamma_2, \Gamma_3}$$

11A

$$\frac{\Gamma_1, s(*[\delta K]\alpha), \Gamma_2 \to \Gamma_3}{s\bigcup \veebar\big[\delta K[\ ]\big](\neg\delta \cap \alpha), \Gamma_1, \Gamma_2 \to \Gamma_3}$$

11B

$$\frac{\Gamma_1 \to \Gamma_2, s(*[\delta K]\alpha), \Gamma_3}{\Gamma_1 \to s\bigcup \veebar\big[\delta K[\ ]\big](\neg\delta \cap \alpha), \Gamma_2, \Gamma_3}$$

Group II

5A $$\frac{\Gamma_1, s(\alpha \cup \beta), \Gamma_2 \to \Gamma_3}{s\alpha, \Gamma_1, \Gamma_2 \to \Gamma_3;\ s\beta, \Gamma_1, \Gamma_2 \to \Gamma_3}$$

4B $$\frac{\Gamma_1 \to \Gamma_2, s(\alpha \cap \beta), \Gamma_3}{\Gamma_1 \to s\alpha, \Gamma_2, \Gamma_3;\ \Gamma_1 \to s\beta, \Gamma_2, \Gamma_3}$$

6A $$\frac{\Gamma_1, s(\alpha \Rightarrow \beta), \Gamma_2 \to \Gamma_3}{\Gamma_1, \Gamma_2 \to s\alpha, \Gamma_3;\ s\beta, \Gamma_1, \Gamma_2 \to \Gamma_3}$$

Group III

8A $$\frac{\Gamma_1, s\bigcup K\alpha, \Gamma_2 \to \Gamma_3}{\{s(K^i\alpha), \Gamma_1, \Gamma_2 \to \Gamma_3\}_{i\in\mathcal{N}}}$$

7B $$\frac{\Gamma_1 \to \Gamma_2, s\bigcap K\alpha, \Gamma_3}{\{\Gamma_1 \to s(K^i\alpha), \Gamma_2, \Gamma_3\}_{i\in\mathcal{N}}}.$$

In all the above schemes $\Gamma_1, \Gamma_2, \Gamma_3$ denote any sequents, s denotes any sequence of substitutions, K, M — any programs, and α, β, δ any formulas such that $\alpha, \beta \in FSF$, $\delta \in F$.

DEFINITION 4. By a *diagram of a formula* a_0 we shall mean an ordered pair $(\mathscr{D}, d)$ where $\mathscr{D}$ is a tree and d is a mapping which to every element of the tree assigns a certain nonempty sequent. The tree $\mathscr{D}$ and the mapping d are defined by induction on the level l of $\mathscr{D}$ as follows:

1. if $l = 0$ then the only element of this level is $\emptyset$ and $d(\emptyset)$ is equal to the sequent $\to a_0$.

Suppose that we have defined all the elements of the tree $\mathscr{D}$ up to the level not higher than n. Now we define the elements of the level $n+1$. Let $c = (i_1, \ldots, i_n)$ and let the sequent $d(c)$ be defined;

2. if $d(c)$ is an indecomposable sequent or an axiom then none of the elements $c' = (i_1, \ldots, i_n, k)$, $k \in \mathscr{N}$ belongs to $\mathscr{D}$; c and $d(c)$ are called the *end-element* and the *end-sequent* of the tree $\mathscr{D}$;

3. the sequent $d(c)$: $\Gamma \to \nabla$ is neither indecomposable nor an axiom. We consider two cases:

Case 1: n is an even number.

A. If the sequence ∇ contains only atomic formulas, then

$$(i_1, \ldots, i_n, 0) \in \mathscr{D} \quad \text{and} \quad d(i_1, \ldots, i_n, 0) = d(c);$$

B. If a is the first on the right-hand side nonatomic formula in ∇, then we consider different forms of the formula a:

1. if the sequent $d(c)$ is the conclusion in the scheme of the group IB, then $(i_1, \ldots, i_n, 0) \in \mathscr{D}$ and $d(i_1, \ldots, i_n, 0)$ is equal to the only premise in this scheme;

2. if the sequent $d(c)$ is the conclusion in the scheme of the group IIB, then $(i_1, \ldots, i_n, 0)$ and $(i_1, \ldots, i_n, 1)$ belong to $\mathscr{D}$ and $d(i_1, \ldots, i_n, 0)$, $d(i_1, \ldots, i_n, 1)$ are the first and the second premise in the scheme;

3. if the sequent $d(c)$ is the conclusion in the scheme of the group IIIB, then $(i_1, \ldots, i_n, k)$ are in $\mathscr{D}$ for every $k \in \mathscr{N}$ and $d(i_1, \ldots, i_n, k)$ is the kth premise in this scheme.

Case 2: n is an odd number.

Points A and B in the above definition have to be changed as follows: the sequence ∇ is replaced by Γ and the groups IB, IIB, IIIB by IA, IIA, IIIA.

From this definition it follows immediately that for every formula its diagram is defined in an unambigous way.

Let Γ be a sequence $\gamma_1, \ldots, \gamma_n$ then by $\bigwedge \Gamma$ we shall understand the formula $\big(\gamma_1 \cap (\gamma_2 \cap \ldots (\gamma_{n-1} \cap \gamma_n) \ldots)\big)$ and by $\bigvee \Gamma$ — the formula $\big(\gamma_1 \cup (\gamma_2 \cup \ldots (\gamma_{n-1} \cup \gamma_n) \ldots)\big)$. If Γ is the empty sequence, then $\bigwedge \Gamma$ denotes any valid formula and $\bigvee \Gamma$ — any false formula.

If S is a sequent of the form $\Gamma_1 \to \Gamma_2$ then by δ_S we shall denote the formula $\bigwedge \Gamma_1 \Rightarrow \bigvee \Gamma_2$.

LEMMA 1. *For every realization R of the language $\mathscr{L}$ and for every valuation v the following conditions hold:*

1. *if $\{S, S_0\}$ is a scheme of inference in the group* I, *then*

$$\delta_{SR}(v) = \delta_{S_0R}(v);$$

2. *if $\{S, S_0; S_1\}$ is a scheme of inference in the group* II, *then*

$$\delta_{SR}(v) = \delta_{S_0R}(v) \wedge \delta_{S_1R}(v);$$

3. *if $\{S, S_0; S_1; \ldots\}$ is a scheme in the group* III, *then*

$$\delta_{SR}(v) = \text{g. l. b.}\,\big(\delta_{S_iR}(v)\big)_{i\in\mathscr{N}}.$$

LEMMA 2. *A formula a_0 is a tautology if and only if the diagram of a_0 is finite and every end-sequent is an axiom.*

Proof: Let $(\mathscr{D}, d)$ be the diagram of the formula a_0.

Part one of the proof. Let us assume that $(\mathscr{D}, d)$ is finite, i.e., $\mathscr{D}$ is finite (see Definition 3 in § 5), and all end-sequents are axioms. Note that if S is an axiom (see Definition 2) then for every R and for every valuation v we have $\delta_{SR}(v) = \mathbf{1}$. Suppose that for all elements $c' = (i_1, \ldots, i_n, j)$ of the level $n+1$ we have $\delta_{d(c')R}(v) = \mathbf{1}$. Let $c = (i_1, \ldots, i_n)$; then c is not an end-element of the tree $\mathscr{D}$, so $d(c) = S$ is the conclusion in a scheme of inference.

1. If S is the conclusion in a scheme of the group I, then by Lemma 1 we have $\delta_{SR}(v) = \delta_{S'R}(v)$ where $S' = d(c')$ and so $\delta_{SR}(v) = \mathbf{1}$.

2. If S is the conclusion in a scheme of the group II, then the tree contains $c' = (i_1, \ldots, i_n, 0)$ and $c'' = (i_1, \ldots, i_n, 1)$. By Lemma 1 we have $\delta_{SR}(v) = \delta_{d(c')R}(v) \wedge \delta_{d(c'')R}(v)$. Since $\mathbf{1} = \delta_{d(c')R}(v) = \delta_{d(c'')R}(v)$, we have $\delta_{SR}(v) = \mathbf{1}$.

3. If S is the conclusion in a scheme of the group III, then $\mathscr{D}$ contains all elements of the set $\{i_1, \ldots, i_n, j\}_{j\in\mathscr{N}}$. By the inductive assumption, $\delta_{S_jR}(v) = \mathbf{1}$ for $S_j = d(i_1, \ldots, i_n, j)$, and so, by Lemma 1, $\delta_{SR}(v) = \mathbf{1}$.

We have thus proved that if c is an element of $\mathscr{D}$ and $S = d(c)$, then $\delta_{SR}(v) = \mathbf{1}$ for every realization R and any valuation v. Hence a_0 is a tautology in algorithmic logic.

Part two of the proof. Let us suppose that a_0 is a tautology and the diagram $(\mathscr{D}, d)$ of the formula a_0 is finite, and assume that there exists an end-sequent S that is not an axiom. Since S is an end-sequent, S is indecomposable. Denote by P the set of all formulas that occur in the predecessor and by N the set of all formulas that occur in the successor of the sequent S. Now we shall define a realization R_0 as follows:

$$\varrho_{R_0}(\tau_1, \ldots, \tau_n) = \begin{cases} \mathbf{1} & \text{if} \quad \varrho(\tau_1, \ldots, \tau_n) \in P, \\ \mathbf{0} & \text{if} \quad \varrho(\tau_1, \ldots, \tau_n) \in N; \end{cases}$$

$$\psi_{R_0}(\tau_1, \ldots, \tau_n) = \psi(\tau_1, \ldots, \tau_n);$$

for every n-argument predicate $\varrho \in P_n$ and every n-argument functor $\psi \in \Phi_n$ and for any terms $\tau_1, \ldots, \tau_n$ in the set T.

Since the sets P and N are disjoint, R_0 indeed determines a realization of the language $\mathscr{L}$. Let ι denote the valuation defined as follows: $\iota(x) = x$ for $x \in V_i$, and for every $a \in V_0$

$$\iota(a) = \begin{cases} 1 & \text{for} \quad a \in P, \\ 0 & \text{for} \quad a \notin P. \end{cases}$$

By the definition of R_0, $\delta_{SR_0}(\iota) = \mathbf{0}$. Let $d(c) = S$ and let G be a branch such that $c \in G$. By Lemma 1 for every sequent $S' = d(c')$ where $c' \in G$ we have $\delta_{SR}(\iota) = \mathbf{0}$. Since $\emptyset \in G$, then $\alpha_{0R_0}(\iota) = \mathbf{0}$, which contradicts our assumption that α_0 is a tautology.

Part three of the proof. Let $(\mathscr{D}, d)$ be the diagram of the formula α and let G be its infinite branch. We shall prove that α is not a tautology. Denote by N the set of all formulas that occur in the successors and by P the set of all formulas that occur in the predecessors of all sequents $S = d(c)$ where $c \in G$. Let F^a denote the set of all atomic formulas from the set $P \cup N$. If $\alpha \in F^a$ then $\alpha \in P - N$ or $\alpha \in N - P$, since in the opposite case we could find a sequent S such that α is in the predecessor and in the successor of S, i.e. the sequent S is an axiom and the branch G is finite. Let R_0 be a realization of the language $\mathscr{L}$ in the set of all classical terms T and in the Boolean algebra B_0 such that R_0 associates to every m-argument functor φ an m-argument operation φ_{R_0} in T, $\varphi_{R_0}(\tau_1, \ldots, \tau_n) = \varphi(\tau_1, \ldots, \tau_n)$, and to every n-argument predicate ϱ the characteristic function of an n-argument relation ϱ_{R_0},

$$\varrho_{R_0}(\tau_1, \ldots, \tau_n) = \begin{cases} 1 & \text{if} \quad \varrho(\tau_1, \ldots, \tau_n) \in P, \\ 0 & \text{otherwise}. \end{cases}$$

Let ι denote the valuation such that $\iota(x) = x$ for $x \in V_i$ and

$$\iota(a) = \begin{cases} 1 & \text{if} \quad a \in P, \\ 0 & \text{in opposite case;} \end{cases} \qquad \text{for all } a \in V_0.$$

See that for every $\tau \in T$, $\tau_{R_0}(\iota) = \tau$. We shall prove now that for every $\alpha \in P \cup N$ we have

$$\alpha_{R_0}(\iota) = \begin{cases} 1 & \text{if} \quad \alpha \in P, \\ 0 & \text{if} \quad \alpha \in N. \end{cases} \tag{1}$$

The proof is by induction.

First we formulate

Definition 5. Let us consider the set of pairs (α_1, α_2), where $\alpha_1, \alpha_2 \in FSF$ such that α_1 is identical with α_2 or the pair (α_1, α_2) is of one of the following forms:

1. $((s_1 \ldots s_k \varrho(\tau_1, \ldots, \tau_n)), (s_1 \ldots s_{k-1} \overline{s_k \varrho(\tau_1, \ldots, \tau_n)}))$ $\tau_i \in T$, $i = 1, \ldots, n$,
2. $(s\varrho(\tau_1, \ldots, \tau_n), s\chi(\varrho(\tau_1, \ldots, \tau_n)))$ $\tau_i \in FST - T$, $i = 1, \ldots, n$,
3. $(s \neg \beta, s\beta)$,

4a. $(s(\gamma \cup \beta), s\gamma)$, 4b. $(s(\gamma \cup \beta), s\beta)$,

5a. $(s(\gamma \cap \beta), s\gamma)$, 5b. $(s(\gamma \cap \beta), s\beta)$,

6a. $(s(\gamma \Rightarrow \beta), s\gamma)$, 6b. $(s(\gamma \Rightarrow \beta), s\beta)$,

7. $(s(\circ[KM]\beta), s(K(M\beta)))$,

8a. $(s(\veebar[\beta KM]\gamma), s(\beta \cap K\gamma))$, 8b. $(s(\veebar[\beta KM]\gamma), s(\neg\beta \cap M\gamma))$,

9i. $(s*([\gamma K]\beta), s(\veebar[\gamma K[\ ]]^i\beta))$, $i \in \mathcal{N}$,

10i. $(s\bigcup K\beta, s(K^i\beta))$, $i \in \mathcal{N}$,

11i. $(s\bigcap K\beta, s(K^i\beta))$, $i \in \mathcal{N}$,

where as usual K, M are programs, β, γ are formulas and s is a sequence of substitutions $s_1 \ldots s_k$, $k \in \mathcal{N}$.

By $>$ we shall denote the transitive closure of the set defined above. If $\alpha > \beta$, then we shall say that β is *submitted to the formula* α.

Let us notice that the binary relation $>$ is an ordering in FSF with the minimality property, i.e. any subset Z of formulas, $Z \subset FSF$, contains a minimal element.

Come back to the proof of condition (1). If α is an atomic formula, then by the definition of realization R_0 and valuation ι, (1) holds.

Let α be an arbitrary fixed formula and assume that (1) holds for all formulas that are submitted to the formula α. We have to prove that (1) holds also for α. We shall consider only some of the forms of the formula α. The proofs in other cases are similar.

1. If α is of the form $s_1 \ldots s_k \varrho(\tau_1, \ldots, \tau_n)$, then by the scheme of inference 1A the formula $\delta = s_1 \ldots s_{k-1} \overline{s_k \varrho(\tau_1, \ldots, \tau_n)}$ belongs to a sequent of the branch G; moreover if $\alpha \in P$ then $\delta \in P$ and if $\alpha \in N$ then $\delta \in N$. So by the inductive assumption, δ being submitted to α, we have

$$\alpha_{R_0}(\iota) = \begin{cases} 1 & \text{if} \quad \alpha \in P, \\ 0 & \text{if} \quad \alpha \in N. \end{cases}$$

2. If the formula α is of the form $s \neg \beta$, then the formula $s\beta$ belongs to N, provided α belongs to P, and $s\beta$ belongs to P, provided α belongs to N. Since $s\beta$ is submitted to α, by the inductive assumption we have (1).

8a. If the formula α is of the form $s(\veebar[\beta KM]\gamma)$, then:

if $\alpha \in P$ then at least one of the following formulas $s(\beta \cap K\gamma)$, $s(\neg\beta \cap M\gamma)$ is in P;

if $\alpha \in N$, then both formulas $s(\beta \cap K\gamma)$, $s(\neg\beta \cap M\gamma)$ are in N.

By the inductive assumption we have $(s(\beta\cap K\gamma))_{R_0}(\iota)=\mathbf{1}$ or $(s(\neg\beta\cap M\gamma))_{R_0}(\iota)=\mathbf{1}$ in the first case and $(s(\beta\cap K\gamma))_{R_0}(\iota)=\mathbf{0}$ and $(s(\neg\beta\cap M\gamma))_{R_0}(\iota)=\mathbf{0}$ in the second case. So (1) holds.

10i. Let α be of the form $s\bigcup K\alpha$. If $\alpha\in N$, then all the formulas $s(\overline{K^i\alpha})$ belong to N by the scheme 8B. Since $s(K^i\alpha)$ is submitted to the formula $s\bigcup K\alpha$, we have $(s\bigcup K\alpha)_{R_0}(\iota)=\text{l.u.b. }\{(sK^i\alpha)_{R_0}(\iota)\}_{i\in\mathcal{N}}=\mathbf{0}$. If $\alpha\in P$, then at least one of the formulas $s(K^i\alpha)$ belongs to P, scheme 8A. So by the inductive assumption $(s\bigcup K\alpha)_{R_0}(\iota)=\text{l.u.b. }(sK^i\alpha)_{R_0}(\iota)=\mathbf{0}$.

By the induction principle (1) holds for all formulas. Now, since $\alpha_0\in N$, we have $\alpha_{0R_0}(\iota)=\mathbf{0}$ which contradicts our assumption that α_0 is a tautology. So we have proved that if α_0 is a tautology, then the diagram is finite. ■

7. The completeness theorem for algorithmic theories

Let $\mathcal{T}$ denote an algorithmic theory $\langle\mathcal{L},\mathcal{C},\mathcal{A}\rangle$.

DEFINITION 1. By $\approx$ we shall denote the equivalence relation in the set of all formulas of the language $\mathcal{L}$ such that $\alpha\approx\beta$ if and only if both formulas $(\alpha\Rightarrow\beta)$ and $(\beta\Rightarrow\alpha)$ are theorems in $\mathcal{T}$, i.e. $(\alpha\Leftrightarrow\beta)$ is in $\mathcal{C}(\mathcal{A})$.

THEOREM 1. *For every formulas* $\alpha,\beta\ \alpha',\beta'$ *and for every program* K, *if* $\alpha\approx\alpha'$ *and* $\beta\approx\beta'$, *then* $(\alpha\cup\beta)\approx(\alpha'\cup\beta')$, $(\alpha\cap\beta)\approx(\alpha'\cap\beta')$, $\neg\alpha\approx\neg\alpha'$, $K\alpha\approx K\alpha'$, $\bigcup K\alpha\approx\bigcup K\alpha'$, $\bigcap K\alpha\approx\bigcap K\alpha'$.

Proof: The proof of the first three equivalences is similar to [10] and is omitted. The proof of the fourth is by induction on the length of program.

1. Let K be an element of the set of substitutions S. By axiom T13, $\mathbf{1}$ is in $\mathcal{C}(\mathcal{A})$. Since $(\alpha\Rightarrow\alpha')$ and $(\alpha'\Rightarrow\alpha)$ are theorems, then $K(\alpha'\Rightarrow\alpha)$ and $K(\alpha\Rightarrow\alpha')$ are theorems, too. By T21 and *modus ponens*, $K\alpha\approx K\alpha'$.

Let us assume that the theorem holds for programs M, N and any formulas such that $\alpha\approx\alpha'$.

2. Consider the program $\circ[MN]$. If $\alpha\approx\alpha'$, then by the inductive assumption the formulas $(N\alpha\Rightarrow N\alpha')$ and $(N\alpha'\Rightarrow N\alpha)$ are in $\mathcal{C}(\mathcal{A})$ and therefore $(M(N\alpha)\Leftrightarrow M(N\alpha'))$ is in $\mathcal{C}(\mathcal{A})$. By axiom T25 and *modus ponens* we have $(\circ[MN]\alpha\Leftrightarrow\circ[MN]\alpha')$.

3. Consider the program $\veebar[\gamma MN]$. If $\alpha\approx\alpha'$, then by the inductive assumption $(N\alpha\Leftrightarrow N\alpha')$ and $(M\alpha\Leftrightarrow M\alpha')$. So the formulas $((\gamma\cap M\alpha)\Leftrightarrow(\gamma\cap M\alpha'))$, $((\neg\gamma\cap N\alpha)\Leftrightarrow(\neg\gamma\cap N\alpha'))$ are in $\mathcal{C}(\mathcal{A})$. By axioms T2, T4 and T26 and by *modus ponens* we have $(\veebar[\gamma MN]\alpha\Leftrightarrow\veebar[\gamma MN]\alpha')$.

4. It remains to consider the program $*[\gamma M]$. Let $\alpha\approx\alpha'$ then by assumption $(M\alpha\Leftrightarrow M\alpha')$; so by 2, $(M^i\alpha\Leftrightarrow M^i\alpha')$ for every $i\in\mathcal{N}$. By axioms T21, T26 we obtain $((\veebar[\gamma M[\ ]]^i(\alpha\cap\neg\gamma))\Leftrightarrow(\veebar[\gamma M[\ ]]^i(\alpha'\cap\neg\gamma)))$, $i\in\mathcal{N}$. Hence by r1 and r3 the formula $\bigcup\veebar[\gamma M[\ ]](\alpha\cap\neg\gamma)\approx$

$\approx \bigcup \veebar [\gamma M[\,]](\alpha' \cap \neg\gamma)$ is in $\mathscr{C}(\mathscr{A})$, and consequently (axiom T27) we have $*([\gamma M]\alpha \Leftrightarrow *[\gamma M]\alpha')$ is in $\mathscr{C}(\mathscr{A})$. ∎

Now let us consider the set $FSF/\approx$. The set of all formulas such that $\alpha \approx \beta$ will be denoted by $\|\beta\|$.

THEOREM 2. *The algebra* $\langle FSF/\approx, \wedge, \vee, \to, -\rangle$ *is a Boolean algebra and for all formulas* α, β *the following equations hold:*

1. $\|\alpha\| \vee \|\beta\| = \|\alpha \cup \beta\|$,
2. $\|\alpha\| \wedge \|\beta\| = \|\alpha \cap \beta\|$,
3. $\|\alpha\| \to \|\beta\| = \|\alpha \Rightarrow \beta\|$,
4. $-\|\alpha\| = \|\neg\alpha\|$,
5. $\|\alpha\| \leqslant \|\beta\|$ *if and only if* $(\alpha \Rightarrow \beta)$ *is a theorem in* $\mathscr{T}$,
6. $\|\alpha\| = \mathbf{1}$ *if and only if* α *is a theorem in the theory* $\mathscr{T}$,
7. $\|\alpha\| \neq \mathbf{0}$ *if and only if* $\neg\alpha$ *is not a theorem in* $\mathscr{T}$.

Moreover, any program can be regarded as an operation in the algebra $FSF/\approx$ and by means of this operation we can define a generalized operation in the set $FSF/\approx$.

THEOREM 3. $FSF/\approx$ *is a generalized algebra and for every program* K *and every formula* α

(i) $K\|\alpha\| = \|K\alpha\|$,

(ii) l.u.b. $(K^i\|\alpha\|)_{i\in\mathscr{N}} = \left\|\bigcup K\alpha\right\|$,

(iii) g.l.b. $(K^i\|\alpha\|)_{i\in\mathscr{N}} = \left\|\bigcap K\alpha\right\|$.

Proof: By axioms T23, T24 we have, for every natural i, $(K^i\alpha \Rightarrow \bigcup K\alpha) \in \mathscr{C}(\mathscr{A})$ and $(\bigcap K\alpha \Rightarrow K^i\alpha) \in \mathscr{C}(\mathscr{A})$. By Theorem 2, $\|K^i\alpha\| \leqslant \left\|\bigcup K\alpha\right\|$ and $\left\|\bigcap K\alpha\right\| \leqslant \|K^i\alpha\|$. Let us suppose that there exist formulas γ and δ such that $\|K^i\alpha\| \leqslant \|\gamma\|$ and $\|\delta\| \leqslant \|K^i\alpha\|$. By inference rules r3 and r4 we have $\left\|\bigcup K\alpha\right\| \leqslant \|\gamma\|$ and $\|\delta\| \leqslant \left\|\bigcap K\alpha\right\|$. Hence $\left\|\bigcup K\alpha\right\|$ is the least upper bound of the set $\{\|K^i\alpha\|\}_{i\in\mathscr{N}}$ and $\left\|\bigcap K\alpha\right\|$ is the greatest lower bound of the set $\{\|K^i\alpha\|\}_{i\in\mathscr{N}}$. ∎

Let $\mathscr{Z}$ be the cartesian product $FS \times FS \times FSF$ and let us denote by Q the set defined in the following way: if $a_{iz} = b_{iz} = \|MK^i\alpha\|$, where $z = (M, K, \alpha) \in \mathscr{Z}$, $i \in \mathscr{N}$ and $a_z = \text{l.u.b.}\,(a_{iz})_{i\in\mathscr{N}}$, $b_z = \text{g.l.b.}\,(b_{iz})_{i\in\mathscr{N}}$, then a_z and b_z are elements of the set Q. The set Q defined in such a way is denumerable since the set of formulas in the set $\{M(K^i\alpha)\}_{i\in\mathscr{N}}$ is denumerable. Hence, by Lemma [10], II, 9.2, there exists a Q-filter ∇ in $FSF/\approx$ such that ∇ is a maximal filter and for every $a_z, b_z \in Q$, if $a_z \in \nabla$ then there exists $j \in \mathscr{N}$ such that $a_{jz} \in \nabla$ and if $b_z \notin \nabla$ then there exists $j \in \mathscr{N}$ such that $b_{jz} \notin \nabla$. Let h denote the natural homomorphism from the algebra $FSF/\approx$ into the two-element Boolean algebra B_0,

$$h(\|\alpha\|) = \begin{cases} \mathbf{1} & \text{if} \quad \|\alpha\| \in \nabla, \\ \mathbf{0} & \text{if} \quad \|\alpha\| \notin \nabla. \end{cases}$$

DEFINITION 2. By the *canonical realization* of the language $\mathscr{L}$ we shall mean the realization R_0 in the set of all classical terms T and in the Boolean algebra B_0 such that for any terms $\tau_1, \ldots, \tau_n$ from the set T, the realization R_0 associates to every n-argument functor $\varphi \in \Phi_n$ the function φ_{R_0} such that $\varphi_{R_0}(\tau_1, \ldots, \tau_n) = \varphi(\tau_1, \ldots, \tau_n)$, and to every n-argument predicate $\varrho \in P_n$ the relation ϱ_{R_0} such that $\varrho_{R_0}(\tau_1, \ldots, \tau_n) = h\big(\|\varrho(\tau_1, \ldots, \tau_n)\|\big)$.

LEMMA 1. *Let* ∇ *be the Q-filter defined above,* R_0 *the canonical realization of the language* $\mathscr{L}$ *and let* ι *denote the valuation given by*

$$\iota(a) = \begin{cases} \mathbf{1} \text{ if } \|a\| \in \nabla, \\ \mathbf{0} \text{ if } \|a\| \notin \nabla, \end{cases} \quad \text{for every } a \in V_0;$$

$$\iota(x) = x \quad \text{for every } x \in V_i.$$

Then for every formula $\alpha \in FSF$, $\alpha_{R_0}(\iota) = h(\|\alpha\|)$.

Proof: 1. Notice that Lemma 1 holds for all open formulas (see [10]). Let us assume that Lemma 1 holds for all formulas that are submitted to the formula α and let us consider the form of the formula α.

2. If α is of the form $s_1 \ldots s_n\beta$ where β is an atomic formula, then $(s_1 \ldots s_n\beta)_{R_0}(\iota) = (s_1 \ldots s_{n-1}\overline{s_n\beta})_{R_0}(\iota) = h(\|s_1 \ldots s_{n-1}\overline{s_n\beta}\|)$. As $(s_1 \ldots \overline{s_n\beta} \Leftrightarrow s_1 \ldots s_n\beta)$ is an axiom, so $(s_1 \ldots s_n\beta)_{R_0}(\iota) = h(\|s_1 \ldots s_n\beta\|)$.

3. The formula α is of the form $s(\beta \cup \gamma)$, where s is a sequence of substitutions, i.e. $s(\beta\cup\gamma)$ denotes the formula $\big(s_1 s_2 \ldots s_n(\beta\cup\gamma)\big)$. By the definition of realization and by Lemma 3.7, $\alpha_{R_0}(\iota) = (s\beta)_{R_0}(\iota) \vee (s\gamma)_{R_0}(\iota)$ and by the inductive assumption $(s\beta)_{R_0}(i) = h(\|s\beta\|)$ and $(s\gamma)_{R_0}(\iota) = h(\|s\gamma\|)$. Since h is a homomorphism, then by Theorem 2 and axiom T17 we have $\alpha_{R_0}(\iota) = h(\|s\beta\| \vee \|s\gamma\|) = h(\|s\beta \cup s\gamma\|) = h\big(\|s(\beta\cup\gamma)\|\big)$. Analogous considerations for the formulas of the form $s(\beta\cap\gamma)$, $s(\beta \Rightarrow \gamma)$, $s\neg\alpha$ are omitted.

4. Let α be of the form $s\varrho(\tau_1, \ldots, \tau_n)$ and let, for some $1 \leqslant i \leqslant n$, $\tau_i \in FST - T$; then by axiom T16 and by the inductive assumption we have

$$\begin{aligned}\big(s\varrho(\tau_1, \ldots, \tau_n)\big)_{R_0}(\iota) &= \Big(s\chi\big(\varrho(\tau_1, \ldots, \tau_n)\big)\Big)_{R_0}(\iota) \\ &= h\Big\|s\chi\big(\varrho(\tau_1, \ldots, \tau_n)\big)\Big\| = h\|s\varrho(\tau_1, \ldots, \tau_n)\|.\end{aligned}$$

5. Let α be of the form $sK\beta$ where K is not a substitution. Now we must consider the form of the program K.

A. $K = \circ[MN]$. By Lemma 3.7 we have $\alpha_{R_0}(\iota) = (sK\beta)_{R_0}(\iota) = \big(sM(N\beta)\big)_{R_0}(\iota)$. But the formula $\big(sM(N\beta)\big)$ is submitted to α, so $\alpha_{R_0}(\iota) = h(\|sMN\beta\|)$. Since the formula $\big(sMN\beta \Leftrightarrow (s\circ[MN]\beta)\big)$ is a theorem, $\alpha_{R_0}(\iota) = h\big(\|s(\circ[MN]\beta)\|\big)$.

B. $K = \veebar[\gamma MN]$. By Lemma 3.7 we have

$$\alpha_{R_0}(\iota) = \big(s(\gamma\cap M\beta)\cup s(\neg\gamma\cap N\beta)\big)_{R_0}(\iota)$$

and by the inductive assumption

$$h(\|s(\gamma \cap M\beta)\|) = (s(\gamma \cap M\beta))_{R_0}(\iota),$$
$$h(\|s(\neg\gamma \cap N\beta)\|) = (s(\neg\gamma \cap N\beta))_{R_0}(\iota).$$

Hence by Theorem 2 and axiom T26

$$a_{R_0}(\iota) = h(\|s(\gamma \cap M\beta) \cup s(\neg\gamma \cap N\beta)\|) = h(\|s \vee [\gamma MN]\beta\|).$$

C. $K = *[\gamma M]$. It is sufficient to consider the case 6, since

$$a_{R_0}(\iota) = (s*[\gamma M]\beta)_{R_0}(\iota) = (s \bigcup \vee [\gamma M[\ ]](\neg\gamma \cap \beta))_{R_0}(\iota).$$

6. Let a be of the form $s\bigcup K\beta$. By the definition of realization (see § 2)

$$a_{R_0}(\iota) = (s\bigcup K\beta)_{R_0}(\iota) = \text{l.u.b. } \{(sK^j\beta)_{R_0}(\iota)\}_{j\in\mathcal{N}}.$$

Now by the inductive assumption $a_{R_0}(\iota) = \text{l.u.b. } (h\|sK^j\beta\|)_{j\in\mathcal{N}}$ and by Theorem 3 we have

$$a_{R_0}(\iota) = h(\text{l.u.b. } (\|sK^i\beta\|)_{i\in\mathcal{N}}) = h(s\|\bigcup K\beta\|) = h(\|s\bigcup K\beta\|).$$

The analogous proof for the formula $s\bigcap K\beta$ is omitted. ■

LEMMA 2. *For every formula $a \in FSF$ and for every valuation $v \in T^{V_i} \times B_0^{V_0}$ there exists a substitution s_v such that $a_{R_0}(v) = (s_v a)_{R_0}(\iota)$.*

Proof: Let $V(a)$ denote the set of all variables that occur in the formula a, and write $V(a) = \{x_1, \ldots, x_n\} \cup \{a_1, \ldots, a_m\}$ where $x_j \in V_i$ and $a_j \in V_0$. Let v be any valuation in the set T and in the Boolean algebra B_0. Presume that

$$s_v = [x_1/v(x_1), \ldots, x_n/v(x_n), a_1/v(a_1), \ldots, a_m/v(a_m)]$$

and $s_{vR_0}(\iota) = \hat{v}$. By definition, $\hat{v}(x_i) = v(x_i)_{R_0}(\iota)$. Since $v(x_i)$ is a classical term, $v(x_i)_{R_0}(\iota) = v(x_i)$. This implies that the valuations v and $\hat{v}$ are indentical on the set $V(a)$. Thus $a_{R_0}(v) = (s_v a)_{R_0}(\iota)$. ■

DEFINITION 3. A realization R of the language $\mathcal{L}$ is a *model for the theory* $\mathcal{T} = \langle \mathcal{L}, \mathcal{C}, \mathcal{A} \rangle$ provided R is a model for the set $\mathcal{A}$ of specific axioms of $\mathcal{T}$.

DEFINITION 4. The theory $\langle \mathcal{L}, \mathcal{C}, \mathcal{A} \rangle$ is *consistent* if and only if there exists a formula a such that $a \notin \mathcal{C}(\mathcal{A})$.

THEOREM 4. *The canonical realization R_0 is a model for any consistent theory $\mathcal{T} = \langle \mathcal{L}, \mathcal{C}, \mathcal{A} \rangle$.*

Proof: Let $a \notin \mathcal{C}(\mathcal{A})$; then by Theorem 2, $\|\neg a\| \neq \mathbf{0}$. By Lemma [10], II, 9.2 there exists a Q-filter ∇ such that $\|\neg a\| \in \nabla$. Let R_0 be the canonical realization determined by ∇. If $\gamma \in \mathcal{C}(\mathcal{A})$, then, by Theorem 2, $\|\gamma\| = \mathbf{1}$. Since the element $\mathbf{1}$ belongs to every filter, $\|\gamma\| \in \nabla$. By Lemma 2, $\gamma_{R_0}(v) = (s_v\gamma)_{R_0}(\iota)$ for every valuation v. By rule r2 the formula $(s_v\gamma)$ is a theorem and therefore $\|s_v\gamma\| \in \nabla$. Since $(s_v\gamma)_{R_0}(\iota) = \mathbf{1}$, then $\gamma_{R_0}(v) = \mathbf{1}$ for every valuation v and consequently R_0 is a model for $\mathcal{T}$.

THEOREM 5. *For every formula a in a consistent theory $\mathcal{T} = \langle \mathcal{L}, \mathcal{C}, \mathcal{A} \rangle$ the following conditions are equivalent:*

(i) *a is a theorem in $\mathcal{T}$,*

(ii) *a is valid in every semantic model for $\mathcal{T}$.*

Proof: (i) implies (ii) by Lemma 2 in § 5.

Let us suppose that $a \notin \mathcal{C}(\mathcal{A})$. By Theorem 2, $\|\neg a\| \neq \mathbf{0}$. Now, by Lemma [10], II, 9.2 there exists a Q-filter ∇ such that $\|\neg a\| \in \nabla$. Let us consider the canonical realization R_0 determined by the Q-filter ∇. By Lemma 1 we have $\neg a_{R_0}(\iota) = \mathbf{1}$, where ι is the identity valuation for all individual variables and $\iota(a) = \mathbf{1}$ iff $\|a\| \in \nabla$ for all propositional variables. Thus R_0 is not a model for a, while by Theorem 4, R_0 is a model for $\mathcal{T}$. This contradicts (ii). ∎

The following theorem corresponds to the lower Skolem–Löwenheim theorem in the classical logic.

THEOREM 6. *If the theory $\mathcal{T}$ has an infinite model then it has a denumerable model.*

Proof: Let us suppose that $\mathcal{T}$ is not consistent. Then there exists a formula a such that $\neg a$ and a are theorems in $\mathcal{T}$. Let R be a model for $\mathcal{T}$. Then $a_R(v) = \mathbf{1}$ and $(\neg a)_R(v) = \mathbf{1}$ for every valuation v. Hence, if the theory $\mathcal{T}$ has a model then it is consistent. Now, by Lemma 1, the canonical realization is a model for $\mathcal{T}$ and it is denumerable. ∎

Let us see that the second theorem of Skolem–Löwenheim is not true in the class of all ordinary semantic models, i.e. the realization of the equality sign is the characteristic function of the relation of identity. One can prove that the theory with the following formulas as axioms

$$\neg(Sx = 0), \quad [x/0]\bigcup[x/Sx](x = y), \quad \big((Sx = Sy) \Rightarrow (x = y)\big)$$

is categorical, i.e. all its models are isomorphic with the standard model for arithmetic.

At the end of this section we quote some algorithmic tautologies:

T39 $\big(\bigcap Ka \Rightarrow K^i a\big)$ for every natural number i,

T40 $\big(K^i a \Rightarrow \bigcup Ka\big)$ for every natural number i,

T41 $\big(\bigcap Ka \Rightarrow \bigcup Ka\big)$,

T42 $\big(\neg\bigcup K\neg a \Rightarrow \bigcup Ka\big)$,

T43 $\big(\bigcap Ka \Rightarrow \neg\bigcap K\neg a\big)$,

T44 $\big(\bigcap Ka \Rightarrow \bigcap K(K\mathbf{1})\big)$,

T45 $\big(\bigcup K\neg a \Rightarrow \neg\bigcap Ka\big)$,

T46 $\big(\bigcap K\neg a \Rightarrow \neg\bigcup Ka\big)$,

T47 $\big(\bigcap K\mathbf{1} \Rightarrow (\neg\bigcap Ka \Rightarrow \bigcup K\neg a)\big)$,

T48 $\big(\bigcap K\mathbf{1} \Rightarrow (\neg\bigcup K\mathbf{1} \Rightarrow \bigcap K\neg a)\big)$.

In the formulas T49–T56 we assume that $V(K) \cap V(\beta) = \emptyset$.

T49 $(K\beta \Rightarrow \beta)$,
T50 $(K\mathbf{1} \Rightarrow (\beta \Rightarrow K\beta))$,
T51 $\left(\bigcap K(\alpha \Rightarrow \beta) \Rightarrow \left(\bigcup K\alpha \Rightarrow \beta\right)\right)$,
T52 $\left(\bigcap K\mathbf{1} \Rightarrow \left(\left(\bigcup K\alpha \Rightarrow \beta\right) \Rightarrow \bigcap K(\alpha \Rightarrow \beta)\right)\right)$,
T53 $\left(\bigcup K(\beta \Rightarrow \alpha) \Rightarrow \left(\beta \Rightarrow \bigcup K\alpha\right)\right)$,
T54 $\left(\bigcap K\mathbf{1} \Rightarrow \left(\left(\beta \Rightarrow \bigcup K\alpha\right) \Rightarrow \bigcup K(\beta \Rightarrow \alpha)\right)\right)$,
T55 $\left(\left(\bigcap K\alpha \cap \beta\right) \Leftrightarrow \bigcap K(\alpha \cap \beta)\right)$,
T56 $\left(\left(\bigcup K\alpha \cup \beta\right) \Leftrightarrow \bigcup K(\alpha \cup \beta)\right)$,

$$\left.\begin{array}{ll}\text{T57} & \left(\bigcup K \bigcap M\alpha \Rightarrow \bigcap M \bigcup K\alpha\right)\\ \text{T58} & \left(\bigcap K \bigcap M\alpha \Rightarrow \bigcap M \bigcap K\alpha\right)\\ \text{T59} & \left(\bigcup K \bigcup M\alpha \Rightarrow \bigcup M \bigcup K\alpha\right)\end{array}\right\} \quad \text{if } V(M) \cap V(K) = \emptyset.$$

8. Herbrand theorem

In this section we present a theorem analogous to the Herbrand theorem in classical logic. This theorem refers only to a narrow class of formulas but it is useful in solving certain decidability problems.

Let n be a natural number, let K_i, M_i, $i = 0, 1, \ldots, n$, be programs in which the sign $*$ does not appear and let α belong to the set of open formulas. Under these assumptions we can formulate the following theorem:

THEOREM 1. *Any formula β of the form $M_0 \bigcup K_0 \ldots M_n \bigcup K_n \alpha$ is a tautology of algorithmic logic if and only if there exists a natural number m such that the formula $\left(M_0 \sum_{i=0}^{m} K_0^i \ldots M_n \sum_{j=0}^{m} K_n^j \alpha\right)$ is a tautology of algorithmic logic.*

Proof: Let us assume that there exists a natural number m such that the formula $\left(M_0 \sum_{i=0}^{m} K_0^i \ldots M_n \sum_{j=0}^{m} K_n^j \alpha\right)$ is a tautology in algorithmic logic. Let R denote any realization of the algorithmic language $\mathscr{L}$ and v any valuation. Then

$$\left(M_0 \bigcup K_0 \ldots M_n \bigcup K_n \alpha\right)_R(v) = \underset{i \in \mathcal{N}}{\sup}\left(\ldots \underset{j \in \mathcal{N}}{\sup}\left(M_0 K_0^i \ldots M_n K_n^j \alpha\right)_R(v) \ldots\right)$$
$$\geqslant \text{l.u.b.}\left\{\left(\ldots \text{l.u.b.}\left\{\left(M_0 K_0^i \ldots M_n K_n^j \alpha\right)_R(v)\right\}_{j \leqslant m} \ldots\right\}_{i \leqslant m} = \mathbf{1}.$$

Hence the formula $M_0 \bigcup K_0 \ldots M_n \bigcup K_n \alpha$ is a tautology.

Conversely, let us denote by H_m the formula $\left(M_0 \sum_{i=0}^{m} K_0^i \ldots M_n \sum_{j=0}^{m} K_n^j \alpha\right)$ and suppose that no formula H_m for $m \in \mathcal{N}$ is a tautology. We shall construct a semantic realization R_0 in the set of all classical terms T and a valuation v_0 such that $(M_0 \bigcup K_0 \ldots M_n \bigcup K_n \alpha)_{R_0}(v_0) = \mathbf{0}$. Let us denote by H'_m the classical open formula obtained from H_m by replacing all formulas of the form $K_0^{i_1} \ldots K_n^{i_n} \alpha$ by the equivalent open formulas (see Lemma 3.6). Now, to every formula $\varrho(\tau_1, \ldots, \tau_n)$ that occur in any H'_m we assign a new propositional variable $a_{\varrho(\tau_1,\ldots,\tau_n)}$ such that $a_{\varrho(\tau_1,\ldots,\tau_n)} \notin V(H'_m)$ for all $m \in \mathcal{N}$. In that way we obtain a set of propositional formulas $\{H''_m\}_{m \in \mathcal{N}}$. For every $m \in \mathcal{N}$, let us denote by W^m the set of all valuations that do not satisfy the formula H''_m restricted to the set $V(H''_m)$. Notice that if, for some m, W^m were empty, then H''_m would be a tautology and consequently H_m would be so, too. Hence W^m is a finite nonempty set and if v is in $W^{m'}$ and $m' > m$, then there exists a valuation $\hat{v}$ in the set W^m such that $\hat{v}(z) = v(z)$ for all $z \in V(H''_m)$.

Let us number the elements of the sets W^m, $m \in \mathcal{N}$, by sequences of the form $(i_0, \ldots, i_m)$ where $i_0, \ldots, i_m$ are natural numbers, in the following way:

(1) elements of W^0 are numbered by natural numbers, no matter how;

(2) if $v_1, \ldots, v_k$ are in W^{m+1} and $v_1, \ldots, v_k$ are all valuations that are identical with $v \in W^m$ on the set $V(H''_m)$ and if $(i_0, \ldots, i_m)$ is the number of v, then v_i has the number $(i_0, \ldots, i_m, i)$ for $i \leqslant k$.

The set $\mathcal{D}$ of all such sequences with the empty sequence adjoined is a tree (see Definition 5.3). By König's theorem, see [4], there exists an infinite branch in $\mathcal{D}$. Denote by v_∞ the valuation such that, for every i, v_∞ restricted to the set $V(H''_i)$ is identical with the element of the set W^i whose number belongs to that infinite branch. For every natural number i we have

$$H''_i(v_\infty) = \mathbf{0}. \tag{1}$$

Let R_0 be the realization in the set of all terms T and in the Boolean algebra B_0 such that $\varrho_{R_0}(\tau_1, \ldots, \tau_n) = \mathbf{1}$ if and only if $a_{\varrho(\tau_1,\ldots,\tau_n)}(v_\infty) = \mathbf{1}$ and let v_0 be the valuation such that $v_0(x) = x$ for $x \in V_i$ and $v_0(a) = a(v_\infty)$ for $a \in V_0$. By (1) we have $H_{iR_0}(v_0) = \mathbf{0}$ for $i \in \mathcal{N}$. Let $m_0, \ldots, m_n$ be any natural numbers and let $m_i = \max(m_0, \ldots, m_n)$. Then

$$H_{m_i R_0}(v_0)$$
$$= (M_0 K_0^{m_0} \ldots M_n K_n^{m_n} \alpha)_{R_0}(v_0) \vee \Big(M_0 \sum_{i \neq m_0}^{m_i} K_0 \ldots M_n \sum_{i \neq m_n}^{m_i} K_n \alpha\Big)_{R_0}(v_0) = \mathbf{0}.$$

So $(M_0 K_0^{m_0} \ldots M_n K_n^{m_n} \alpha)_{R_0}(v_0) = \mathbf{0}$ and consequently $\beta_{R_0}(v_0) = \mathbf{0}$. □

Let us denote by $\mathcal{K}$ the class of all formulas of the form $M_0 \bigcup K_0 \ldots \ldots M_n \bigcup K_n$ where K_i, M_i, α $(i \leqslant n)$ are as above. From the theorem just proved it follows immediately that the set of all tautologies in $\mathcal{K}$ is recursively enumerable. Indeed, let β be a formula of the class $\mathcal{K}$. By Theorem 8.1, β is a tautology if and only if there exists a formula in which signs of quantifiers do not appear and which is a tautology of algorithmic logic. So, by Lemma 3.6 we can find out an open formula β_0 such that $\models \beta$ if and only if $\models \beta_0$. Now, in a finite number of steps we can check whether β_0 is a substitution in a tautology of propositional calculus.

As a simple generalization of Theorem 8.1 we have

THEOREM 2. *Let α be any formula of the form*

$$(\bigcap K_0' \ldots \bigcap K_m' M_0 \bigcup K_0 \ldots M_n \bigcup K_n \beta),$$

where $K_0', \ldots, K_m' M_0, K_0, \ldots, M_n$ are programs without $*$, and such that in β the symbols $\cap$, $\cup$, $*$ do not occur. Then α is a tautology of algorithmic logic if and only if there exist natural numbers $m_0, \ldots, m_n$ such that the formula $(\sum_{i=0}^{m_0} K_0^i \ldots \sum_{j=0}^{m_n} K_n^j \beta)$ is a tautology. ∎

9. The normal form of a program

In the sequel we assume that in the algorithmic language $\mathscr{L}$ there exists a two-argument predicate $-$ and we restrict our considerations to ordinary semantic realizations. We shall consider a theory $\mathscr{T}' = \langle \mathscr{L}', \mathscr{C}, \mathscr{A}' \rangle$ with a consequence operation defined as in § 5 and we assume that the set of specific axioms $\mathscr{A}'$ includes the axioms of identity, i.e. set $\mathscr{E}$ of formulas of the form

e1. $x = x$,
e2. $(x = y \Rightarrow y = x)$,
e3. $(x = y \Rightarrow (y = z \Rightarrow x = z))$,
e4. $((x_1 = y_1 \cap (\ldots \cap x_n = y_n) \ldots) \Rightarrow \varphi(x_1, \ldots, x_n) = \varphi(y_1, \ldots, y_n))$, $n \in \mathscr{N}$,
e5. $((x_1 = y_1 \cap (\ldots \cap x_n = y_n) \ldots) \Rightarrow (\varrho(x_1, \ldots, x_n) \Leftrightarrow \varrho(y_1, \ldots, y_n)))$,

where $x, y, z, x_1, \ldots, x_n, y_1, \ldots, y_n$ are individual variables, φ is an n-argument functor, and ϱ is an n-argument predicate.

The following theorem can be proved:

The theory $\mathscr{T}'$ with equality is consistent if and only if the theory $\mathscr{T} = \langle \mathscr{L}, \mathscr{C}, \mathscr{A}' - \mathscr{E} \rangle$ is consistent.

DEFINITION 1. Two programs K, M are *equivalent in the sense of the set of variables Z*, in symbols $K \underset{Z}{\sim} M$, if and only if for every realization R and every valuation v the following conditions hold:

(1) $K_R(v)$ is defined if and only if $M_R(v)$ is defined;

(2) if both valuations $K_R(v)$ and $M_R(v)$ are defined then for any variable $z \in Z$, $z_R(K_R(v)) = z_R(M_R(v))$.

In the case where Z is the set of all variables in the language $\mathscr{L}$ we shall write $K \sim M$ instead of $K \underset{V_0 \cup V_i}{\sim} M$.

LEMMA 1. *For every programs K, M the following conditions are equivalent:*

(i) $K \sim M$;

(ii) *for every term* $\tau \in FST$, *for every realization* R *and every valuation* v, $(K\tau)_R(v) = (M\tau)_R(v)$;

(iii) *for every formula* $\alpha \in FSF$, *every realization* R *and every valuation* v, $(K\alpha)_R(v) = (M\alpha)_R(v)$;

(iv) *for every formula* $\alpha \in FSF$ *the formula* $(K\alpha \Leftrightarrow M\alpha)$ *is a theorem in the theory* $\langle \mathcal{L}', \mathcal{C}, \mathcal{E} \rangle$.

LEMMA 2. *For every programs K, M we can find in a constructive way a formula $\alpha' \in FSF$ such that α' is a tautology of algorithmic logic if and only if $K \sim M$.*

Let K, M be in FS and let us put

$$\alpha' = \Big((\neg K\mathbf{1} \cap \neg M\mathbf{1}) \cup (K\mathbf{1} \cap M\mathbf{1}) \cap \Big(\prod_{i=1}^{n} (Kx_i = Mx_i) \cap \prod_{i=1}^{m} (Ka_i \Leftrightarrow Ma_i)\Big)\Big),$$

where $x_1, \ldots, x_n$ are all individual variables in $\circ[KM]$ and $a_1, \ldots, a_m$ are all propositional variables in $\circ[KM]$. If α' is a tautology of algorithmic logic, then for every valuation v in a fixed realization R we have

$$(\neg K\mathbf{1} \cap \neg M\mathbf{1})_R(v) = \mathbf{1}$$

or

$$\Big(\prod_{i=1}^{n} (Kx_i = Mx_i) \cap \prod_{i=1}^{m} (Ka_i \Leftrightarrow Ma_i)\Big)_R(v) = \mathbf{1}.$$

So α' is a tautology if and only if $K_R(v)$ and $M_R(v)$ are undefined or $K_R(v)$ and $M_R(v)$ are defined and for every variable $z \in V(\circ[KM])$, $z_R\big(K_R(v)\big) = z_R\big(M_R(v)\big)$, i.e., $K \sim M$. ■

LEMMA 3. *For every programs K, M, N, L and for every open formulas α, β, γ the following equivalences hold:*

1. $\circ\big[K \circ [MN]\big] \sim \circ\big[\circ[KM]N\big]$,

2. $\circ\big[*[\alpha K]N\big] \underset{-\{c\}}{\sim} \circ\Big[[c/\mathbf{1}] * \Big[c \veebar \big[\alpha K \circ [N[c/\mathbf{0}]]\big]\Big]\Big]$,

3. $\veebar\big[\alpha * [\beta K] * [\gamma M]\big] \underset{-\{c\}}{\sim} \circ\Big[[c/\alpha] * \big[\big((c \cap \beta) \cup (\neg c \cap \gamma)\big) \veebar [cKM]\big]\Big]$,

4. $\circ\big[*[\alpha K] * [\beta M]\big] \underset{-\{c\}}{\sim} \circ\Big[[c/\mathbf{1}] * \Big[(c \cup \beta) \veebar \big[(\alpha \cap c) K \circ [[c/\mathbf{0}]\, M]\big]\Big]\Big]$,

5. $*\big[\alpha \circ [K * [\beta M]]\big] \underset{-\{c\}}{\sim}$
$$\circ\Big[[c/\mathbf{1}] * \Big[\big((c \cap \alpha) \cup (\neg c \cap \beta)\big) \circ \big[\veebar [cKM] \veebar [\beta [c/\mathbf{0}][c/\mathbf{1}]]\big]\Big]\Big],$$

6. $\veebar\big[\alpha \circ [KM] \circ [NL]\big] \underset{-\{c\}}{\sim} \circ\big[[c/\alpha] \circ [\veebar [cKN] \veebar [cML]]\big]$,

7. $\circ\big[K * [\beta N]\big] \underset{-\{c\}}{\sim} \circ\Big[[c/\mathbf{1}] * \big[(c \cup \beta) \veebar [c \circ [K[c/\mathbf{0}]]\, N]\big]\Big]$,

where c is a propositional variable that occurs in no one of the left-hand sides of equivalences 3–7.

Proof: Let us prove, for example, equivalence 5. Write

$$N = \circ\left[\veebar[cKM]\veebar\left[\beta[c/\mathbf{0}][c/\mathbf{1}]\right]\right] \quad \text{and} \quad \gamma = \left((c\cap a)\cup(\neg c\cap\beta)\right).$$

Further, let R be any realization and v any valuation. $\circ\left[[c/\mathbf{1}]*[\gamma N]\right]_R(v) = v_1$ if and only if there exists a natural number i such that $\gamma_R\left(N_R^i\left([c/\mathbf{1}]_R(v)\right)\right) = \mathbf{0}$ and $N_R^i\left([c/\mathbf{1}]_R(v)\right)$ is defined, and for $j = 1, \ldots, i-1$, we have $(N^j\gamma)_R\left([c/\mathbf{1}]_R(v)\right) = \mathbf{1}$. But

$$N_R^i\left([c/\mathbf{1}]_R(v)\right) = \circ\left[\ldots\circ\left[\circ[KM^{i_1}]\circ[KM^{i_2}]\right]\ldots\circ[KM^{i_n}]\right]_R(v),$$

where $i_1+i_2+\ldots+i_n+n = i$, and for every $k < n$ we have

$$a_R\left(\circ[KM^{i_k}]_R(\hat{v})\right) = \mathbf{1}, \quad \beta_R\left(\circ[KM^{i_k}]_R(\hat{v})\right) = \mathbf{0}, \quad \beta_R\left(\circ[KM^j]_R(\hat{v})\right) = \mathbf{1}$$

for $j < i_k$ and for $\hat{v} = \circ\left[\circ[KM^{i_1}]\ldots\circ[KM^{i_{k-1}}]\right]$. So $N_R^i\left([c/\mathbf{1}]_R(v)\right) = \circ\left[\circ\left[K*[\beta M]\right]\right]_R^n(v)$. By assumption, $\gamma_R(v') = \mathbf{1}$, $\gamma_R\left(N_R(v')\right) = \mathbf{0}$, where $v' = N_R^{i-1}\left([c/\mathbf{1}]_R(v)\right)$. Thus either $\left(((c\cap a)\cap\neg Nc)\cap\neg N\beta\right)_R(v') = \mathbf{1}$ or $\left(((c\cap a)\cap Nc)\cap\neg Na\right)_R(v') = \mathbf{1}$, or $\left(((\neg c\cap\beta)\cap\neg Nc)\cap\neg N\beta\right)_R(v') = \mathbf{1}$, or $\left(((\neg c\cap\beta)\cap Nc)\cap\neg Na\right)_R(v') = \mathbf{1}$. The cases one and three are impossible for

$$N_R(v') = \veebar\left[\beta[c/\mathbf{0}][c/\mathbf{1}]\right]_R\left(\veebar[cKM]_R(v')\right)$$
$$= \begin{cases} \veebar\left[\beta[c/\mathbf{0}][c/\mathbf{1}]\right]_R\left(K_R(v')\right) & \text{if} \quad c_R(v') = \mathbf{1} \\ \veebar\left[\beta[c/\mathbf{0}][c/\mathbf{1}]\right]_R\left(M_R(v')\right) & \text{if} \quad c_R(v') = \mathbf{0} \end{cases}$$
$$= \begin{cases} [c/\mathbf{0}]_R\left(K_R(v')\right) & \text{if} \quad c_R(v') = \mathbf{1} \text{ and } \beta_R\left(K_R(v')\right) = \mathbf{1}, \\ [c/\mathbf{1}]_R\left(K_R(v')\right) & \text{if} \quad c_R(v') = \mathbf{1} \text{ and } \beta_R\left(K_R(v')\right) = \mathbf{0}, \\ [c/\mathbf{0}]_R\left(M_R(v')\right) & \text{if} \quad c_R(v') = \mathbf{0} \text{ and } \beta_R\left(M_R(v')\right) = \mathbf{1}, \\ [c/\mathbf{1}]_R\left(M_R(v')\right) & \text{if} \quad c_R(v') = \mathbf{0} \text{ and } \beta_R\left(M_R(v')\right) = \mathbf{0}, \end{cases}$$

and in both these cases we have simultaneously $c_R\left(N_R(v')\right) = \mathbf{0}$ and $c_R\left(N_R(v')\right) = \mathbf{1}$. As a consequence we get $a_R\left(N_R(v')\right) = \mathbf{0}$ and $\hat{v}_1 = *\left[a\circ\left[K*[\beta M]\right]\right]_R(v)$. ■

Lemma 4. *If $K \underset{Z}{\sim} M$ then for any program L and open formula a we have*

(i) $\circ[KL] \underset{Z}{\sim} \circ[ML]$, $\qquad \circ[LK] \underset{Z}{\sim} \circ[LM]$,

(ii) $\veebar[aKL] \underset{Z}{\sim} \veebar[aML]$, $\qquad \veebar[aLK] \underset{Z}{\sim} \veebar[aLM]$,

(iii) $*[aK] \underset{Z}{\sim} *[aM]$ *if* $V(a) - Z = \emptyset$, $Z \subset V_0 \cup V_i$.

The proof follows immediately from Lemmas 9.1 and 3.7. ■

DEFINITION 2. A program $K \in FS$ is *in the normal form* if and only if it is in the form $\circ[M*[\alpha N]]$, where the sign $*$ does not appear in the programs M, N.

Remark 1: For every programs K, M, N and every sets Z_1, Z_2, if $K \underset{Z_1}{\sim} M$ and $M \underset{Z_2}{\sim} N$, then $K \underset{Z_1 \cap Z_2}{\sim} N$.

THEOREM 1. *To any program K we can effectively assign a program K' in the normal form (called the normal form of K) such that $V(K') = V(K) \cup Z$, where $\emptyset \neq Z \subset V_0$ and $K \underset{V \cup V_0 - Z}{\sim} K'$.*

Proof: To simplify our considerations let us denote by C (with indices if necessary) any program without $*$ and by I (or $I_1, I_2, \ldots$) any program of the form $*[\alpha C]$. The proof of the theorem is by induction on the length of program.

1. Notice that $\circ[C*[\mathbf{0}[\ ]]] \sim C$ and $\circ[[\ ]I] \sim I$, so Theorem 1 holds for every programs C and I. Let us assume that the theorem holds for any programs K, M and let $K \underset{Z_1}{\sim} \circ[C_1 I_1]$ and $M \underset{Z_2}{\sim} \circ[C_2 I_2]$ for some $Z_1, Z_2 \subset V_0 \cup V_i$.

2. Let us consider a program of the form $\circ[KM]$ and let $V(\circ[KM]) - Z_1 \cap Z_2 = \emptyset$. By Lemma 4, $\circ[KM] \underset{Z_1 \cap Z_2}{\sim} \circ[\circ[C_1 I_1] \circ [C_2 I_2]]$ and by equivalence 1 of Lemma 3, $\circ[KM] \underset{Z_1 \cap Z_2}{\sim} \circ[C_1 \circ [I_1 \circ [C_2 I_2]]] \underset{Z_1 \cap Z_2}{\sim} \circ[C_1 \circ [\circ [I_1 C_2] I_2]]$. Now, by equivalence 2 of the same lemma, we can find a program I_3 such that $\circ[I_1 C_2] \underset{-\{c\}}{\sim} \circ[[c/\mathbf{1}] I_3]$. So we have $\circ[KM] \underset{Z_1 \cap Z_2 - \{c\}}{\sim} \circ[\circ[C_1[c/\mathbf{1}]] \circ [I_3 I_2]]$. Let $c \notin V(\circ[I_2 I_3])$. Then there exists a program I_4 such that $\circ[I_3 I_2] \underset{-\{c,c'\}}{\sim} \circ[[c'/\mathbf{1}] I_4]$. Consequently $\circ[KM] \underset{Z_1 \cap Z_2 - \{c,c'\}}{\sim} \circ[[c'/\mathbf{1}] I_4]$.

3. Let us consider a program of the form $\veebar[\alpha KM]$ where $\alpha \in F$ and $V(\veebar[\alpha KM]) - Z_1 \cap Z_2 = \emptyset$. By Lemma 4, $\veebar[\alpha KM] \underset{Z_1 \cap Z_2}{\sim} \veebar[\alpha[C_1 I_1] \circ [C_2 I_2]]$. Suppose that $c \notin \veebar[\alpha \circ [C_1 I_1] \circ [C_2 I_2]]$. Then by equivalence 6 of Lemma 3 there is a program C_3 such that $\veebar[\alpha KM] \underset{Z_1 \cap Z_2 - \{c\}}{\sim} [C_3 \veebar [c I_1 I_2]]$. What more, $\veebar[c I_1 I_2] \underset{-\{c'\}}{\sim} \circ[[c'/\alpha] I_3]$ and therefore $\veebar[\alpha KM] \underset{Z_1 \cap Z_2 - \{c,c'\}}{\sim} \circ[C_3 \circ [[c'/\alpha] I_3]]$. So, by equivalence 1 of Lemma 3, $\veebar[\alpha KM] \underset{Z_1 \cap Z_2 - \{c,c'\}}{\sim} \circ[\circ[C_3[c'/\alpha]] I_3]$.

4. Let $*[\alpha K]$ be a program such that $V(*[\alpha K]) - Z_1 = \emptyset$. By Lemma 4 and the inductive assumption we have $*[\alpha K] \underset{Z_1}{\sim} *[\alpha \circ [C_1 I_1]]$. Let $c \notin V(*[\alpha \circ [C_1 I_1]])$. Then by equivalence 5 there exists such a program I_3 that $*[\alpha K] \underset{Z_1 - \{c\}}{\sim} \circ[[c/\mathbf{1}] I_3]$.

Thus Theorem 9.1 holds for every program. ∎

Remark 2: In Theorem 9.1, Z is the set of auxiliary variables. The value of the program K does not depend on the values of these variables.

Remark 3: By equivalence 7 of Lemma 3 we can simplify the normal form of a program in such a way that M is identical with a substitution of the form $[c/a]$ where $c \in V_0$.

10. Applications

The aim of this section is to indicate some simple applications of the above results in the theory of programs.

LEMMA 1. *Let K be a program in a normal form, $K = \circ[M*[aN]]$, and let $\models K\mathbf{1}$. Then there exists a natural number n_0 such that for every realization R and every valuation v if $K_R(v) = \circ[MN^i_{;}]_R(v)$, then $i < n_0$.*

Proof: By assumption, the formula $K\mathbf{1}$ is a tautology and so $M \bigcup \veebar [aN[\ ]](\neg a)$ is a tautology of algorithmic logic. By Theorem 8.1 there exists a natural number n_0 such that the formula $M \sum_{i=0}^{n_0} \veebar [aN[\ ]](\neg a)$ is a tautology. Since for every natural number i there exists $j \leqslant i$ such that $\veebar [aN[\ ]]^i = N^j$, it follows that

$$\models M \sum_{i=0}^{n_0} N^i \neg a. \tag{1}$$

Now suppose that for some realization R and some valuation v, $K_R(v) = \circ[MN^i]_R(v)$ and $i > n_0$. By the definition of a realization we have $a_R(\circ[MN^i]_R(v)) = \mathbf{0}$ and for every $j < i$, $a_R(\circ[MN^j]_R(v)) = \mathbf{1}$. Consequently, for every $j \leqslant n_0$,

$$a_R(\circ[MN^j]_R(v)) = \mathbf{1}.$$

This leads to contradiction with (1). ■

Lemma 1 can be formulated in a more universal form as follows.

Let P be the set of all programs that in every realization and every valuation have finite calculation.

LEMMA 2. *For every program $K \in P$ there exists a natural number n_0 such that the length of every calculation of this program is less than n_0.*

Let us consider the halting problem formulated in the following way: given a program K, is K an element of the set P or not?

LEMMA 3. *The set P is recursively enumerable.*

Proof: Let K be any program, $K \in FS$. At first let us construct the normal form K' of the program K. By Theorem 9.1, for every realization and every valuation v, $K_R(v)$ is defined if and only if $K'_R(v)$ is defined. Since $K'_R(v)$ is defined if and only if $K\mathbf{1}$ is a tautology of algorithmic logic, by Lemma 3.7 and Theorem 8.1, if $K \in P$, we will know this after finitely many steps. ■

LEMMA 4. *The relation $\sim$ of equivalence of programs relativized to the class P is recursively enumerable.*

Proof: Let K, M are any programs. By Lemma 9.2 we can construct a formula α such that $K \sim M$ if and only if $\models \alpha$. However, K and M are in P, so $((\neg K\mathbf{1} \cap \neg M\mathbf{1}) \Leftrightarrow \mathbf{0})$ and, moreover, we can find out a formula α' in the class K (see § 8) such that $\alpha'_R(v) = \alpha_R(v)$ for every realization R and every valuation v. Indeed,

$$\Big(\Big(\sum_{i=0}^{n}(Kx_i' = Mx_i) \cap \sum_{i=0}^{m}(Ka_i \Leftrightarrow Ma_i)\Big) \Leftrightarrow s^{-1}Ks\overline{M}\big((x_i = x_i') \cap (a_i \Leftrightarrow a_i')\big)\Big).$$

Now by Theorems 9.1 and 3.7 we have $\alpha' \in K$. So, by Theorem 8.1, if $K \sim M$, we can check this in finitely many steps. ■

References

[1] Banachowski, L., *Data structures*, Reports of IMMUW 46.

[2] — *Investigations of properties of programs by means of the extended algorithmic logic*, doctoral dissertation, Warsaw University, 1975.

[3] Kreczmar, A., *The set of all tautologies of algorithmic logic is hyperarithmetical*, Bull. Acad. Polon Sci., Sér. Math. 21 (1971), 781–783.

[4] Kuratowski, K., and Mostowski, A., *Set theory*, PWN, Warsaw 1966.

[5] Mirkowska, G., *On formalized systems of algorithmic logic*, Bull. Acad. Polon. Sci., Sér. Math. 19 (1971), 421–428.

[6] — *Algorithmic logic and its applications in the theory of programs*, doctoral dissertation, Warsaw University 1972 (in Polish).

[7] — *Herbrand theorem in the algorithmic logic*, Bull. Acad. Polon. Sci., Ser. Math. 22 (1974), 539–543.

[8] Rasiowa, H., *On logical structure of programs*, Bull. Acad. Pol. Sci., Sér. Math. 20 (1972), 319–324.

[9] — *Extended ω^+-valued algorithmic logic*, ibid. 22 (1974), 605–610.

[10] Rasiowa, H. and Sikorski, R., *Mathematics of metamathematics*, PWN, Warsaw 1963.

[11] Salwicki, A., *Formalized algorithmic languages*, Bull. Acad. Polon. Sci., Sér. Math. 18 (1970), 227–232.

[12] — *Programmability and recursiveness*, to appear.