ANNALES SOCIETATIS MATHEMATICAE POLONAE
Series IV: FUNDAMENTA INFORMATICAE I (1977), 167–193

Investigations of properties of programs by means of the extended algorithmic logic II*

LECH BANACHOWSKI

Institute of Informatics, University of Warsaw

Received March 10,1976

AMS Categories: 68A05

I. THE STRONGEST CONSEQUENT AND PROPERTIES OF PROGRAMS

1. The strongest consequent

Let K be a program, let α be a formula in F and R a realization in a non-empty set J. Let us consider the following property of results v:

(∗) *there exists an initial data v' such that $\alpha_R(v') = V$, $K_R(v')$ is defined, and $K_R(v') = v$.*

Thus a result v has property (∗) iff v can be obtained by the execution of K from an initial data v' satisfying α. Like the first order predicate connectives, this construction has also an interpretation in the field of all subsets of the set of all valuations.

Namely, v has property (∗) iff v belongs to the set $\vec{K}_R(\{v: \alpha_R(v) = V\})$. Observe that the construction $K\alpha$ has the following interpretation: $\vec{K}_R^{-1}(\{v: \alpha_R(v) = V\})$.

Property (∗) is expressible in the extended algorithmic logic. Namely, let $\vec{x} = (x_1, \ldots, x_p, x_{p+1}, \ldots, x_n)$ be the sequence of all different variables occurring in the formula $K\alpha$ and such that $x_1, \ldots, x_p$ are individual variables and $x_{p+1}, \ldots, x_n$ are propositional variables.

Let $\vec{y} = (y_1, \ldots, y_p, \ldots, y_n)$ be a copy of $\vec{x}$ constructed in the following way: for each $i = 1, \ldots, p$, y_i is the first variable in the sequence V_i, not equal to any variable in the sequence $(x_1, \ldots, x_p, y_1, \ldots, y_{i-1})$ and for each $i = p+1, \ldots, n$, y_i is the first variable in the sequence

* This paper is a continuation of [6].

V_0, not equal to any variable in the sequence $(x_{p+1}, \ldots, x_n, y_{p+1}, \ldots, y_{i-1})$.

By aK we shall denote the formula

$$\exists \vec{y}\big(a(\vec{y}) \wedge K(\vec{y})\,(\vec{x} = \vec{y})\big).$$

The formula aK will be called the *strongest consequent* of the formula a with respect to the program K.

1.1. *A valuation* v *has property* $(*)$ *iff* $(aK)_R(v) = V$.

Proof: Let us observe that $(aK)_R(v) = V$ is equivalent to the following fact:

(1) *there exist elements* $j_1, \ldots, j_p$ in J and $j_{p+1}, \ldots, j_n$ in B_0 *such that the valuation*

$$v'(z) = \begin{cases} j_q & \text{if } z = y_q \text{ for some } q = 1, \ldots, n, \\ v(z) & \text{otherwise} \end{cases}$$

satisfies the following conditions:

(11) $a(\vec{y})_R(v') = V$,

(12) $K(\vec{y})_R(v')$ *is defined*,

(13) *for each* $i = 1, \ldots, n,\ v(x_i) = K(y)_R(v')(x_i) = K(\vec{y})_R(v')(y_i)$.

Now suppose that assertion (1) holds. Consider the valuation v'' defined as follows:

$$v''(z) = \begin{cases} j_q & \text{if } z = x_q \text{ for some } q = 1, \ldots, n; \\ v(z) & \text{otherwise.} \end{cases}$$

From (1) it follows that:

(21) $a_R(v'') = V$,

(22) $K_R(v'')$ is defined,

(23) for each $i = 1, \ldots, n,\ v(x_i) = K_R(v'')(x_i)$.

Thus, condition $(*)$ holds.

Conversely, if $(*)$ holds then there exists a valuation v'' satisfying conditions (21), (22) and (23). Now let v' be the following valuation:

$$v'(z) = \begin{cases} v''(x_i) & \text{if } z = y_i \text{ for some } q = 1, \ldots, n; \\ v(z) & \text{otherwise.} \end{cases}$$

From conditions (21), (22), and (23) it follows that conditions (11), (12), and (13) also hold for this v'. This means that (1) is satisfied. ∎

For example, let us consider the following formulas:

(A) $$[x/1](x \geqslant 1[x/1]),$$

(B) $$([x/1]x \geqslant 1)[x/1],$$

and the realization in the field of real numbers.

(A) is equivalent to $\mathbf{1}$ and (B) is equivalent to $x = 1$. Therefore the realization of formulas (A) and (B) depends essentially on the distribution of the parentheses.

The strongest consequent plays an important role in further considerations. It cannot be defined in the algorithmic logic without classical quantifiers. This follows from [6], 7.2 and the following simple lemma.

1.2. *For every formula α and every distinct variables x, y*

$$\exists x\alpha \equiv [x/y](\alpha[x/y]).$$

Proof: Let R be a realization and v a valuation. Applying 1.1 we obtain: $[x/y](\alpha[x/y])_R(v) = V$ iff there exists a valuation v' such that $\alpha_R(v') = V$ and $v^x_{v(y)} = [x/y]_R(v') = v'^x_{v'(y)}$. The last statement is equivalent to the following: there exists a valuation v' such that $\alpha_R(v') = V$ and $v' = v^x_j$ for some j in the universe of R. Finally, the above statement is equivalent to that $(\exists x\alpha)_R(v) = V$. ∎

1.3. *The construction of the strongest consequent has the following properties*:

(n1) $(\alpha \vee \beta)K \equiv \alpha K \vee \beta K$,

(n2) $(\alpha \wedge \beta)K \leqslant \alpha K \wedge \beta K$,

(n3) $\alpha \circ [KM] \equiv (\alpha K)M$,

(n4) $\alpha \veebar [\gamma KM] \equiv (\alpha \wedge \gamma)K \vee (\alpha \wedge \neg\gamma)M$,

(n5) *if* $\alpha \leqslant_{\mathscr{R}} \beta$ *then* $\alpha K \leqslant_{\mathscr{R}} \beta K$,

(n6) $K\mathbf{1} \wedge \alpha \leqslant_{\mathscr{R}} K\beta$ *iff* $\alpha K \leqslant_{\mathscr{R}} \beta$,

where α and β are formulas, γ is an open formula, K, M are programs and $\mathscr{R}$ is a class of realizations.

Proof: We shall use 1.1 and the definition of a realization. Let R be a realization and v a valuation.

(n1) $((\alpha \vee \beta)K)_R(v) = V$ iff there exists a valuation v' such that $(\alpha \vee \beta)_R(v') = V$ and $K_R(v') = v$ iff there exists a valuation v' such that $\alpha_R(v') = V$ and $K_R(v') = v$ or there exists a valuation v'' such that $\alpha_R(v'') = V$ and $K_R(v'') = v$ iff $(\alpha K \vee \beta K)_R(v) = V$.

The proof of (n2) is similar. Let us observe that the implication (n2) cannot be replaced by the equivalence.

For example, we can take as α: $x = 1$, β: $x = 2$, K: $[x/2]$, a realization R' such that $1_{R'} \neq 2_{R'}$ and a valuation v_0 such that $v_0(x) = 2_{R'}$.

(n3) $(\alpha \circ [KM])_R(v) = V$ iff there exists a valuation v' such that $\alpha_R(v') = V$ and $M_R(K_R(v')) = v$ iff there exists a valuation v'' such that $M_R(v'') = v$ and there exists a valuation v' such that $\alpha_R(v') = V$ and $K_R(v') = v''$ iff $((\alpha K)M)_R(v) = V$.

The proof of (n4) is similar.

In order to prove (n5) and (n6), let us assume that R is a realization belonging to the class of realizations $\mathscr{R}$.

(n5) Let $\alpha \leqslant_R \beta$ and let v be a valuation such that $(\alpha K)_R(v) = V$. That is, there exists a valuation v' such that $\alpha_R(v') = V$ and $K_R(v') = v$. By assumption, $\beta_R(v') = V$ and hence $(\beta K)_R(v) = V$.

(n6) First suppose that $K\mathbf{1} \wedge \alpha \leqslant_R K\beta$ and let v be a valuation such that $(\alpha K)_R(v) = V$. That is, there exists a valuation v' such that $\alpha_R(v') = V$ and $K_R(v') = v$. Hence $(K\mathbf{1} \wedge \alpha)_R(v') = V$ and by assumption, $(K\beta)_R(v') = V$. Since $v = K_R(v')$, then $\beta_R(v) = \beta_R(K_R(v')) = V$.

Now suppose that $\alpha K \leqslant_R \beta$ and let v be a valuation such that $(K\mathbf{1} \wedge \alpha)_R(v) = V$. That is, denoting $v' = K_R(v)$ we have $(\alpha K)_R(v') = V$ and according to our assumption we get $\beta_R(v') = V$. So $(K\beta)_R(v) = \beta_R(K_R(v)) = \beta_R(v') = V$. ■

2. The iteration of the strongest consequent

Let K be a program, α a formula and R a realization. As it will be seen, the following property of an output v is also expressible in the extended algorithmic logic:

(**) *there exists an input* v' *such that* $\alpha_R(v') = V$ *and* $K^i_R(v') = v$ *for some natural* i.

Let $\vec{x}$ be the sequence of all variables occurring in the expressions K and α and let $\vec{y}$ be the copy of x defined in § 1.

By $\bigcup \alpha K$ we shall denote the following formula:

$$\exists \vec{y}\big(\alpha(\vec{y}) \wedge \bigcup K(\vec{y})(\vec{x} = \vec{y})\big).$$

This formula will be called the *iteration of the strongest consequent* of the formula α with respect to the program K.

2.1. *The following facts are equivalent*:

(1) *a valuation* v *has the property* (**),

(2) $(\bigcup \alpha K)_R(v) = V$,

(3) $\underset{i \in N}{\text{l.u.b.}}\, (\alpha K^i)_R(v) = V$.

Proof: If M is a program and i is a natural number then by $M^{(i)}$ we shall denote the composition of i copies of the program M. That is: $M^{(0)}$: [ ], $M^{(1)}$: M and $M^{(i+1)}$: $\circ[M^{(i)} M]$ for $i \geqslant 1$. On account of the scheme (A7) of the logical axioms and of the equivalence (n3) in 1.3, it follows by induction on i that:

(4) $M^{(i)}\beta \equiv M^i\beta$,

(5) $\beta M^{(i)} \equiv \beta M^i$,

where β is a formula and i is a natural number.

Let us denote by W_v the set of all valuations v' such that $v'(z) = v(z)$ if $z \notin \vec{y}$. Applying (4) we obtain

$$(\bigcup aK)_R(v) = \underset{v' \in W_v}{\text{l.u.b.}} \big(a(\vec{y})_R(v') \wedge \underset{i \in N}{\text{l.u.b.}} \big(K(\vec{y})^{(i)}(\vec{x} = \vec{y})\big)_R(v')\big)$$
$$= \underset{i \in N}{\text{l.u.b.}}\ \underset{v' \in W_v}{\text{l.u.b.}} \big(a(\vec{y})_R(v') \wedge \big(K(\vec{y})^{(i)}(\vec{x} = \vec{y})\big)_R(v')\big).$$

Hence, using the definition of aK and the fact (5), we get

$$(\bigcup aK)_R(v) = \underset{i \in N}{\text{l.u.b.}} (aK^{(i)})_R(v) = \underset{i \in N}{\text{l.u.b.}} (aK^i)_R(v).$$

So conditions (2) and (3) are equivalent. By virtue of 1.1 and (4) and (5), conditions (3) and (1) are equivalent. In fact, $\underset{i \in N}{\text{l.u.b.}} (aK^i)_R(v) = \underset{i \in N}{\text{l.u.b.}} (aK^{(i)})_R(v) = V$ iff there exists a natural number i and a valuation v' such that $a_R(v') = V$ and $K_R^{(i)}(v') = v$ iff condition (**) holds for v. ∎

2.2. *The construction of the iteration of the strongest consequent has the following properties:*

(n7) $(\bigcup aK)_R(v) = \underset{n \in N}{\text{l.u.b.}} (aK^n)_R(v)$,

(n8) $a * [\gamma K] \equiv \neg\gamma \wedge \bigcup a \veebar [\gamma K[\ ]]$,

(n9) $\bigcup a \veebar [\gamma K[\ ]] \equiv a \vee (\gamma \wedge \bigcup a \veebar [\gamma K[\ ]])K$,

(n10) *if for all natural n $aK^n \leqslant_{\mathfrak{R}} \beta$ then $\bigcup aK \leqslant_{\mathfrak{R}} \beta$,*

where a and β are formulas, K is a program, γ is an open formula, R is a realization and $\mathfrak{R}$ is a class of realizations.

Proof: The assertions (n7) and (n10) follow from 2.1.

(n8) Let R' be a realization and v a valuation. Let us consider the relationship between the following facts:

(i) $(a * [\gamma K])_{R'}(v) = V$,

(ii) there exists a valuation v' such that $a_{R'}(v') = V$ and $*[\gamma K]_{R'}(v') = v$,

(iii) there exists a valuation v' such that $a_{R'}(v') = V$, $\veebar[\gamma K[\ ]]_{R'}^n(v') = v$ for some natural n and $\gamma_{R'}(v) = \Lambda$,

(iv) $(\neg\gamma \wedge \bigcup a \veebar [\gamma K[\ ]])_{R'}(v) = V$.

The consecutive equivalence of the above assertions follows from 1.1, from the definition of a realization and from 2.1, respectively.

(n9) Let R' be a realization and v a valuation. Let us consider the relationship between the following facts:

(j) $(\bigcup a \veebar [\gamma K[\ ]])_{R'}(v) = V$,

(jj) there exist a valuation v' and a natural number n such that $a_{R'}(v') = V$ and $\veebar[\gamma K[\ ]]_{R'}^n(v') = v$,

(jjj) there exist a valuation v' and a natural number n such that $a_{R'}(v') = V$, $\veebar[\gamma K[\ ]]_{R'}^n(v') = v$ and for each natural number $m < n$, $\gamma_{R'}(\veebar[\gamma K[\ ]]_{R'}^m(v')) = V$, (jv) either $a_{R'}(v) = V$ (i.e. $n = 0$) or there exist valuations v', v'' and a natural number $n \geq 1$ such that $\gamma_{R'}(v'') = V, \veebar[\gamma K[\ ]]_{R'}^{n-1}(v') = v''$, for each $m < n-1$, $\gamma_{R'}(\veebar[\gamma K[\ ]]_{R'}^{m}(v')) = V$ and $K_{R'}(v'') = v$,

(v) $(a \vee (\gamma \wedge \bigcup a \veebar [\gamma K[\ ]])K)_{R'}(v) = V$.

The consecutive equivalence of the above assertions follows, as previously, solely from 1.1, 2.1 and from the definition of a realization. ∎

In Chapter II we shall need the following lemma:

2.3. *The following facts hold:*

(p1) $*[\gamma K]\beta \equiv (\beta \wedge \neg\gamma) \vee (K * [\gamma K]\beta \wedge \gamma)$,

(p2) $\bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta) \equiv (\neg\gamma \wedge \beta) \vee (\gamma \wedge K \bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta))$,

(p3) $\bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta) \equiv *[\gamma K]\beta \vee \bigcap K\gamma$,

(p4) *if for all natural i, $a \leqslant_{\mathcal{R}} K^i\beta$ then $a \leqslant_{\mathcal{R}} \bigcap K\beta$,*

where $\mathcal{R}$ is a class of realizations, K is a program, γ is an open formula and a, β are arbitrary formulas.

Proof: The fact (p1) follows from the equivalence of the following two programs: $*[\gamma K]$ and $\veebar[\gamma \circ [K * [\gamma K]][\ ]]$.

Now, let R be a realization and v a valuation. In order to prove (p2) it is sufficient to observe that if $\gamma_R(v) = \Lambda$ then $(\bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta))_R(v) = \beta_R(v)$ and if $\gamma_R(v) = V$ then

$$(\veebar[\gamma K[\ ]]^n(\gamma \vee \beta))_R(v) = \begin{cases} V & \text{if} \quad n = 0, \\ (K \veebar [\gamma K[\ ]]^{n-1}(\gamma \vee \beta))_R(v) & \text{if} \quad n > 0. \end{cases}$$

The equivalence (p3) follows from the following remarks. If for every natural i, $K_R^i(v)$ is defined and $\gamma_R(K_R^i(v)) = V$, then $(\bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta))_R(v) = V$ and $(\bigcap K\gamma)_R(v) = V$. If i is in N and for each $j < i$, $K_R^j(v)$ is defined, $\gamma_R(K_R^j(v)) = V$ and $K_R^i(v)$ is undefined, then $(\bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta))_R(v) = \Lambda$, $(\bigcap K\gamma)_R(v) = \Lambda$ and $(*[\gamma K]\beta)_R(v) = \Lambda$.

It remains to consider the case when for every $j \leqslant i$, $K_R^j(v)$ is defined, $\gamma_R(K_R^i(v)) = \Lambda$ and for each $j < i$, $\gamma_R(K_R^j(v)) = V$. In that case $(\bigcap \veebar [\gamma K[\ ]](\gamma \vee \beta))_R(v) = \beta_R(K_R^i(v))$, $(\bigcap K\gamma)_R(v) = \Lambda$ and $(*[\gamma K]\beta)_R(v) = \beta_R(K_R^i(v))$.

Finally, if the inequality $a \leqslant_R K^i\beta$ holds for every i in N and every realization R in $\mathcal{R}$, then for every valuation v, $a_R(v) \leqslant \text{g.l.b.}_{i \in N} (K^i\beta)_R(v)$ and hence $a \leqslant_{\mathcal{R}} \bigcap K\beta$. ∎

3. Correctness and partial correctness of programs

We say that a program K is *correct with respect to an input formula a and an output formula β in a class of realizations $\mathcal{R}$* provided the formula $(a \Rightarrow K\beta)$ is valid in $\mathcal{R}$.

We say that a program K is *partially correct with respect to an input formula α and an output formula β in a class of realizations $\mathcal{R}$* provided the formula $(K\mathbf{1} \wedge \alpha \Rightarrow K\beta)$ is valid in $\mathcal{R}$.

For example, let us consider the following program:

$$M: \circ\big[[t/0\ \ z/x] * [z \geqslant y[z/z-y\ \ t/t+1]]\big]$$

and its algolic realization in the system of integers.

The program M is correct with respect to formulas $(x \geqslant 0 \wedge y > 0)$ and $\big(t \geqslant 0 \wedge t\cdot y \leqslant x \wedge x < (t+1)\cdot y\big)$. On account of the case $y = 0$, M is not correct with respect to formulas $(x \geqslant 0 \wedge y \geqslant 0)$ and $\big(t \geqslant 0 \wedge t\cdot y \leqslant x \wedge x < (t+1)\cdot y\big)$, but it is partially correct with respect to these formulas.

Let $\mathcal{T} = \{\mathcal{L}, \mathcal{C}, \mathcal{A}\}$ be an algorithmic theory.

A program K is said to be *correct with respect to formulas α and β in the theory $\mathcal{T}$* provided the formula $(\alpha \Rightarrow K\beta)$ is a theorem of the theory $\mathcal{T}$.

A program K is said to be *partially correct with respect to formulas α and β in the theory $\mathcal{T}$* provided the formula $(K\mathbf{1} \wedge \alpha \Rightarrow K\beta)$ is a theorem of that theory.

From (n6) in 1.3 it follows that

3.1. (1) *A program K is partially correct with respect to formulas α and β in a class of realizations $\mathcal{R}$ (in the theory $\mathcal{T}$) iff the formula $(\alpha K \Rightarrow \beta)$ is valid in $\mathcal{R}$ (is a theorem of $\mathcal{T}$).*

(2) *A program K is correct with respect to formulas α and β in a class of realizations $\mathcal{R}$ (in the theory $\mathcal{T}$) iff K is partially correct with respect to α, β in $\mathcal{R}$ (in $\mathcal{T}$) and $\models_{\mathcal{R}}(\alpha \Rightarrow K\mathbf{1})$ ($(\alpha \Rightarrow K\mathbf{1})$ is a theorem of $\mathcal{T}$).*

II. MODULAR STRUCTURE OF PROGRAMS

In Chapter II we shall deal with properties of the modular structure of programs. We shall investigate the possibility of deriving properties of programs from appropriate properties of their modular structures. The modular structure is regarded as a tree with modules in vertices. Such a definition has proved useful in the machine implementation of proving correctness of programs [7].

1. Descriptions of programs

By a *module of an FS-program K* we shall understand any subexpression of K which is also a program. The set of all modules of a program K will be denoted by $\mathrm{Mod}(K)$.

A pair $H = (I, \hat{K})$ is said to be a *tree of the program K* if $I \subset \{1, 2\}^*$, $\hat{K}: I \xrightarrow[\text{onto}]{} \mathrm{Mod}(K)$ and the following conditions are fulfilled:

(1) the empty sequence e belongs to I and $\hat{K}_e$: K,

(2) if i is in I and $\hat{K}_i$: $\circ[LM]$ or $\hat{K}_i$: $\veebar[\gamma LM]$ then $i1$, $i2$ are in I and $\hat{K}_{i1}$: L, $\hat{K}_{i2}$: M,

(3) if i is in I and $\hat{K}_i$: $*[\gamma M]$ then $i1$ is in I and $\hat{K}_{i1}$: M,

(4) every element in I can be obtained from the vertex e by means of the rules (2) and (3).

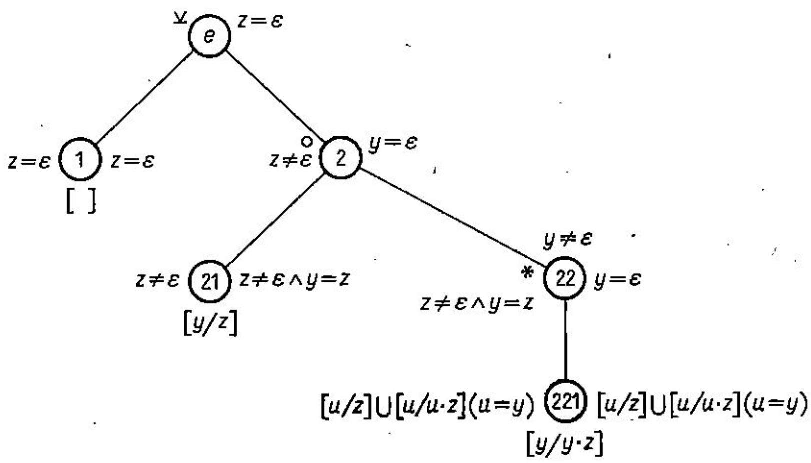
The function $\hat{K}$ is said to be the *modular structure of the program* K and is denoted by $\hat{K} = \{K_i\}_{i\epsilon I}$.

By a *description of the program* K we shall understand any mapping A: $I \to F \times F$. We shall denote it by $A = \{(a_i, b_i)\}_{i\epsilon I}$, where $(a_i, b_i) = A(i)$ for i in I. For every i in I, the pair $A(i) = (a_i, b_i)$ defines a *subtask of the module* $\hat{K}_i$. The formulas a_i and b_i are called the *input formula* and the *output formula of the module* $\hat{K}_i$, respectively. In particular, the integral task is defined by the pair $A(e) = (a_e, b_e)$.

In order to illustrate the notions just introduced, we give below two examples of program trees with associated descriptions. We assume that $\cdot$, $-$, $\in$, $\cup$ are 2-argument functors, f, g are 1-argument functors and ε is a 0-argument functor.

EXAMPLE 1.

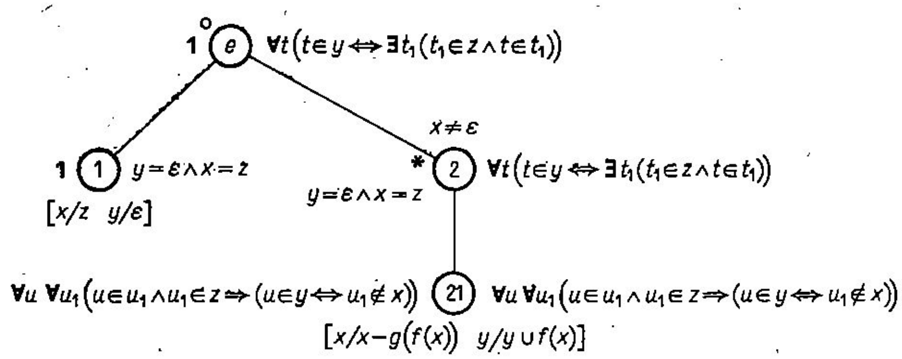
$$K\colon \veebar\Big[z = \varepsilon[\,]\circ\big[[y/z]*[y \neq \varepsilon[y/y\cdot z]]\big]\Big].$$



EXAMPLE 2.

$$M\colon \circ\Big[[x/z\ \ y/\varepsilon]*\big[x \neq \varepsilon[x/x-g(f(x))\ \ y/y\cup f(x)]\big]\Big].$$

Remark: According to the definition, the modular structure of a program is determined uniquely by the program alone. However, in practice, the choice of a partition of a program into modules depends also on the programmer. The results of this chapter will not change if we allow any

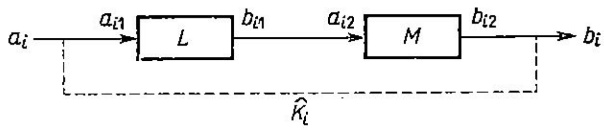


loop-free programs to stand at the terminal vertices of the program tree. Such an approach has been applied in [7].

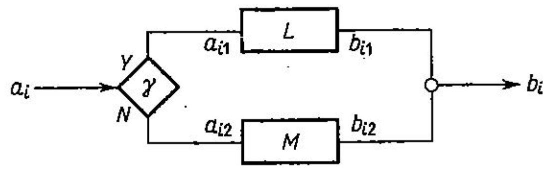
By a *verification condition of a vertex i in I with respect to a description* $A = \{(a_i, b_i)\}_{i\in I}$ we shall understand the formula VC_i defined as follows:

(1) if $\hat{K}_i$: s is a substitution then VC_i: $\big(a_i \Rightarrow (sb_i)'\big)$,

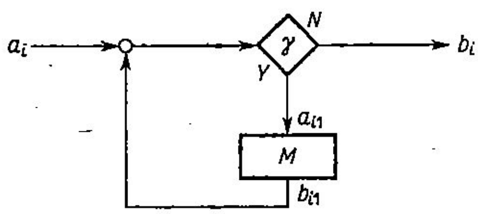
(2) if $\hat{K}_i$: $\circ[LM]$ then VC_i: $\big((a_i \Rightarrow a_{i1}) \wedge (b_{i1} \Rightarrow a_{i2}) \wedge (b_{i2} \Rightarrow b_i)\big)$, i.e.



(3) if $\hat{K}_i$: $\veebar[\gamma LM]$ then VC_i: $\big((a_i \wedge \gamma \Rightarrow a_{i1}) \wedge (a_i \wedge \neg\gamma \Rightarrow a_{i2}) \wedge (b_{i1} \vee b_{i2} \Rightarrow$ $\Rightarrow b_i)\big)$, i.e.



(4) if $\hat{K}_i$: $*[\gamma M]$ then VC_i: $\Big(\big((a_i \vee b_{i1}) \wedge \gamma \Rightarrow a_{i1}\big) \wedge \big((a_i \vee b_{i1}) \wedge \neg\gamma \Rightarrow b_i\big)\Big)$, i.e.



By a *verification condition of the program K with respect to the description A* we shall understand the formula VC: $\bigwedge_{i\in I} \mathrm{VC}_i$. The description A is called *compatible with the modular structure $\hat{K}$ in a class of realizations $\mathcal{R}$ (in a realization R)* if $\models_{\mathcal{R}} \mathrm{VC}$ ($\models_R \mathrm{VC}$).

2. A theorem on compatible descriptions

In this section we assume that $\hat{H} = (I, \hat{K})$ is the tree of a program K, $A = \{(a_i, b_i)\}_{i\in I}$ is a description of K and $\mathcal{R}$ is a class of realizations.

The modular structure K is said to be *correct with respect to A in $\mathcal{R}$* provided A is compatible with the modular structure of the program K in $\mathcal{R}$ and for each i in I, $\models_{\mathcal{R}}(a_i \Rightarrow \hat{K}_i b_i)$.

The modular structure $\hat{K}$ is said to be *partially correct with respect to A in $\mathcal{R}$* provided A is compatible with the modular structure of K in $\mathcal{R}$ and for each i in I, $\models_{\mathcal{R}}(a_i\hat{K}_i \Rightarrow b_i)$.

2.1. *If A is compatible with the modular structure of the program K in $\mathcal{R}$ then the modular structure $\hat{K}$ is partially correct with respect to A in $\mathcal{R}$.*

Proof: Let VC_i be the verification condition of a vertex i in I with respect to the description A. We have to prove that $\models_{\mathcal{R}} \bigwedge_{i\in I} \mathrm{VC}_i$ implies $\models_{\mathcal{R}}(a_i\hat{K}_i \Rightarrow b_i)$ for each i in I.

The proof proceeds by induction on the length of modules of K. If $\hat{K}_i$: s is a substitution then by assumption $\models_{\mathcal{R}} \big(a_i \Rightarrow (sb_i)'\big)$. Since s is a substitution, then $\models s\mathbf{1}$ and therefore $\models_{\mathcal{R}} \big(a_i \Rightarrow (sb_i)'\big)$ is equivalent to $\models_{\mathcal{R}} (a_i \wedge s\mathbf{1} \Rightarrow sb_i)$ and in virtue of I, 3.1 (1), the latter is equivalent to $\models_{\mathcal{R}} (a_i s \Rightarrow b_i)$.

Now let $\hat{K}_i$: $\circ[LM]$ and

(o1) $a_{i1}L \leqslant_{\mathcal{R}} b_{i1}$, $a_{i2}M \leqslant_{\mathcal{R}} b_{i2}$, (induction hypothesis).

According to the assumption $\models_{\mathcal{R}} \mathrm{VC}_i$ we have

(o2) $a_i \leqslant_{\mathcal{R}} a_{i1}$, $b_{i1} \leqslant_{\mathcal{R}} a_{i2}$, $b_{i2} \leqslant_{\mathcal{R}} b_i$.

Applying (o1), (o2) and I, 1.3, (n3) and (n5), we get consecutively $a_i\circ[LM] \equiv a_iLM \leqslant_{\mathcal{R}} b_{i1}M \leqslant_{\mathcal{R}} a_{i2}M \leqslant_{\mathcal{R}} b_{i2} \leqslant_{\mathcal{R}} b_i$.

Now let $\hat{K}_i$: $\veebar[\gamma LM]$ and

($\veebar$1) $a_{i1}L \leqslant_{\mathcal{R}} b_{i1}$, $a_{i2}M \leqslant_{\mathcal{R}} b_{i2}$, (induction hypothesis).

According to the assumption $\vdash_{\mathfrak{R}} VC_i$ we have

$$a_i \wedge \gamma \leqslant_{\mathfrak{R}} a_{i1},$$

$$(\veebar 2) \qquad a_i \wedge \neg\gamma \leqslant_{\mathfrak{R}} a_{i2},$$

$$b_{i1} \vee b_{i2} \leqslant_{\mathfrak{R}} b_i.$$

Applying ($\veebar$1), ($\veebar$2) and I, 1.3, (n4) and (n5), we get consecutively $a_i \veebar [\gamma LM] \equiv (a_i \wedge \gamma)L \vee (a_i \wedge \neg\gamma)\, M \leqslant_{\mathfrak{R}} a_{i1} L \vee a_{i2}\, M \leqslant_{\mathfrak{R}} b_{i1} \vee b_{i2} \leqslant_{\mathfrak{R}} b_i$.

Now we shall carry out the induction step in the case $\hat{K}_i$: $*[\gamma M]$. Let

(*1) $\quad a_{i1} M \leqslant_{\mathfrak{R}} b_{i1}$ $\qquad$ (induction hypothesis)

and

(*2) $\quad (a_i \vee b_{i1}) \wedge \gamma \leqslant_{\mathfrak{R}} a_{i1}, \quad (a_i \vee b_{i1}) \wedge \neg\gamma \leqslant_{\mathfrak{R}} b_i,$ $\qquad$ (assumption).

First we shall show by induction that for every natural number q

(*3) $\quad a_i \veebar [\gamma M[\ ]]^q \leqslant_{\mathfrak{R}} a_i \vee b_{i1}.$

For $q = 0$ this is evident. Suppose that (*3) holds for some q. Using (*3) and I, 1.3, (n4) and (n5), we get

$$a_i \veebar [\gamma M[\ ]]^{q+1} \equiv a_i \veebar [\gamma M[\ ]]^q \veebar [\gamma M[\ ]]$$
$$\leqslant_{\mathfrak{R}} \big((a_i \vee b_{i1}) \wedge \gamma\big) M \vee \big((a_i \vee b_{i1}) \wedge \neg\gamma\big).$$

Hence, by (*1), (*2) and I, 1.3 (n5), we infer that

$$a_i \veebar [\gamma M[\ ]]^{q+1} \leqslant_{\mathfrak{R}} a_{i1} M \vee (a_i \vee b_{i1}) \leqslant_{\mathfrak{R}} a_i \vee b_{i1}.$$

Next, applying the rule I, 2.2 (n10) to the implication (*3) we obtain $\bigcup a_i \veebar [\gamma M[\ ]] \leqslant_{\mathfrak{R}} (a_i \vee b_{i1})$ and hence, by I, 2.2 (n8) and by (*2), it follows that

$$a_i \hat{K}_i \equiv \neg\gamma \wedge \bigcup a_i \veebar [\gamma M[\ ]] \leqslant_{\mathfrak{R}} (a_i = b_{i1}) \wedge \neg\gamma \leqslant_{\mathfrak{R}} b_i.$$

As a corollary to 2.1 and to the theorem on completeness we obtain the following fact.

2.2. *Let* VC *be the verification condition of the program K with respect to the description A. Then, if* VC *is a theorem of an algorithmic theory $\mathcal{T} = \{\mathcal{L}, \mathcal{C}, \mathcal{A}\}$, then the program K is partially correct with respect to the formulas a_e and b_e in the theory $\mathcal{T}$.*

Below we give an example of an application of 2.2 to the examination whether a given program is partially correct with respect to a given task.

EXAMPLE. Let Boolf be the formal algorithmic theory of Boolean algebras with the following additional axiom:

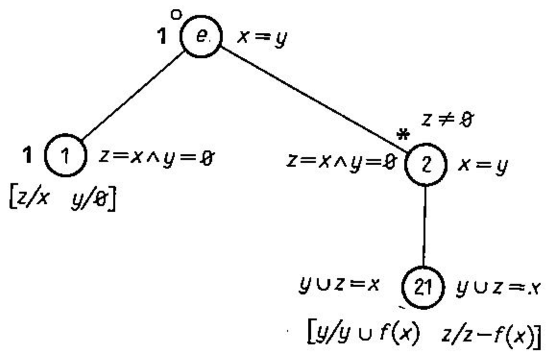
(Af) $$x \cup f(x) = x,$$

where f is an additional 1-argument functor.

Let us consider the program

$$M\colon \circ\Big[[z/x\ \ y/\emptyset]*\big[z \neq \emptyset\,[y/y\cup f(x)\ \ z/z-f(x)]\big]\Big]$$

and the task defined by the formulas $\mathbf{1}$ and $x = y$. Let us assume that we want to prove the partial correctness of M with respect to the above task in the theory Boolf. In order to apply Theorem 2.2 we must construct a description and prove the appropriate verification condition. Let us consider the following description:



Then

$\mathrm{VC}_e\colon \big((\mathbf{1} \Rightarrow \mathbf{1})\wedge (z = x\wedge y \neq \emptyset \Rightarrow z = x\wedge y \neq \emptyset)\wedge (x = y \Rightarrow x = y)\big)$,

$\mathrm{VC}_1\colon (\mathbf{1} \Rightarrow x = x\wedge \emptyset = \emptyset)$,

$\mathrm{VC}_2\colon \Big(\big(((z = x\wedge y = \emptyset)\vee y\cup z = x\big)\wedge z = \emptyset \Rightarrow x = y\big)\big(((z = x\wedge y = \emptyset)\vee y\cup z = x\big)\wedge z \neq \emptyset \Rightarrow y\cup z = x\big)\Big)$,

$\mathrm{VC}_{21}\colon \Big(y\cup z = x \Rightarrow \big(y\cup f(z)\cup (z-f(z))\big) = x\Big)$.

Evidently, VC_e, VC_1 and VC_2 are tautologies and it remains to prove that VC_{21} is a theorem of the theory Boolf. Firstly, $\big(y\cup f(z)\big)\cup\big(z-f(z)\big) = y\cup z\cup f(z)$ is a theorem of the theory of Boolean algebras. Moreover, by the axiom (Af), the formula $z\cup f(z) = z$ is a theorem of Boolf. Hence VC_{21} is also a theorem.

3. Extremal descriptions

In the extended algorithmic logic we can express the greatest relation of an input data v such that a program K halts for v and the output data satisfies a formula β. Namely, this relation is defined by the formula $K\beta$. Analogously (see I, 1), the formula αK defines the least relation of results v which are obtainable from input data satisfying a formula α. In this section we shall prove that the extended algorithmic logic is large enough to enable defining analogous relations for modular structures of programs.

Let us assume that K is a fixed program, $H = (I, \hat{K})$ is the tree of K and $\mathscr{R}$ is a class of realizations. Let $A = \{(a_i, b_i)\}_{i \in I}$ and $B = \{(c_i, d_i)\}_{i \in I}$ be descriptions of K. By $\leqslant_{\mathscr{R}}$ and $\equiv_{\mathscr{R}}$ we shall mean the following relations:

$$A \leqslant_{\mathscr{R}} B \quad \text{iff} \quad a_i \leqslant_{\mathscr{R}} c_i \text{ and } b_i \leqslant_{\mathscr{R}} d_i \text{ for all } i \text{ in } I,$$

$$A \equiv_{\mathscr{R}} B \quad \text{iff} \quad A \leqslant_{\mathscr{R}} B \text{ and } B \leqslant_{\mathscr{R}} A.$$

By the *natural description* $a\hat{K} = \{(a_i, b_i)\}_{i \in I}$ *of the program* K *with respect to an input formula* a we shall mean the following description:

(1) a_e: a, b_e: aK,

(2) if $i \in I$ and $\hat{K}_i$: $\circ[LM]$, then a_{i1}: a_i, b_{i1}: $a_i L$, a_{i2}: $a_i L$ and b_{i2}: $a_i LM$,

(3) if $i \in I$ and $\hat{K}_i$: $\veebar[\gamma LM]$, then a_{i1}: $a_i \wedge \gamma$, a_{i2}: $a_i \wedge \neg\gamma$, b_{i1}: $a_{i1} L$ and b_{i2}: $a_{i2} M$,

(4) if $i \in I$ and $\hat{K}_i$: $*[\gamma M]$ then a_{i1}: $\gamma \wedge \bigcup a_i \veebar [\gamma M[\;]]$ and b_{i1}: $a_{i1} M$.

By the *natural description* $\hat{K}\beta = \{(a_i, b_i)\}_{i \in I}$ *of the program* K *with respect to an output formula* β we shall mean the following description:

(1) a_e: $K\beta$, b_e: β,

(2) if $i \in I$ and $\hat{K}_i$: $\circ[LM]$, then a_{i1}: LMb_i, b_{i1}: Mb_i, a_{i2}: b_{i1} and b_{i2}: b_i,

(3) if $i \in I$ and $\hat{K}_i$: $\veebar[\gamma LM]$, then a_{i1}: Lb_i, a_{i2}: Mb_i, b_{i1}: b_i and b_{i2}: b_i,

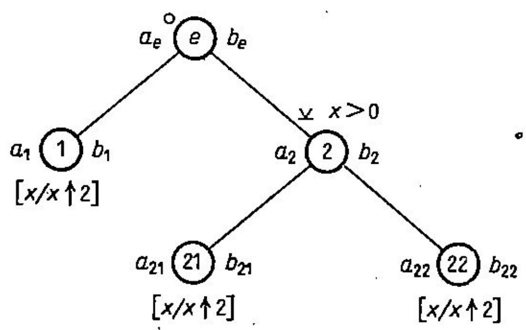
(4) if $i \in I$ and $\hat{K}_i$: $*[\gamma M]$, then b_{i1}: $\bigcap \veebar [\gamma M[\,]](\gamma \vee b_i)$ and a_{i1}: Mb_{i1}.

Let us consider an algorithmic language containing 2-argument functors $\uparrow$ and $+$, 1-argument functor abs, 0-argument functors 1, 2 and 2-argument predicate $>$. Let $\mathscr{R}$ be the 1-element class containing the usual algolic realizations of the above constants in the set of integers.

EXAMPLE 1. Let us consider the program

$$M\colon \circ\big[[x/x\uparrow 2]\veebar[x > 0[x/x\uparrow 2][x/x\uparrow 2]]\big],$$

its tree



and the following four descriptions:

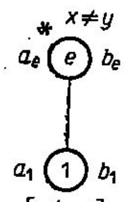
	a_e, a_1	b_1	a_2	a_{21}	b_{21}	a_{22}	b_{22}	b_2, b_e
E_1	$\mathrm{abs}(x) = 1$	$x = 1$	$x = 1$	$\mathrm{abs}(x) = 1$	$x = 1$	$x = 1$	$x = 1$	$x = 1$
E_2	$\mathrm{abs}(x) = 1$	$x = 1$	$x = 1$	$\mathrm{abs}(x) = 1$	$x = 1$	0	0	$x = 1$
E_3	$\mathrm{abs}(x) = 1$	$\mathrm{abs}(x) = 1$	$\mathrm{abs}(x) = 1$	$\mathrm{abs}(x) = 1$	$x = 1$	$\mathrm{abs}(x) = 1$	$x = 1$	$x = 1$
E_4	$\mathrm{abs}(x) = 1$	$x = 1$	$\mathrm{abs}(x) = 1$	$\mathrm{abs}(x) = 1$	$x = 1$	0	0	$x = 1$

As can be easily verified, the modular structure $\hat{M}$ is correct with respect to every one from the above descriptions, and moreover, $E_2 \equiv_{\mathscr{R}} (\mathrm{abs}(x) = 1)\hat{M}$ and $E_3 \equiv_{\mathscr{R}} \hat{M}(x = 1)$.

Now let us consider the program

$$P\colon * [x \neq y\, [x/x+1]],$$

its tree



and the following four descriptions:

	a	a_1	b_1	b_e
Q_1	1	1	1	1
Q_2	1	$x \neq y$	$\exists z(z \neq y \wedge \wedge x = z+1)$	$x = y$
Q_3	$y > 0 \wedge (y > x \vee y = x)$	$y > 0 \vee x+1 > y$	$y > 0 \vee x > y$	$y > 0$
Q_4	$y > 0 \wedge (y > x \vee y = x)$	$y > 0 \wedge (y > x+1 \vee \vee y = x+1)$	$y > 0 \wedge (y > x \vee \vee y = x)$	$y > 0 \wedge x = y$

As can be easily verified, the modular structure $\hat{P}$ is correct with respect to the descriptions Q_3 and Q_4. $\hat{P}$ is not correct with respect to Q_1 and Q_2. It is only partially correct in those cases. Moreover, $Q_2 \equiv_{\mathscr{R}} \mathbf{1}\hat{P}$ and $Q_3 \equiv_{\mathscr{R}} \hat{P}(y > 0)$.

4. Theorems on extremal descriptions

The natural descriptions have the following properties:

4.1. *Let α and β be arbitrary formulas. Let $A = \{(a_i, b_i)\}_{i\in I}$ be a description compatible with the modular structure $\hat{K}$ in a class of realizations $\mathfrak{R}$.*

(1) *The modular structure $\hat{K}$ is partially correct with respect to the natural description $\alpha\hat{K}$ in $\mathfrak{R}$.*

(2) *The modular structure $\hat{K}$ is partially correct with respect to A in $\mathfrak{R}$ iff $a_e\hat{K} \leqslant_{\mathfrak{R}} A$.*

(3) *The modular structure $\hat{K}$ is correct with respect to the natural description $\hat{K}\beta$ in $\mathfrak{R}$.*

(4) *The modular structure $\hat{K}$ is correct with respect to A in $\mathfrak{R}$ iff $A \leqslant_{\mathfrak{R}} \hat{K}b_e$.*

Proof of (1): Let $\alpha\hat{K} = \{(c_i, d_i)\}_{i\in I}$ be the natural description of the program K with respect to the formula α. The definition of $\alpha\hat{K}$ (see § 3) implies that for all i in I

$$d_i\colon\; c_i\hat{K}_i. \tag{4.1.1}$$

So it remains to prove that for each i in I, $\models_{\mathfrak{R}} \mathrm{VC}_i$, where VC_i is the verification condition of the vertex i in I with respect to the description $\alpha\hat{K}$. The proof proceeds by induction on the length of modules $\hat{K}_i$, for i in I.

If $\hat{K}_i\colon s$ is a substitution then $\mathrm{VC}_i \equiv (c_i \Rightarrow (s(c_i s))') \equiv (c_i \Rightarrow s(c_i s)) \equiv (c_i \wedge s\mathbf{1} \Rightarrow s(c_i s))$. Further, on account of I, 3.1(1), $\models_{\mathfrak{R}} (c_i \wedge s\mathbf{1} \Rightarrow s(c_i s))$ is equivalent to $\models_{\mathfrak{R}} (c_i s \Rightarrow c_i s)$. Hence the formula VC_i is valid in $\mathfrak{R}$.

In the cases $\hat{K}_i\colon \circ[LM]$, $\hat{K}_i\colon \veebar[\gamma LM]$, and $\hat{K}_i\colon *[\gamma M]$ we immediately apply the definitions of the verification condition and of the natural description $\alpha\hat{K}$ and the facts I, 1.3 (n3), (n4), and I, 2.2 (n9), respectively.

Proof of (2): On account of Theorem 2.1 it is sufficient to show that if the modular structure $\hat{K}$ is partially correct with respect to A in $\mathfrak{R}$ then $a_e K \leqslant_{\mathfrak{R}} A$, where $a_e\hat{K} = \{(c_i, d_i)\}_{i\in I}$ is the natural description of the program K with respect to $c_e\colon a_e$.

We have to prove that for each i in I

$$c_i \leqslant_{\mathfrak{R}} a_i, \tag{4.2.1}$$

$$c_i\hat{K}_i \leqslant_{\mathfrak{R}} b_i. \tag{4.2.2}$$

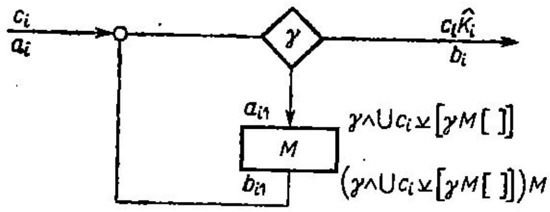
Since A is a compatible description, then by Theorem 2.1 it follows that the modular structure $\hat{K}$ is partially correct with respect to A in $\mathfrak{R}$. Hence (4.2.1) implies (4.2.2). The proof of the assertion (4.2.1) proceeds by induction on the length of an index i in I.

If $i = e$ then $c_e : a_e$. Now suppose that condition (4.2.1) holds for some i.

If $\hat{K}_i$: $\circ[LM]$, then applying the definition of $a_e\hat{K}$, the induction hypothesis (4.2.1) and the assumption that A is a compatible description, we get that $c_{i1} \equiv c_i \leqslant_{\mathcal{R}} a_i \leqslant_{\mathcal{R}} a_{i1}$. In virtue of (4.2.2) we can analogously prove that $c_{i2} \leqslant_{\mathcal{R}} a_{i2}$.

The considerations in the case $\hat{K}_i$: $\veebar[\gamma LM]$ are similar.

Now let us examine the case $\hat{K}_i$: $*[\gamma M]$.



In accordance with the definition of the natural description $a_e\hat{K}$, we must show that $\gamma\wedge\bigcup c_i \veebar [\gamma M[\ ]] \leqslant_{\mathcal{R}} a_{i1}$. On account of the rule I, 2.2 (n10) it is sufficient to prove that for each q in N

(4.2.3) $$\left(\gamma\wedge c_i \veebar [\gamma M[\ ]]^q\right) \leqslant_{\mathcal{R}} a_{i1}.$$

The proof of this fact proceeds by induction on a number q. For $q = 0$, (4.2.3) follows from (4.2.1) and from the assumption of compatibility of A. Suppose that the fact (4.2.3) holds for some q. Hence by I, 1.3 (n4) it follows that

$$\gamma\wedge c_i \veebar [\gamma M[\ ]]^{q+1} \leqslant_{\mathcal{R}} \gamma\wedge\Big(\left(\gamma\wedge c_i \veebar [\gamma M[\ ]]^q\right) M \vee \left(\neg\gamma\wedge c_i \veebar [\gamma M[\ ]]^q\right)\Big)$$
$$\leqslant_{\mathcal{R}} \gamma\wedge\left(\gamma\wedge c_i \veebar [\gamma M[\ ]]^q\right) M \leqslant_{\mathcal{R}} \gamma\wedge a_{i1}\, M.$$

Since the modular structure $\hat{K}$ is partially correct with respect to A in $\mathcal{R}$, then $\gamma\wedge a_{i1}\, M \leqslant_{\mathcal{R}} \gamma\wedge b_{i1} \leqslant_{\mathcal{R}} a_{i1}$, which completes the proof of (4.2.3), (4.2.1) and of the whole assertion (2).

Proof of (3): Let $\hat{K}\beta = \{(c_i, d_i)\}_{i\in I}$ be the natural description of the program K with respect to the formula β. The definition of $\hat{K}\beta$, § 3 implies that for each i in I

(4.3.1) $$c_i:\ \hat{K}_i d_i.$$

So it remains to prove that for each i in I we have $\vdash_{\mathcal{R}} \mathrm{VC}_i$, where VC_i is the verification condition of the vertex i in I with respect to the description $\hat{K}\beta$. The proof proceeds by induction on the length of modules $\hat{K}_i$ for i in I.

In the cases when $\hat{K}_i$: s is a substitution, $\hat{K}_i$: $\circ[LM]$, or $\hat{K}_i$: $\veebar[\gamma LM]$ we apply the appropriate definitions, [6], 5.1, and the schemata (A7), (A8) of axioms, [6], 4, respectively. In order to cope with the case $\hat{K}_i$: $*[\gamma M]$ it is sufficient to observe that I, 2.3 (p1), (p2), (p3) imply the following facts:

$$*[\gamma M]b_i \wedge \neg\gamma \leqslant b_i,$$
$$\bigcap \veebar [\gamma M[\ ]] (\gamma \vee b_i) \wedge \neg\gamma \leqslant b_i,$$
$$\bigcap \veebar [\gamma M[\ ]] (\gamma \vee b_i) \wedge \gamma \leqslant M \bigcap \veebar [\gamma M[\ ]] (\gamma \vee b_i),$$
$$*[\gamma M]b_i \leqslant \bigcap \veebar [\gamma M[\ ]] (\gamma \vee b_i).$$

Proof of (4): Let $\hat{K}b_e = \{(\hat{K}_i c_i, c_i)\}_{i\in I}$ be the natural description with respect to the formula c_e: b_e.

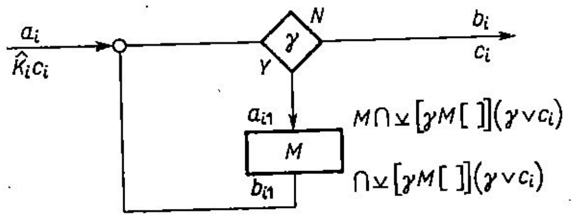
Suppose that the modular structure $\hat{K}$ is correct with respect to the description A in $\mathfrak{R}$. We have to show that for each i in I

(4.4.1) $$a_i \leqslant_{\mathfrak{R}} \hat{K}_i c_i,$$

(4.4.2) $$b_i \leqslant_{\mathfrak{R}} c_i.$$

Since K is correct with respect to A in $\mathfrak{R}$, then assertion (4.4.2) implies (4.4.1). The proof of (4.4.2) proceeds by induction on the length of an index i in I. If $i = e$ then c_e: b_e.

We shall consider only the case $\hat{K}_i$: $*[\gamma M]$.



From $b_i \leqslant_{\mathfrak{R}} c_i$ we have to deduce that $b_{i1} \leqslant_{\mathfrak{R}} \bigcap \veebar [\gamma M[\ ]] (\gamma \vee c_i)$. In virtue of the rule I, 2.3 (p4) it is sufficient to show that for all q in N

(4.4.3) $$b_{i1} \leqslant_{\mathfrak{R}} \veebar [\gamma M[\ ]]^q (\gamma \vee c_i).$$

We shall use induction with respect to q. For $q = 0$, on account of the assumption on correctness of A and the induction hypothesis (4.4.2), we get that $b_{i1} \equiv (\gamma \wedge b_{i1}) \vee (\neg\gamma \wedge b_{i1}) \leqslant_{\mathfrak{R}} \gamma \vee b_i \leqslant_{\mathfrak{R}} \gamma \vee c_i$. Suppose that (4.4.3) holds for some q. On account of the correctness of A we obtain that

$$b_{i1} \equiv (\gamma \wedge b_{i1}) \vee (\neg\gamma \wedge b_{i1}) \leqslant_{\mathfrak{R}} (\gamma \wedge a_{i1}) \vee (\neg\gamma \wedge b_{i1}) \leqslant_{\mathfrak{R}} (\gamma \wedge M b_{i1}) \vee (\neg\gamma \wedge b_{i1})$$
$$\equiv \veebar [\gamma M[\ ]]\, b_{i1}.$$

According to (4.4.3) it follows that

$$b_{i1} \leqslant_{\mathscr{R}} \underline{\vee} [\gamma M[\ ]] \underline{\vee} [\gamma M[\ ]]^q (\gamma \vee c_i) \equiv \underline{\vee} [\gamma M[\ ]]^{q+1} (\gamma \vee c_i).$$

So the inequalities (4.4.3) and (4.4.2) are true.

Now let us assume that $A \leqslant_{\mathscr{R}} \hat{K} b_e = \{(\hat{K}_i c_i, c_i)\}_{i \in I}$. Hence for every i in I, $a_i \leqslant_{\mathscr{R}} \hat{K}_i c_i \leqslant_{\mathscr{R}} \hat{K}_i \mathbf{1}$. Since A is a description compatible with $\hat{K}$ in $\mathscr{R}$, then by Theorem 2.1 for every i in I, $a_i \wedge \hat{K}_i \mathbf{1} \leqslant_{\mathscr{R}} b_i$. So $a_i \leqslant_{\mathscr{R}} \hat{K}_i b_i$ for all i in I, i.e. the modular structure K is correct with respect to A in $\mathscr{R}$. ■

In the sequel we shall assume that the sets of predicates ψ_n, for all n in N, are enumerable. Now we return to the problem which was formulated at the beginning of § 3.

Let $H = (I, \hat{K})$ be the tree of a program K and let α and β be formulas. Let $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{z} = (z_1, \ldots, z_m)$ be sequences of all different individual and propositional variables, respectively, which occur in the expressions K, α and β. Let Γ be the set of all sequences $\gamma = (\gamma_1, \ldots, \gamma_m)$ such that γ_i is equal to z_i or $\neg z_i$ for every $i = 1, \ldots, m$. Let $\{\varepsilon_i^\gamma\}_{i \in I}^{\gamma \in \Gamma}$ and $\{\sigma_i^\gamma\}_{i \in I}^{\gamma \in \Gamma}$ be sequences of different n-argument predicates not occurring in K, α, and β.

By a *predicate description of the program K* we shall mean the description $P = \{(P_i^1, P_i^2)\}_{i \in I}$ such that for every i in I

$$P_i^1: \bigwedge_{\gamma = (\gamma_1, \ldots, \gamma_m) \in \Gamma} \big(\gamma_1 \wedge \ldots \wedge \gamma_m \Rightarrow \varepsilon_i^\gamma(\vec{x})\big),$$

$$P_i^2: \bigwedge_{\gamma = (\gamma_1, \ldots, \gamma_m) \in \Gamma} \big(\gamma_1 \wedge \ldots \wedge \gamma_m \Rightarrow \sigma_i^\gamma(\vec{x})\big).$$

Let R be a fixed realization in a non-empty set J and a two-element Boolean algebra B_0.

By a *relational description of the program K* we shall understand a sequence $\mu = \{(\mu_i^1, \mu_i^2)\}_{i \in I}$ where $\mu_i^k\colon J^n \times B_0^m \to B_0$ for each i in I and $k = 1, 2$.

By a *realization generated by the relational description* $\mu = \{(\mu_i^1, \mu_i^2)\}_{i \in I}$ we shall mean the realization R_μ defined as follows:

(1) for every functor or predicate not appearing in the sequences $\{\varepsilon_i^\gamma\}_{i \in I}^{\gamma \in \Gamma}$ and $\{\sigma_i^\gamma\}_{i \in I}^{\gamma \in \Gamma}$ the realization R_μ coincides with R;

(2) for every i in I, every sequence $\gamma = (\gamma_1, \ldots, \gamma_m)$ in Γ and every objects $j_1, \ldots, j_n$ in J:

$$\varepsilon_{iR_\mu}^\gamma(j_1, \ldots, j_n) = \mu_i^1(j_1, \ldots, j_n, \eta_1, \ldots, \eta_m),$$

$$\sigma_{iR_\mu}^\gamma(j_1, \ldots, j_n) = \mu_i^2(j_1, \ldots, j_n, \eta_1, \ldots, \eta_m),$$

where

$$\eta_k = \begin{cases} \vee & \text{if} \quad \gamma_k\colon z_k \\ \wedge & \text{if} \quad \gamma_k\colon \neg z_k \end{cases} \qquad \text{for} \quad k = 1, \ldots, m.$$

Let us observe that the realization R_μ has the following property: for all i in I, $j_1, \ldots, j_n$ in J and $\eta_1, \ldots, \eta_m$ in B_0,

$$P^1_{iR_\mu}(j_1, \ldots, j_n, \eta_1, \ldots, \eta_m) = \mu^1_i(j_1, \ldots, j_n, \eta_1, \ldots, \eta_m)$$

and

$$P^2_{iR_\mu}(j_1, \ldots, j_n, \eta_1, \ldots, \eta_m) = \mu^2_i(j_1, \ldots, j_n, \eta_1, \ldots, \eta_m).$$

Let M_a be the set of all relational descriptions $\mu = \{(\mu^1_i, \mu^2_i)\}_{i\in I}$ such that

(1) $P^1_{eR}(v) = a_R(v)$ for every valuation v,

(2) the modular structure $\hat{K}$ is partially correct with respect to the predicate description P in the class $\{R_\mu\}$.

Let M^β be the set of all relational descriptions $\mu = \{(\mu^1_i, \mu^2_i)\}_{i\in I}$ such that

(1) $P^2_{eR}(v) = \beta_R(v)$ for every valuation v,

(2) the modular structure $\hat{K}$ is correct with respect to the predicate description P in the class $\{R_\mu\}$.

4.2. *If* $a\hat{K} = \{(a_i, b_i)\}_{i\in I}$ *and* $\hat{K}\beta = \{(c_i, d_i)\}_{i\in I}$ *are natural descriptions, then*

(1) *for every relational description* $\mu = \{(\mu^1_i, \mu^2_i)\}_{i\in I}$ *in* M_a, *for every* i *in* I $a_{iR} \subseteq \mu^1_i$ *and* $b_{iR} \subseteq \mu^2_i$,

(2) *for every relational description* $\varrho = \{(\varrho^1_i, \varrho^2_i)\}_{i\in I}$ *in* M^β *and every* i *in* I $\varrho^1_i \subseteq c_{iR}$ *and* $\varrho^2_i \subseteq d_{iR}$.

Proof: Since the formulas a_i, b_i, c_i, and d_i, for i in I, do not contain any predicate from the sequences $\{\varepsilon^\gamma_i\}^{\gamma\in\Gamma}_{i\in I}$ and $\{\sigma^\gamma_i\}^{\gamma\in\Gamma}_{i\in I}$, we have

$$a_{iR} = a_{iR_\mu}, \; b_{iR} = b_{iR_\mu} \quad \text{for all } \mu \text{ in } M_a$$

and

$$c_{iR} = c_{iR_\varrho}, \; d_{iR} = d_{iR_\varrho} \quad \text{for all } \varrho \text{ in } M^\beta.$$

Now we can apply Theorem 4.1 to the classes $\{R_\mu\}$ and $\{R_\varrho\}$, respectively, getting that

$$\text{for every } \mu \text{ in } M_a, \quad a\hat{K} \leqslant_{\{R_\mu\}} P$$

and

$$\text{for every } \varrho \text{ in } M^\beta, \quad P \leqslant_{\{R_\varrho\}} \hat{K}\beta.$$

Hence by the equalities above we obtain assertions (1) and (2). ■

5. An extension of a task of a program to a description

5.1. *For every program K, every formulas α, β, γ and δ and every class of realizations $\mathscr{R}$*

(1) *if K is partially correct with respect to α and β in $\mathscr{R}$, then the modular structure $\hat{K}$ is partially correct with respect to the description A such that $A(e) = (\alpha, \beta)$ and $A(i) = (\alpha\hat{K})(i)$ for all i in $I-\{e\}$;*

(2) *if K is correct with respect to the formulas γ and δ in $\mathscr{R}$, then the modular structure $\hat{K}$ is correct with respect to the description C such that $C(e) = (\gamma, \delta)$ and $C(i) = (\hat{K}\delta)(i)$ for all i in $I-\{e\}$.*

Proof: Let VC^1, VC^2, VC^3 and VC^4 be the verification conditions of the program K with respect to the descriptions $\alpha\hat{K}$, A, $\hat{K}\beta$ and C, respectively. By our assumptions we have $\alpha K \leqslant_{\mathscr{R}} \beta$ and $\gamma \leqslant_{\mathscr{R}} K\delta$. Hence $VC^1 \leqslant_{\mathscr{R}} VC^2$ and $VC^3 \leqslant_{\mathscr{R}} VC^4$. On account of assertions (1) and (3) of Theorem 4.1, $\models_{\mathscr{R}} VC^1$ and $\models_{\mathscr{R}} VC^3$. So $\models_{\mathscr{R}} VC^2$ and $\models_{\mathscr{R}} VC^4$ and therefore the descriptions A and C are compatible with $\hat{K}$ in $\mathscr{R}$. From the assumptions on A and C and from assertions (2) and (4) of Theorem 4.1 it follows that $\hat{K}$ is partially correct with respect to A in $\mathscr{R}$ and is correct with respect to C in $\mathscr{R}$. ∎

Now we shall define a certain deductive system for proving partial correctness of programs. Let $\mathscr{R}$ be a fixed class of realizations. We assume the following formulas as axioms:

$$(\alpha[\;] \Rightarrow \beta) \quad \text{if} \quad \models_{\mathscr{R}} (\alpha \Rightarrow \beta) \text{ and } \big((s\beta)'s \Rightarrow \beta\big)$$

for all formulas α, β and substitutions s.

We admit also the following rules of inference:

$$(\text{R}1)\ \frac{\delta[\;] \Rightarrow \alpha,\ \alpha K \Rightarrow \beta}{\delta K \Rightarrow \beta}, \qquad (\text{R}2)\ \frac{\alpha K \Rightarrow \beta,\ \beta[\;] \Rightarrow \delta}{\alpha K \Rightarrow \delta},$$

$$(\text{R}3)\ \frac{\alpha K \Rightarrow \beta,\ \beta M \Rightarrow \delta}{\alpha \circ [KM] \Rightarrow \delta}, \qquad (\text{R}4)\ \frac{(\alpha\wedge\gamma)K \Rightarrow \beta,\ (\alpha\wedge \neg\gamma)\, M \Rightarrow \beta}{\alpha \veebar [\gamma K M] \Rightarrow \beta},$$

$$(\text{R}5)\ \frac{\alpha_1 K \Rightarrow \beta_1,\ \big((\alpha\vee\beta_1)\wedge\gamma\big)[\;] \Rightarrow \alpha_1,\ \big((\alpha\vee\beta_1)\wedge\neg\gamma\big)[\;] \Rightarrow \beta}{\alpha * [\gamma K] \Rightarrow \beta},$$

for all programs K, M, open formulas γ and arbitrary formulas $\alpha, \beta, \delta, \alpha_1, \beta_1$.

5.2. *For every program K and every formulas α and β, K is partially correct with respect to α and β in $\mathscr{R}$ iff the formula $(\alpha K \Rightarrow \beta)$ is provable in the formal system defined above.*

Proof: As is easy to verify, the axioms of the above system are valid and the rules are consistent, i.e. they allow to deduce from valid premises the appropriate valid conclusions. So if the formula $(\alpha K \Rightarrow \beta)$ is provable then the program K is partially correct with respect to the formulas α and β. Let us assume that $H = (I, \hat{K})$ is the tree of the program K and that K is partially correct with respect to α and β in $\mathcal{R}$. By 5.1, (1) it follows that there exists a description $A = \{(a_i, b_i)\}_{i\in I}$ compatible with $\hat{K}$ in $\mathcal{R}$ and such that a_e: α and b_e: β. Let VC be the verification condition of K with respect to A. Thus $\models_{\mathcal{R}}$ VC. Now it is easy to construct a proof p of the formula $(\alpha K \Rightarrow \beta)$ from the verification condition VC. We shall define by induction on the length of ith module the proof p_i where i belongs to I.

Namely, if $\hat{K}_i$: s is a substitution then

$$p_i = \{(a_i[\ ] \Rightarrow (sb_i)'),\ ((sb_i)'s \Rightarrow b_i),\ (a_i s \Rightarrow b_i)\}.$$

If $\hat{K}_i$: $\circ[LM]$, then

$$p_i = \{(a_i[\ ] \Rightarrow a_{i1}),\ p_{i1},\ (b_{i1}[\ ] \Rightarrow a_{i2}),\ p_{i2},\ (b_{i2}[\ ] \Rightarrow b_i), (a_i\hat{K}_i \Rightarrow b_i)\}.$$

If $\hat{K}_i$: $\veebar[\gamma LM]$, then

$$p_i = \{((a_i \wedge \gamma)[\ ] \Rightarrow a_{i1}),\ p_{i1},\ (b_{i1}[\ ] \Rightarrow b_i),\ ((a_i \wedge \neg\gamma)[\ ] \Rightarrow a_{i2}),$$
$$p_{i2},\ (b_{i2}[\ ] \Rightarrow b_i),\ (a_i\hat{K}_i \Rightarrow b_i)\}.$$

If $\hat{K}_i$: $*[\gamma M]$, then

$$p_i = \{(((a_i \vee b_{i1}) \wedge \gamma)[\ ] \Rightarrow a_{i1}),\ (((a_i \vee b_{i1}) \wedge \neg\gamma)[\ ] \Rightarrow b_i),\ p_{i1},\ (a_i K_i \Rightarrow b_i)\}.$$

In this way p_e is the proof of the formula $(\alpha K \Rightarrow \beta)$. ∎

EXAMPLE. Now let us return to the example in §2. Let $\mathcal{R}_{\mathrm{Boolf}}$ be the class of all models of the theory Boolf. Below we shall give a proof of the formula $(\mathbf{1}M \Rightarrow x = y)$ for the class of realizations $\mathcal{R}_{\mathrm{Boolf}}$.

(1) $(\mathbf{1}[\ ] \Rightarrow (x = x \wedge \emptyset = \emptyset))$ axiom,

(2) $((x = x \wedge \emptyset = \emptyset)[z/x\ y/\emptyset] \Rightarrow (z = x \wedge y = \emptyset))$ axiom,

(3) $(\mathbf{1}[z/x\ y/\emptyset] \Rightarrow (z = x \wedge y = \emptyset))$ (R1)(1, 2),

(4) $((y \cup z = x)[\ ] \Rightarrow ((y \cup f(x) \cup (z - f(x))) = x))$ axiom,

(5) $(((y \cup f(x) \cup (z - f(x))) = x[y/y \cup f(x)\ z/z - f(x)] \Rightarrow (y \cup z = x))$ axiom,

(6) $(y \cup z = x[y/y \cup f(x)\ z/z - f(x)] \Rightarrow y \cup z = x)$ (R1)(4, 5),

(7) $((((z = x \wedge y = \emptyset) \vee y \cup z = x) \wedge z \neq \emptyset)[\ ] \Rightarrow y \cup z = x)$ axiom,

(8) $((((z = x \wedge y = \emptyset) \vee y \cup z = x) \wedge z = \emptyset)[\ ] \Rightarrow x = y)$ axiom,

(9) $((z = x \wedge y = \emptyset) * [z \neq \emptyset[y/y \cup f(x)\ z/z - f(x)]] \Rightarrow x = y)$ (R5)(6, 8, 7),

(10) $(\mathbf{1} \circ [[z/x\ y/\emptyset] * [z \neq \emptyset[y/y \cup f(x)\ z/z - f(x)]]] \Rightarrow x = y)$ (R3)(3, 9).

6. Open descriptions

In practice we meet usually descriptions consisting solely of open formulas. Such descriptions will be called *open*. The following theorem establishes the complexity degrees of properties of a modular structure with respect to open descriptions.

6.1. *In every class of realizations* $\mathscr{R}$

(1) *the properties of partial correctness of a modular structure with respect to open descriptions and of validity of open formulas are recursively reducible to each other;*

(2) *the properties of correctness of a modular structure with respect to open descriptions and of correctness of a program with respect to open formulas are recursively reducible to each other.*

Proof: Assertion (1) results from Theorem 2.1 and the following simple fact: for an arbitrary formula α, $\alpha \equiv (\mathbf{1}[\;] \Rightarrow \alpha)$.

Now we consider assertion (2). Let K be a program correct with respect to open formulas α and β in a class of realizations $\mathscr{R}$. Observe that the correctness of K is reducible to the stop property of the program M: $\veebar\big[\alpha\circ\big[K*[\neg\beta[\;]]\big][\;]\big]$. Namely,

(i) $$(\alpha \Rightarrow K\beta) \equiv M\mathbf{1}.$$

Now let $\circ[P*[\gamma Q]]$ be a program in the normal form equivalent to M [6], § 8. We remind that P and Q are loop-free programs. In turn, the convergence of the program $\circ[P*[\gamma Q]]$ is equivalent to the correctness of the program L: $*[\gamma \vee p \veebar [p\circ[P[p/\mathbf{0}]]Q]$ with respect to the formulas p and $\mathbf{1}$, where p is a propositional variable not occurring in the expressions P, Q and γ.

Let $H = (I, \hat{L})$ be the tree of the program L. We have for L the equivalence $\models_{\mathscr{R}} (p \Rightarrow L\mathbf{1})$ iff the modular structure $\hat{L}$ is correct in $\mathscr{R}$ with respect to the description A given by $A(e) = (p, \mathbf{1})$ and $A(i) = (\mathbf{1}, \mathbf{1})$ for $i \in I - \{e\}$.

The converse follows from (i) and the following simple observations:

$$\models_{\mathscr{R}} (K\mathbf{1} \wedge M\mathbf{1}) \text{ iff } \models_{\mathscr{R}} \veebar[pKM]\mathbf{1} \quad \text{and} \quad \alpha \equiv *[\neg\alpha[\;]]\mathbf{1}$$

where K, M are programs, p is a propositional variable occurring neither in K nor in M and α is an open formula. ∎

7. Properties of programs and second order logic

The main result of this section is included in Theorem 7.2.

First we must prove some auxilliary fact.

Let K be a program and let α, β be formulas. We shall assume that $\vec{x}$ and $\vec{z}$ are sequences of all different individual and propositional variables, respectively, occurring in K, α and β.

Let $\hat{H} = (I, \hat{K})$ be the tree of K and let $P = \{(P_i^1, P_i^2)\}_{i\in I}$ be the predicate description of the program K. The quantifiers binding all the predicates of the predicate description P will be written in the abbreviated form as $\exists P$ or $\forall P$.

7.1. *For every class of realizations* $\mathcal{R}$

$$\models_{\mathcal{R}} (K\mathbf{1} \wedge a \Rightarrow K\beta) \quad \text{iff} \quad \models_{\mathcal{R}} \exists P \forall \vec{x} \forall \vec{z} \, [\mathrm{VC} \wedge (P_e^1 \Leftrightarrow a) \wedge (P_e^2 \Leftrightarrow \beta)],$$

where VC *is the verification condition of the program* K *with respect to the predicate description* P.

Proof: Suppose that $\models_{\mathcal{R}} (K\mathbf{1} \wedge a \Rightarrow K\beta)$. By virtue of 5.1 (1), the modular structure $\hat{K}$ is partially correct with respect to some description $A = \{(a_i, b_i)\}_{i\in I}$ such that $a_e: a$ and $b_e: \beta$. Now we can define the meaning of the predicates from P in the following way: $P_i^1 \equiv_{\mathcal{R}'} a_i$ and $P_i^2 \equiv_{\mathcal{R}'} b_i$ for all i in I, where $\mathcal{R}'$ is the class of appropriate extensions of realizations from $\mathcal{R}$. Thus $\models_{\mathcal{R}'} (\mathrm{VC} \wedge (P_e^1 \Leftrightarrow a_e) \wedge (P_e^2 \Leftrightarrow b_e))$ and hence

$$\models_{\mathcal{R}} \exists P \forall \vec{x} \forall \vec{z} \, (\mathrm{VC} \wedge (P_e^1 \Leftrightarrow a) \wedge (P_e^2 \Leftrightarrow \beta)).$$

The converse results immediately from Theorem 2.1. ∎

By F_{II} we shall understand the set of all formulas of the second order predicate calculus, i.e., F_{II} is the least set containing F_{I} and closed under propositional connectives and the following rules:

if a is in F_{II} and x is an individual variable then $(\exists x a)$ and $(\forall x a)$ are in F;

if a is in F_{II} and ϱ is n-argument predicate in Ψ_n then $(\exists \varrho a)$ and $(\forall \varrho a)$ belong to F_{II}.

We say that a formula a in F is F_{II}-*existential* (F_{II}-*universal*) provided there exists a formula γ in F_{I} and a sequence of predicates $\vec{u}$ such that $a \equiv (\exists \vec{u} \gamma)$ $(a \equiv (\forall \vec{u} \gamma))$.

In the sequel K, M will denote any programs and a, β will denote any formulas from F_{I}, fixed from now on.

7.2. (1) *The following formulas are* F_{II}-*existential*:

a. $K\mathbf{1} \wedge a \Rightarrow K\beta$
b. $aK \Rightarrow \beta$
} *formulas of partial correctness.*

(2) *The following formulas are* F_{II}-*universal*:

a. $a \Rightarrow K\beta$	*formula of correctness,*
b. $K\mathbf{1}$	*halting formula,*
c. $K\mathbf{1} \wedge M\mathbf{1} \wedge (Ka \Leftrightarrow M\beta)$	*formula of equivalence of total programs,*
d. $\mathbf{1}K$	*formula of counter-domain.*

Proof: We can assume that the sets of free and bounded variables occurring in α and β are disjoint.

Let $\vec{y} = (y_1, \ldots, y_n, y_{n+1}, \ldots, y_m)$ be the sequence of all different variables occurring in the expressions K, α, β and such that $y_1, \ldots, y_n$ are individual variables and $y_{n+1}, \ldots, y_m$ are propositional variables.

Let $\vec{t} = (t_1, \ldots, t_n, t_{n+1}, \ldots, t_m)$ be a sequence of different variables not occurring in K, α, β and such that $t_1, \ldots, t_n$ are individual variables and $t_{n+1}, \ldots, t_m$ are propositional variables.

Let $\vec{c} = (c_1, \ldots, c_n, c_{n+1}, \ldots, c_m)$ be a sequence of different constants not occurring in K, α, β and such that $c_1, \ldots, c_n$ are 0-argument functors and $c_{n+1}, \ldots, c_m$ are 0-argument predicates.

Let K', α' and β' be the expressions resulting from K, α, β, respectively, by the simultaneous replacement of all occurrences of the variables from $\vec{y}$ by the variables from $\vec{t}$. Moreover, K'': $\circ[[\vec{t}/\vec{c}]K']$ and let α'' and β'' be the formulas obtained from α' and β' respectively, by the simultaneous replacement of all free occurrences of the variables from $\vec{t}$ by the constants from $\vec{c}$, respectively.

Let $P = \{(P_i^1, P_i^2)\}_{i \in I}$ be the predicate description of the program K'' and let VC be the verification condition of the program K'' with respect to the description P. Let VC$'$ be the formula resulting from VC by the simultaneous replacement of the constants from $\vec{c}$ by the variables from $\vec{y}$, respectively.

The following two lemmas can be regarded as the analogues of the results of Manna [16].

7.3. $\models \big((K\mathbf{1} \wedge \alpha \Rightarrow K\beta) \Leftrightarrow \exists P \forall \vec{t}\,(\mathrm{VC}' \wedge (P_e^1 \Leftrightarrow \alpha) \wedge (P_e^2 \Leftrightarrow \beta'))\big)$.

Proof: Applying 7.1 to the program K and to formulas α'' and β' we get that for every realization R

$$(7.3.1)\quad \models_R (K''\mathbf{1} \wedge \alpha'' \Rightarrow K''\beta') \quad \text{iff} \quad \models_R \exists P \forall \vec{t}\,(\mathrm{VC} \wedge (P_e^1 \Leftrightarrow \alpha'') \wedge (P_e^2 \Leftrightarrow \beta')).$$

Observe that the formulas in 7.3.1 have constant values in every realization R. Hence

$$\models \big((K''\mathbf{1} \wedge \alpha'' \Rightarrow K''\beta') \Leftrightarrow \exists P \forall \vec{t}\,(\mathrm{VC} \wedge (P_e^1 \Leftrightarrow \alpha'') \wedge (P_e^2 \Leftrightarrow \beta')\big).$$

Since the above formula is a tautology, we can replace all the occurrences of the constants from $\vec{c}$ by the variables from $\vec{y}$ which do not occur in the formulas in question. We get that

$$\models \big((\circ[[\vec{t}/\vec{y}]K']\mathbf{1} \wedge \alpha \Rightarrow [[t/y]K']\beta') \Leftrightarrow \exists P \forall \vec{t}(\mathrm{VC}' \wedge (P_e^1 \Leftrightarrow \alpha) \wedge (P_e^2 \Leftrightarrow \beta')\big).$$

Hence, because of $\circ[[\vec{t}/\vec{y}]K']\mathbf{1} \equiv K\mathbf{1}$ and $\circ[[\vec{t}/\vec{y}]K']\beta' \equiv K\beta$, we finally obtain the fact 7.3. ■

7.4. $\models \Big((\alpha \Rightarrow K\beta) \Leftrightarrow \forall P \exists \vec{t}\big(\alpha \Rightarrow \neg(\mathrm{VC}' \wedge P_e^1 \wedge (P_e^2 \Leftrightarrow \neg\beta'))\big)\Big)$.

Proof: First we apply 7.3 to the program K and formulas $\mathbf{1}$ and $\neg\beta$. We get that

$$\models \Big((K\mathbf{1} \Rightarrow K\neg\beta) \Leftrightarrow \exists P \forall \vec{t}\big(\mathrm{VC}' \wedge P_e^1 \wedge (P_e^2 \Leftrightarrow \neg\beta')\big)\Big).$$

Hence, by negating both sides of the above equivalence and introducing the formula α, we finally obtain the fact 7.4. ■

Now let us consider the program M: $\circ\big[K' * [\vec{t} \neq \vec{c}\,[\ ]]\big]$. Let $\tilde{P}$ be the predicate description of M. Let $\widetilde{\mathrm{VC}}$ be the verification condition of the program M with respect to the description $\tilde{P}$. Let $\widetilde{\mathrm{VC}}'$ be the formula resulting from $\widetilde{\mathrm{VC}}$ by the simultaneous replacement of all occurrences of the constants from $\vec{c}$ by the variables from $\vec{y}$, respectively. Then the following lemma holds.

7.5. $\models \Big((\alpha K \Rightarrow \beta) \Leftrightarrow \exists \tilde{P} \forall \vec{t}\big(\widetilde{\mathrm{VC}}' \wedge (\tilde{P}_e^1 \Leftrightarrow \alpha') \wedge (\tilde{P}_e^2 \Leftrightarrow \beta)\big)\Big)$.

Proof: Applying Lemma 7.1 and I, 3.1(1), we get that for every realization R

$$(7.5.1) \qquad \models_R (\alpha' M \Rightarrow \beta'') \quad \text{iff} \quad \models_R \exists P \forall \vec{t}\big(\widetilde{\mathrm{VC}} \wedge (P_e^1 \Leftrightarrow \alpha') \wedge (P_e^2 \Leftrightarrow \beta'')\big).$$

Observe that the formulas in (7.5.1) have constant values in every realization R. Hence

$$\models \Big((\alpha' M \Rightarrow \beta'') \Leftrightarrow \exists \tilde{P} \forall \vec{t}\big(\widetilde{\mathrm{VC}} \wedge (P_e^1 \Leftrightarrow \alpha') \wedge (P_e^2 \Leftrightarrow \beta'')\big)\Big).$$

Since the above formula is a tautology, we can replace all the occurrences of the constants from $\vec{c}$ by the variables from $\vec{y}$ which do not occur in the formula in question. We get

$$\models \Big(\big(\alpha' K' * [\vec{t} \neq \vec{y}\,[\ ]] \Rightarrow \beta\big) \Leftrightarrow \exists P \forall \vec{t}\big(\widetilde{\mathrm{VC}}' \wedge (\tilde{P}_e^1 \Leftrightarrow \alpha') \wedge (\tilde{P}_e^2 \Leftrightarrow \beta)\big)\Big).$$

Hence, because of $\alpha' K' * [\vec{t} \neq \vec{y}[\ ]] \equiv \alpha K$, we obtain the fact 7.5. ■

Assertions (1)a, (2)a, (2)b, (1)b and (2)d of 7.2 follow from Lemmas 7.3, 7.4 and 7.5. Assertion (2)c can be reduced to assertion (2)a. Namely, let $\vec{y}$ be a sequence of all variables occuring in the expressions K, M, α and β and let $\vec{x}$ be a copy of $\vec{y}$. Then

$$\big(K\mathbf{1} \wedge M\mathbf{1} \wedge (K\alpha \Leftrightarrow M\beta)\big) \equiv \big(\mathbf{1} \Rightarrow \circ[[\vec{x}/\vec{y}]\circ[KM(\vec{x})]]\,(\alpha \Leftrightarrow \beta(\vec{x}))\big).$$

On account of the deduction theorem and of the completeness theorem, Theorem 7.2 implies the following important fact.

7.6. *Let A be a finite set of formulas of the first order predicate calculus and let $\mathcal{T} = \{\mathcal{L}, C, A\}$ be the algorithmic theory based on the set A of axioms. Let a be an F_{II}-universal formula.*

Then a is a theorem of $\mathcal{T}$ iff a certain effectively constructible formula of the first order predicate calculus is unsatisfiable.

This theorem reduces the examination of an appropriate property of programs to an automatic theorem proving based on the Herbrand theorem (compare 7.6 with the partial Herbrand theorem in [21]).

References

[1] de Bakker, J. W., de Roever, W. P., *Calculus for recursive program schemes*, Stichting Mathematisch Centrum, MR 131/72, Amsterdam 1972.

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

[3] — *An axiomatic approach to the theory of data structures*, Bull. Acad. Polon. Sci., Sér. Sci. Math. Astronom. Phys. 23 (1975), 315–323.

[4] — *Extended algorithmic logic and properties of programs*, ibidem, 325–330.

[5] — *Modular properties of programs*, ibidem, 331–337.

[6] — *Investigations of properties of programs by means of the extended algorithmic logic* (part I of this paper), Fund. Inf. 1, 93–119.

[7] Banachowski, L., Szczepańska, D., Wassersztrum, W., Żurek, J., *Automatic verification of program correctness*, Problemy węzłowe, Warsaw University 1974 (in Polish).

[8] Floyd, R. W., *Assigning meanings to programs*, Proc. of Symposia in Applied Mathematics 19, AMS, 1967.

[9] Góraj, A., Mirkowska, M., Paluszkiewicz, A., *On the notion of description of program*, Bull. Acad. Polon. Sci., Sér. Sci. Math. Astronom. Phys. 18 (1970), 499–506.

[10] Hoare, C. A. R., *An axiomatic basis of computer programming*, Comm. ACM 12 (1969), 576–583.

[11] — *Procedures and parameters: an axiomatic approach*, Symp. on Semantics of Algorithmic Languages, Springer-Verlag, 1971.

[12] King, J. C., *A program verifier*, Proc. IFIP, 1971, 234–249.

[13] King, J. C., Floyd, R. W., *An interpretation oriented theorem prover over integers*, J. Comput. System. Sci. 6 (1972), 305–323.

[14] Kreczmar, A., *Effectivity problems of algorithmic logic*, doctoral dissertation, Warsaw University 1973 (in Polish).

[15] — *Effectivity problems of algorithmic logic*, 2nd Colloquium on Automata, Languages and Programming, Springer-Verlag, 1974.

[16] Manna, Z., *The correctness of programs*, J. Comput. System Sci. 3 (1969), 119–127.

[17] Manna, Z., Pnueli, A., *Formalization of properties of functional programs*, J. Assoc. Comput. Mach. 17 (1970), 555–569.

[18] —, — *Axiomatic approach to total correctness of programs*, Acta Informaticae 3 (1974), 243–263.

[19] Mazurkiewicz, A., *Proving properties of processes*, Algorytmy 11 (1974), 5–21.

[20] Mirkowska, M., *On formalized systems of algorithmic logic*, Bull. Acad. Polon. Sci., Sér. Sci. Math. Astronom. Phys. 19 (1971), 421–428.
[21] — *Herbrand theorem in algorithmic logic*, ibidem, 22 (1974), 539–543.
[22] — *Algorithmic logic and its applications in the theory of programs*, doctoral dissertation, Warsaw University 1972 (in Polish).
[23] Rasiowa, H., Sikorski, R., *Mathematics of metamathematics*, PWN, Warszawa 1968.
[24] Salwicki, A., *Formalized algorithmic languages*, Bull. Acad. Polon. Sci., Sér. Sci. Math. Astronom. Phys. 18 (1970), 227–232.
[25] — *On the equivalence of FS-expressions and programs*, ibidem, 275–278.
[26] — *Programmability and recursiveness (an application of algorithmic logic to procedures)*, to appear.
[27] *Efficient production of large programs*, Proc. of Intern. Workshop Jablonna 1970, CC PAS, PWN, Warszawa 1971.