Prace Naukowe Wyższej Szkoły Pedagogicznej w Częstochowie
Matematyka II, Częstochowa 1996

On a three-valued internal logic of programs

Andrzej Zbrzezny

Introduction. The only logical connectives occuring in programming languages are and, or and not. They belong to the so-called internal logic. This logic is 3-valued because during the execution of a program some variables can be undefined, and therefore some boolean expressions can be undefined, too. Thus our logical values are true, false and undefined. Moreover, our interpretation of logical connectives is due to a certain implementation principle, namely the one which consists in parallel evaluation of components of boolean expresions. (of[1]).

In this paper we consider the propositional fragment of the internal logic of programs.

1. Syntax and semantics of the internal logic of programs.

We denote by N the set of all positive integers.

Let At=$\{p_i : i \in N\}$ be the denumerable set of propositional variables. We write p, q and r instead of p_1, p_2 *and* p_3 respectively. As metavariables for elements of At we use letters a and b. (Always when we say that some letters are metavariables for elements of a given set we mean that these variables vary through this set and that we allow their indexed or primed variants of these letters just as well as variables over the some set. However, no letter is used as a variable for a given set unless specifically indicated in this way. Only a and b (and a_1, b_1, a_2, b_2 ... , A', b', a, b, ..., etc.) range over At and not, e.g., c, d or x (compare with a notion of the syntactic variable in [3]).

Definition 1.1.

The set S of propositional formulae of internal logic is the smallest set for which three following conditions hold:

(a) $At \subset S$

(b) if $A \in S$ then $(\sim A) \in S$

(c) if A, $B \in S$ then $(A \wedge B) \in S$ and $(A \vee B) \in S$

End 1.1

We usually do not write the most outside paranthesis. As metavariables for elements of S we use letters A, B, C and D. Symbols $\wedge$, $\vee$ and $\sim$ correspond to boolean operators of programming languages: and, or and not respectively.

The ordered 4-tuple $\underline{S} =< S, \wedge, \vee, \sim >$ is said to be the propositional language of internal logic of programs. As the interpretation of $\underline{S}$ we take a logical matrix defined as follows:

$$\underline{M_3} =< M_3, M_3^+, f_1, f_2, f_3 > \text{ where}$$
$$M_3 = \{ 0, 1, 2 \}, \; M_3^+ = \{ 2 \}$$
$$f_1 : M_3 \times M_3 \to M_3 \text{ and}$$
$$(\forall x \in M_3)(\forall y \in M_3)[f_1(x,y)=\min(x,y)],$$
$$f_2 : M_3 \times M_3 \to M_3 \text{ and}$$
$$(\forall x \in M_3)(\forall y \in M_3)[f_2(x,y)=\max(x,y)],$$
$$f_3 : M_3 \to M_3 \text{ and}$$
$$(\forall x \in M_3)[f_3(x)=2\text{-}x].$$

Intuitivitely, 0 stands for false, 1 for undefined, 2 for true. The general definition of logical matrix and also many other definitions used in this paper are to be found in [2].

Let $\underline{M}$ be an arbitrary logical matrix, Functions from At into M are called valuations. Every valuations can be extended to the unique homomorphism $h^e : S \to M$.

In case of $\underline{M_3}$ homomorphism is given by the following equations:

$$(\forall a)[h^e(a) = e(a)]$$
$$(\forall A)(\forall B)[h^e(A \wedge B) = \min (h^e(A), h^e(B))]$$
$$(\forall A)(\forall B)[h^e(A \vee B) = \max (h^e(A), h^e(B))]$$
$$(\forall A)[h^e(\sim A) = 2 - h^e(A)].$$

For arbitrary matrix $\underline{M}$ one can define the content of $\underline{M}$ as the set:

$$E(\underline{M}) = \{A \in S : (\forall e) h^e(A) \in M^+\}.$$

If $A \in E(\underline{M})$ then we say that A is true in the matrix $\underline{M}$ or that A is a tautology of the matrix $\underline{M}$.

Unfortunately, $E(\underline{M_3})$ is the empty set. How can we improve this situation? Well, we could add to our language new connectives, e.g. $\to$ which might be interpreted as a function f_4 such that:

$$(\forall x \in M_3)(\forall y \in M_3)[f_4(x,y) = \min(2,2 - x + y)]$$

In this case the set of all tautologies wouldn't be empty. But let's consider such formulae A and B and a valuation e for which $h^e(A) = h^e(B) = 1$ and observe that $f_4(1,10)$=2. This would mean that the logical value of the boolean expression A $\rightarrow$ B is true while both A and B are undefined. This isn't satisfactory from the programming point of view. On the other hand we can add to our language one-argument connective D which might be interpreted as a function f_5 defined as follows:

$$f_5(0) = f_5(2) = 2,$$
$$f_5(1) = 1.$$

Also in this case the set of all tautologies wouldn't be empty. Intuitively, we can read D_p as "p is defined". However, f_5 is not implementable in any computer! so this isn't satisfactory, too.

This is why we will not deal with the internal logic. Our purpose is to give a synthetic characterization of all pairs of equivalent formulae of the internal logic. First, we define the relation # on S as follows:

$$A\#B \text{ iff } (\forall e)[h^e(A) \leq h^e(B)].$$

This relations is a quasi-order on S, and the relation $= =^{-1}$ is an equivalence relation such that for every A and B $< A, B > \in \# \cap \#^{-1}$ iff A and B are logically equivalent. (A and B are logically equivalent iff $(\forall e)[h^e(A) = h^e(B)]$.) Next, we use the symbol $\vdash$ as a synthetic counterpart of the relation # and we define the set Form of all ordered pairs of propositional formulae as the smallest set for which the following condition holds:

$$\text{if } A \in S \text{ and } B \in S \text{ then } A \vdash B \in Form.$$

As a metavariable for elements of Form we use a letter F. A formula $A \vdash B$ is said to be valid iff $< A, B > \in \#$.

2. Proof system.

A proof system is an ordered 3-tuple $< Form, Ax, R >$ such that Form is the set of all formulae of given language, Ax is a recursive set of formulae called axioms (i.e. Ax is a subset of Form and there is an algorithm (e.g. Turing machine) that takes an arbitrary formula as input and determines

whether or not it is an axiom), and R is the finite set of relations called rules of inference. Each rule of inference is a relation with domain Fin(Form) and codomain Form and it is of the form:

$$\frac{F_1, F_2, ..., F_n}{F}$$

which indicates that from the formulas given above the line we can infer the one given below the line. $F_1, F_2, ..., F_n$ are said to be the premises and F is said to be the conclusion of the rule. (For a given set X we denote the set of all finite subset of X by Fin(X).) A formula F is said to be provable in such a system (in other words F is a theorem of such a system) if there exists a finite sequence of formulae such that F is the last formula in the sequence and each formula in the sequence is either an axiom or it is derived from previous formulae in the sequence by one of the rules of inference. Such a sequence of formulae is called a proof of F.

In case of our logic let Form be the set defined in the previous section. We must define sets Ax and R.

The only elements of Ax are:

(ax1) $\mathrm{p} \vdash \sim\sim p$
(ax2) $\sim\sim p \vdash \mathrm{p}$
(ax3) $p \wedge q \vdash \mathrm{p}$
(ax4) $\mathrm{p} \vdash q \vee p$
(ax5) $(p \vee q) \wedge r \vdash (p \wedge r) \vee (q \wedge r)$
(ax6) $\sim (p \vee q) \vdash \sim p \wedge \sim q$
(ax7) $\sim p \wedge \sim q \vdash \sim (p \vee q)$
(ax8) $p \wedge \sim p \vdash q \vee \sim q$

The only elements of R are:

(r1)
$$\frac{F}{F(a/A)}$$

(r2)
$$\frac{F, A \vdash B, B \vdash A}{F(A//B)}$$

(r3)
$$\frac{A \vdash B}{A \vdash B \vee C}$$

(r4)

$$\frac{A \vdash C, B \vdash C}{A \vee B \vdash C}$$

In (r1) F(a/A) results from F by simultaneous replacing all occurrences of a in F by A. In (r2) F(A//B) results from F by simultaneous replacing some (may be all) occurrences of A in F by B.

We denote our proof system by G_3, and all provable formulae by T_3. We list below some theorems of G_3 and derived rules of inference needed in the next section. Each derived rule of inference has an effective proof in the sense that we can effectively show how to replace any derived rule of inference whenever it is used in a proof by an appropriate sequence of formulae using only the "primitive" rules of inference. Moreover, in any proof of the given theorem of G_3 (just as in any proof of the given theorem of an arbitrary proof system) one can use theorems proved before because in such not quite formal proof one can replace all already proved theorems by their proofs, and obtain the formal proof of the given theorem.

(th1)

$$p \vdash p$$

proof:

(1)	$p \vdash \sim\sim p$	(ax1)
(2)	$\sim\sim p \vdash p$	(ax2)
(3)	$p \vdash p$	from (1), (2), (2) by (r2)

(th2)

$$p \vdash p \vee q$$

proof:

(1)	$p \vdash p$	(th1)
(2)	$p \vdash p \vee q$	from (1) by (r3)

(th3)

$$p \vee q \vdash q \vee p$$

proof:

(1)	$p \vdash q \vee p$	(ax4)
(2)	$p \vdash p \vee q$	(th2)
(3)	$q \vdash q \vee p$	from (2) by (r1)
(4)	$p \vee q \vdash q \vee p$	from (1), (3) by (r4)

(r5)

$$\frac{A \vdash B, B \vdash C}{A \vdash C}$$

proof:

(1)	$A \vdash B$	given
(2)	$B \vdash C$	given
(3)	$C \vdash C$	from (th1) by (r1)
(4)	$B \vee C \vdash C$	from (2), (3) by (r4)
(5)	$C \vdash B \vee C$	from (ax4) by (r3)
(6)	$A \vdash B \vee C$	from (1) by (r3)
(7)	$A \vdash C$	from (6), (4), (5) by (r2)

(r6)

$$\frac{A \vdash B}{B \vdash A}$$

proof:

(1)	$A \vdash B$	given
(2)	$B \vdash A \vee B$	from (ax4) by (r1)
(3)	$B \vdash B$	from (th1) by (r1)
(4)	$A \vee B \vdash B$	from (1), (3) by (r4)
(5)	$\sim (A \vee B) \vdash \sim (A \vee B)$	from (th1) by (r1)
(6)	$\sim B \vdash \sim (A \vee B)$	from (5), (2), (4) by (r2)
(7)	$\sim (A \vee B) \vdash \sim A \wedge \sim B$	from (ax6) by (r1)
(8)	$\sim B \vdash \sim A \wedge \sim B$	from (6), (7) by (r5)
(9)	$\sim A \wedge \sim B \vdash \sim A$	from (ax3) by (r1)
(10)	$\sim B \vdash \sim A$	from (8), (9) by (r5)

(th4) $\sim (p \wedge q) \vdash \sim p \vee \sim q$

(th5) $\sim p \vee \sim q \vdash \sim (p \wedge q)$

(th6) $p \wedge q \vdash q \wedge p$

(th7) $p \wedge q \vdash q$

(th8) $(p \wedge r) \vee (q \wedge r) \vdash (p \vee q) \wedge r$

(th9) $(p \vee r) \wedge (q \vee r) \vdash (p \wedge q) \vee r$

(th10) $(p \wedge q) \vee r \vdash (p \vee r) \wedge (q \vee r)$

(r7)

$$\frac{A \vdash C}{A \wedge B \vdash C}$$

(r8)
$$\frac{A \vdash B, A \vdash C}{A \vdash B \wedge C}$$

(th11) $\mathrm{p} \vdash p \wedge q$
(th12) $p \vee p \vdash \mathrm{p}$
(th13) $p \vee (q \vee r) \vdash (p \vee q) \vee r$
(th14) $(p \vee q) \vee r \vdash p \vee (q \vee r)$
(th15) $(p \wedge (q \wedge r)) \vdash (p \wedge q) \wedge r$
(th16) $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$

Ideas of proofs of theorems (th4), ..., (th10), rules (r7), (r8), and theorems (th11), ..., (th16) one can find in [4].

3. Soundness and completeneas of G_3.

In this section we show that the set of all theorem of G_3 is equal to the set of all valid formulae of Form. This result divides into two parts: soundness theorem and completeness theorem. The method of proving the latter was inspired by idea given in [4].

Theorem 3.1. (soundness).

For every formula F Form the following holds: if $F \in T_3$ then F is valid.

Proof.

We must show that each formally provable F is valid. Let F_1, ..., F_n be a formal proof for F. We prove that, for each i=1, ..., n, F_i is valid (thus F is valid). The proof proceeds by showing that F_1 is valid and, for each $1 < i \leq n$, if F_1, ..., F_{i-1} are valid then F_i is valid.

(i) i=1. Clearly, F_1 has to be an axiom and, therefore, we must show that each axiom is valid. But this is straightforward, and we omit the proof.

(ii) Assuming F_1, ..., F_{i-1} are valid, we show that F_i is valid. Either F_i is an axiom, in which case its validity follows as in part (i), or there exists F_{j_1}, ..., $F_{j_{n_i}}$, $1 \leq j_k < i$ for k=1, 2, ..., n_i such that

$$\frac{F_{j_1}, \ldots, F_{j_{n_i}}}{F_i}$$

is an inference rule. By the induction hypothesis, F_{j_1}, ..., $F_{j_{n_i}}$ are valid. But it is easy to show that validity of its conclusion. So F_i is valid.

End 3.1.

In order to prove completeness of G_3 we must recall some notions.

Definition 3.2.

$$\overline{At} \stackrel{d}{=} \{A \in S : (\exists a)[A \equiv\sim a]\}; EC \stackrel{d}{=} At \cup \overline{At}$$

Elements of the set EC are said to be elementary components. As metavariables for elements of EC we use letters s and t.

End 3.2.

In the above definition we have used symbol " $\equiv$". It denotes syntactical identity of syntactic constructions. Two constructions are syntacticaly identical whenever they consist of the same sequence of symbols. For example, $p \wedge q \equiv p \wedge q$, $a \vee b \equiv a \vee b$, but $p \wedge q \equiv q \wedge p$ and $a \vee b \equiv\sim a$. Observe that $a \equiv b$ may or may not be satisfied, depending on whether a and b are not themselves (sequences of symbols, but are variables ranging over a set of symbols).

Definition 3.3.

$$A_1 \wedge ... \wedge A_n \stackrel{d}{=} \begin{cases} A_1, & if\ n = 1 \\ A_1 \wedge A_2, & if\ n = 2 \\ (A_1 \wedge ... \wedge A_{n-1}) \wedge A_n, & if\ n > 2 \end{cases}$$

An expresion $A_1 \wedge ... \wedge A_n$ is said to be a generalized conjunction. If $A_1, ..., A_n$ are elementary components then we say that $A_1 \wedge ... \wedge A_n$ is an elementary conjunction.

$$A_1 \vee ... \vee A_n \stackrel{d}{=} \begin{cases} A_1, & if\ n = 1 \\ A_1 \vee A_2, & if\ n = 2 \\ (A_1 \vee ... \vee A_{n-1}) \vee A_n, & if\ n > 2 \end{cases}$$

An expresion $A_1 \vee ... \vee A_n$ is said to be a generalized disjunction. If $A_1, ..., A_n$ are elementary components then we say that $A_1 \vee ... \vee A_n$ is an elementary disjunction.

End 3.3.

Definition 3.4.

A propositional formula A is said to be in a conjunctive normal form (symbolically $A \in CNF$) if there exist elementary disjunctions $A_1, ..., A_n$ such that $A \equiv A_1 \wedge ... \wedge A_n$.

End 3.4.

If X is an arbitrary set of propositional formulae ($X \subset S$) then by At(X) we denote the set of all propositional variables occurring in formulae from X. We write At(A) instead of At({A}).

Theorem 3.5.

For every propositional formula A there exists a propositional formula A' such that $A' \in CNF$, At(A')=At(A), $A \vdash A' \in T_3$ and $A' \vdash A \in T_3$.

End 3.5.

Theorem 3.6.

For every propositional formula A there exists a propositional formula A' such that $A' \in DNF, At(A), A \vdash A' \in T_3$, and $A' \vdash A \in T_3$.

End 3.6.

Both metatheorem 3.5. and metatheorem 3.6. can be proved by induction on the structure of a given formula (of[2]) and they are theorems of the metatheory of our proof system G_3 because all theorems needed to construct a proper normal form are theorems of G_3.

Definition 3.7.

A propositional formula A is said to be satisfiable iff there exists e: $At \rightarrow M_3$ such that $h^e(A) = 2$.

A propositional formula A is said to be refutable iff there exist e: $At \rightarrow M_3$ such that $h^e(A) = 0$.

End 3.7.

Lemma 3.8.

(a) Let A be an elementary conjunction, $A \equiv s_1 \wedge ... \wedge s_m$. Then A is satisfible iff

$$(\forall a)[\exists 1 \leq j \leq m)(s_j \equiv a) \Rightarrow (\forall 1 \leq k \leq n)(s_k \equiv \sim a)].$$

(b) Let B be an elementary disjunction, $B \equiv t_1 \vee \ldots \vee t_n$. Then B is refutable iff

$$(\forall b)[\exists 1 \leq j \leq n)(t_j \equiv b) \Rightarrow (\forall 1 \leq k \leq n)(t_k \equiv \sim b)].$$

End 3.8.

Proof of the above lemma is obvious and therefore we omit it. In the next three theorems the following condition:

$(\exists m)(\exists A_m)\ldots(\exists A_m)(\exists n)(\exists B_1)\ldots(\exists B_n)[A \in DNF$ and $B \in CNF$ and $A \equiv A_1 \vee \ldots \vee A_m$ and $B \equiv B_1 \wedge \ldots \wedge B_n$ and
$(\forall 1 \leq i \leq m)(\forall 1 \leq j \leq n)(A_i$ and B_j have a common component or (A_i is not satisfiable and B_j is not refutable))] is denoted by V(A,B).

Theorem 3.9.

For every two propositional formulae A and B such that $A \in DNF$ And $B \in CNF$: if $A \models B$ holds then V(A,B) holds.

Proof.

Let us fix A and B such that $A \in DNF$, $A \equiv A_1 \vee \ldots \vee A_m$, $B \in CNF$, $B \equiv B_1 \wedge \ldots \wedge B_n$ and

(1) $\quad A \models B$

From (1) we obtain by the definition of the relation $\models$

(2) $\quad (\forall e)[h^e(A) \leq h^e(B)]$

Now, from (2) we have by the definition of the homomorphism h

(3) $\quad (\forall e)[max\{h^e(A_1), \ldots, h^e(A_n)\} \leq min\{h^e(B_1), \ldots, h^e(B_n)\}]$

But by the definitions of maximum and minimum

(4) $\quad (\forall 1 \leq i \leq m)(\forall e)[h^e(A_i) \leq max\{h^e(A_1), \ldots, h^e(A_m)\}]$

and

(5) $(\forall 1 \leq j \leq m)(\forall e)[min\{h^e(B_1), ..., h^e(B_n)\} \leq h^e(B_j)]$

hold. So (3), (4), and (5) imply

(6) $(\forall 1 \leq i \leq m)(\forall 1 \leq j \leq n)(\forall e)[h^e(A_i) \leq h^e(B_j)]$.

Now let's suppose that V(A, B) does not hold. Thus there exist i and j, $1 \leq i \leq m, 1 \leq j \leq n$, such that

(7) A_i and B_j haven't a common elementary component and (A_i is satisfiable or B_j is refutable). We must consider two cases:

(7.1) A_i and B_j haven't a common elementary component and A_i is satisfiable and

(7.2) A_i and B_j haven't a common elementary component and B_j is refutable.

From (7.1) we obtain

(7.1.1) A_i and B_j haven't a common elementary component and there exist we such that $h^e(A_i)$=2.
Now let's consider a valuation e_1 defined as follows:

$$e_1(a) = \begin{cases} e(a) & iff\ either\ a\ or\ \sim a\ is\ an\ elementary\ component\ of\ A_i \\ 1 & otherwise \end{cases}$$

From this definition and from (7.1.1) we obtain

(7.1.3) $h^{e_1}(A_i) = h^e(A_i) = 2\ and\ h^{e_1}(B_j) = 1$.

But (7.1.2) and (6) are contradictory.
Analogously, from (7.2) we obtain

(7.2.1) A_i and B_j haven't a common elementary component and there exist we such that $h^e(B_j) = 0$ and now let's a consider a valuation e_2 defined as follows:

$$e_2(a) = \begin{cases} e(a) & iff\ either\ a\ or\ \sim a\ is\ an\ elementary\ component\ of\ B_j \\ 1 & otherwise \end{cases}$$

From this definition and from (7.2.1) we obtain

(7.2.2) $h^{e_2}(B_j) = h^e(B_j) = 0$ *and* $h^{e_2}(A_i) = 1$.

But (7.2.2) and (6) are contradictory.

Finally, in both cases (7.1) and (7.2) we obtained a contradiction. Thus V(A,B) holds.

End 3.9.

Lemma 3.10.

For every A,B,$C_1, ..., C_n, D_1, ...D_n \in S$ the following condition hold:
(a) if $A \vdash B \in T_3$, then $A \wedge C_1 \wedge ... \wedge C_n \vdash B \in T_3$
(b) If $A \vdash B \in T_3$, then $A \vdash B \vee D_1 \vee ... \vee D_n \in T_3$.

End 3.10.

Lemma 3.10 can be proved by induction on n. The proof is straightforward and therefore we omit it.

Lemma 3.11.

Let C be an elementary conjunction, $C \equiv s_1 \wedge .. \wedge s_m$, and D be an elementary disjunction, $D \equiv t_1 \vee ... \vee t_n$. Then the following holds:

(a) If C is not satisfiable, then there exist $a \in At$ such that a and $\sim a$ are elementary components of C and $C \vdash a \wedge \sim a \in T_3$.
(b) If D is not refutable, then there exist $b \in At$ such that b and $\sim b$ are elementary components of D and $b \vee \sim b \vdash D \in T_3$.

End 3.11.

The idea of the proof of the above lemma is quite similar to the idea applied in part (i) of the proof of the next theorem. Thus we omit this proof.

Theorem 3.12.

For every two propositional formulae A and B: if V (A,B) holds, then $A \vdash B \in T_3$.

Proof.

If V (A,B) holds, then $A \in DNF$, $A \equiv A_1 \vee ... \vee A_m$, $B \in CNF$, $B \equiv B_1 \wedge ... \wedge B_n$, and

(*) $A_j \equiv s_i, 1 \wedge ... \wedge s_{i,mi}$, for i = 1, ... , m

and

(**) $B_j \equiv t_{j,1} \vee ... \vee t_{j,nj}$, for j = 1, ... , n.

Now let's fix i and j. We must consider two cases:
(i) A_i and B_j have a common elementary component i.e. there exist k,l such
that $1 \leq k \leq m_i$, $1 \leq 1 \leq n_j$, and $s_{i,k} \equiv t_{j,l}$. From this and from (th 1) by
(r1) we obtain

(1) $s_{i,k} \vdash t_{j,l}$

Also from (ax3) by (r1) we have

(2) $(s_{i,1} \wedge ... \wedge s_{i,k-1}) \wedge s_{i,k} \vdash s_{i.k}$

and from (2) by the definition of generalized conjunction we have

(3) $s_{i,1} \wedge ... \wedge s_{i,k} \vdash s_{i,k}$.

Now applaying lemma 3.10 to (3) we obtain

(4) $s_{i,1} \wedge ... \wedge s_{i,k} \wedge s_{i,k+1} \wedge ... \wedge s_{i,mi} \vdash s_{i,k}$

and applying (*) to (4) we obtain

(5) $A_i \vdash s_{i,k}$.

On the other hand, from (ax4) by (r1) and the definition of generalized disjunction we have

(6) $t_{j,1} \vdash t_{j,1} \vee ... \vee t_{j,1}$

and applying lemma 3.10 to (6) we obtain

(7) $t_{j,1} \vdash t_{j,1} \vee ... \vee t_{j,1} \vee t_{j,1+1} \vee ... \vee t_{j,nj}$.

Now applying (**) to (7) we obtain

(8) $t_{j,1} \vdash B_j$.

Finally, (1), (5) and (8) imply

(9) $A_i \vdash B_j$

by (r5).

(ii) A_i is not satisfiable and B_j is not refutable. By lemma 3.11 there exist a and b such that

(1) $A_i \vdash a \wedge \sim a$

and

(2) $b \vee \sim b \vdash B_j$.

But from (ax8) we obtain by (r1)

(3) $a \wedge \sim a \vdash b \vee \sim b$.

Finally, (1), (2) and (3) imply

(4) $A_i \vdash B_j$

by (r5).

In the both of cases we obtained that $A_i \vdash B_j \in T_3$. (in all this proof we wrote $C \vdash D$ instead of $C \vdash D \in T_3$ for the simplicity of the notation). Since i and j were arbitrary we have

(1) $(\forall 1 \leq i \leq m)(\forall 1 \leq j \leq n)[A_i \vdash B \in T_3]$.

Writing (1) explicitly we obtain

$$
\begin{array}{ll}
A_1 \vdash B_1 \in T_3 & \text{and} \\
\vdots & \\
A_1 \vdash B_n \in T_3 & \text{and} \\
\vdots & \\
A_m \vdash B_1 \in T_3 & \text{and} \\
\vdots & \\
A_m \vdash B_n \in T_3 & \text{and}
\end{array}
\qquad (2)
$$

From (2) we obtain

$$(3)\qquad \begin{array}{l} A_1 \vdash B_1 \wedge \ldots \wedge B_n \in T_3 \quad \text{and} \\ \vdots \\ A_m \vdash B_1 \wedge \ldots \wedge B_n \in T_3 \end{array}$$

by (r8) and from (3) we obtain

$$(4)\qquad A_1 \vee \ldots \vee A_m \vdash B_1 \wedge \ldots \wedge B_n \in T_3$$

by (r4). Finally, $A \vdash B \in T_3$.

End 3.12

Theorem 3.13 (completeness theorem).

For every propositional formulae A and B: if $A \models B$ holds, then $A \vdash B \in T_3$.

Proof.

Suppose that

$$(1)\qquad A \models B$$

holds. By theorems 3.5 and 3.6 there exist $A' \in DNF$ and $B' \in CNF$ such that

$$(2)\qquad A' \vdash A \in T_3,$$
$$(3)\qquad A \vdash A' \in T_3,$$
$$(4)\qquad B \vdash B' \in T_3,$$

and

$$(5)\qquad B' \vdash B \in T_3,$$

Now (2) and (4) imply

$$(6)\qquad A' \models A$$

and

$$(7)\qquad B \models B'$$

by the theorem 3.1. Since $\models$ is a transitive relation then we obtain from (1), (6), and (7) that

(8) $A' \models B'$.

But $A \in DNF$ and $B \in CNF$, so (8) implies

(9) V (A',B')

by theorem 3.9.

Now from (9) we obtain

(10) $A' \vdash B' \in T_3$

by theorem 3.12. Finally, (3), (5), and (10) imply

(11) $A \vdash B \in T_3$

by (r5).

End 3.13.

References

[1] A.BLIKLE: *Notes on the mathematical semantics of programing languages,* ICS PAS Reports, 445(1981), Warszaw.

[2] A.W.POGORZELSKI: *Klasyczny rachunek zadań,* PWN, Warszawa, 1975.

[3] J.SHOENFIELD: *Mathematical logic,* Addison-Wesley, Toronto, 1967 (Russian translation).

[4] E.A.SIDORENKO: *Logiceskoie sledovanie i uslovanie vyskazyvania,* Izdatelstvo Nauka, Moskva, 1983 (in Russian).

On a three-valued internal logic of programs

Andrzej Zbrzezny

Streszczenie

Celem pracy jest przedstawienie pewnej aksjomatyzacji zdaniowego fragmentu trójwartościowej wewnętrznej logiki programów. Wykazujemy, że wprowadzony system formalny jest niesprzeczny i pełny.

On a three-valued internal logic of programs

Abstract

The purpose of the work is a certain axiomatization of the propositional fragment of the three-valued internal logic of programs. We prove that the introduced formal system is sound and complete.