

УДК 519.766

## ФОРМАЛЬНОЕ ОПИСАНИЕ ЯЗЫКОВ ДЕЙСТВИЙ

М.К.Тимофеева

### В в е д е н и е

В [1] был введен класс формальных языков - языков действий, - предназначенных для управления сложными динамическими системами произвольной природы посредством текстов предметно и функционально ограниченного естественного языка (далее называемого входным языком). В данной публикации приводится возможный способ формальной реализации языков действий в рамках ограничений, сформулированных в [1].

Сущность языков действий состоит в том, что они являются средствами преобразования действительности (в данном случае и естественный язык, моделируемый посредством языка действий, понимается как средство преобразования действительности). В преобразуемую действительность включаются три части: сам язык действий (точнее, его формальное описание), управляемая система, формальное описание этой управляемой системы. Специфика преобразования действительности посредством применения языка действий состоит в том, что каждое такое преобразование описывается как последовательная реализация процессов *выбора* элементов из заданных множеств; *преобразования категорий* этих элементов; *поиска (создания)* или *построения* процессов преобразования и их последовательностей; *модификации* процессов.

Любой язык может быть описан как функция. Так, результатом употребления естественного языка (т.е. некоторого текста  $T$  на естественном языке) является либо изменение самого окружающего мира, либо представление (оценки) какого-то фрагмента этого мира у производящего или у слушающего текст  $T$ . Поэтому различие процедурных и декларативных способов формального описания языка характеризует не наличие/отсутствие динамического компонента в описании, а местонахождение этого компонента (в описании семантики или программной реализации). В случае языков действий строится динамическая семантика, ставящая в соответствие каждому тексту последовательность процессов указанных типов. Особенностью естественного языка является то, что он представляется как функция, при каждом своем употреблении воздействующая прежде всего на свое собственное описание, а также, возможно, на окружающий мир (это пояснялось на ряде примеров в [1]). В случае языков действий это свойство отражается следующим образом: текст на естественном языке задает стратегию вывода в формальном исчислении  $G$ , описывающем язык действий  $D$ , а интерпретацией этой стратегии является преобразование самого языка  $D$ , описываемое указанными типами процессов.

Пусть  $D$  - некоторый язык действий,  $\Phi$  - управляемая система,  $T$  - множество осмысленных (т.е. описывающих осуществимые преобразования  $\Phi$ ) текстов входного языка,  $G$  - внутренний язык описания  $D$ . Множество текстов  $T$  может быть бесконечным. Считается, что каждый текст  $t \in T$  является описанием некоторой стратегии вывода в  $G$ , интерпретируемой как преобразование  $D$  (т.е.  $t$  является описанием преобразования языком  $D$  самого себя). Следствиями выполнения этого преобразования могут быть изменения действительности  $M$ , включающей  $T$ ,  $D$  и  $\Phi$ . В любом языке действий  $D$  выделяются:  $L$  - средство перевода описаний стратегий преобразования  $D$  с входного языка на язык  $G$ ;  $\mathfrak{M}$  - формальное описание  $\Phi$  ("мир  $\mathfrak{M}$ ");  $Q$  - средство построения машинной програм-

мы, реализующей нужные преобразования  $M$ . Мир  $\mathfrak{M}$  описывается тройкой  $\langle G, B, I \rangle$ , где  $G$  - язык, задаваемый типовым исчислением предикатов первого порядка с условиями на применимость правил вывода;  $B$  - базовый мир - многосортная модель той же сигнатуры, что и  $G$  (элементы  $B$  будем называть "внутренними процессами");  $I$  - средство построения интерпретаций формул  $G$  на  $B$ .

Языки действий рассматриваются при следующих ограничениях.

1. Считается, что стратегия изменения языком  $D$  самого себя, задаваемая входным текстом (это изменение происходит при каждом применении  $D$  и в общем случае может касаться всех его составных частей), содержит только описание изменений  $B$ . Следствиями этих изменений  $B$  могут быть изменения других частей действительности  $M$ .

2. Предполагается, что управляемая система  $\Phi$  (которая в общем случае может совпадать с самим языком  $D$  или с множеством осмысленных текстов  $T$ ) является внешним объектом, не совпадающим ни с  $D$ , ни с  $T$ .

Кроме того, будем считать, что преобразования  $\Phi$  состоят в создании/удалении или пространственном перемещении элементов  $\Phi$ .

Для иллюстрации будем пользоваться примером языка действий  $D_0$  из [1]. Язык  $D_0$  предназначен для управления системой  $\Phi_0$ , представляющей собой экран дисплея, на котором могут строиться, уничтожаться и перемещаться различные геометрические фигуры (квадраты, круги, треугольники и т.д.), характеризующиеся определенными цветами, размерами и расположением. Для наименования понятий языка  $D_0$ , аналогичных  $T$ ,  $B$ ,  $G$ , будут использоваться обозначения  $T_0$ ,  $B_0$ ,  $G_0$  соответственно.

Каждая формула  $\phi$  языка  $G$  получается в результате преобразования некоторого входного теста  $t \in T$  посредством  $L$ . Если некоторый функциональный символ  $f$  формулы  $\phi$  получен в результате преобразования фрагмента  $t_1$  текста  $T$ , то будем называть  $t_1$  про-

образом  $f$ . Например, в  $D_0$  текст  $t = \text{"синий квадрат"}$  будет преобразован в формулу  $\varphi_1 = (\text{синий квадрат})(\rho)$  языка  $G_0$ . Преобразованиями функциональных символов *синий* и *квадрат* соответственно будут слова "синий" и "квадрат".

При введенных ограничениях на класс рассматриваемых языков действий результатом каждого изменения мира  $B$  является либо введение/удаление элементов  $B$  (т.е. внутренних процессов), либо переопределение семантических, ситуационных или программных категорий, поставленных в соответствие этим элементам. Например, если (при использовании языка  $D_0$ ) на экране еще нет изображения квадрата, то применение текста  $t_1 = \text{"квадрат"}$  приведет к построению на нем произвольного квадрата и введению имени этого квадрата в  $G$  и  $B_0$ ; применение текста  $t_2 = \text{"найти квадрат"}$  приведет к изменению ситуационных категорий, поставленных в соответствие выбранному (из числа имеющихся в  $B_0$ ) квадрату.

Языки действий могут быть использованы в  $\Sigma$ -программировании [2] в качестве специализированных модулей, обеспечивающих естественно-языковое взаимодействие. Приводимое ниже формальное описание языков действий можно рассматривать как функциональную спецификацию, определяющую принципы формирования таких модулей, их внутреннюю организацию и особенности функционирования.

## 1. Синтаксис языка действий

Синтаксис  $D$  описывается посредством языка  $G$  - типовое исчисление предикатов первого порядка с условиями на применимость правил вывода, задаваемое в следующей сигнатуре  $\langle F, Ex, X, C, N, \sigma, type, k, R \rangle$ , где  $F$  - множество функциональных символов одно- или двухместных, состоящее из множеств  $F_1, F_2, M^0 \cup M^d, \Omega$ , обозначающих соответственно элементарные имена объектов ( $F_1$ ) и действий ( $F_2$ ), модификаторы объектов ( $M^0$ ) и действий ( $M^d$ ), воз-

можно,  $M^0 \cap M^d \neq \emptyset$  (модификаторы служат для уточнения описаний объектов и действий, их прообразами могут быть прилагательные, наречия, предлоги и т.д.); функции, определяющие применимость правил вывода ( $\Omega$ );  $Ex$  - множество предикатных символов, определенных на множестве правильных формул  $G$  (предикаты реализуемости);  $X$  - множество символов переменных;  $C$  - множество константных символов, включающее символы неопределенных объектов, ситуаций, действий, процессов (соответственно  $\rho, \xi, \delta, \alpha$ ),  $C^0, C^C, C^P$  - константы типа объектов, ситуаций, процессов соответственно;  $\eta$  - символ невыполненного (в частности, невыполнимого) процесса,  $\omega$  - символ множества всех объектов;  $N$  - множество натуральных чисел;  $\mathcal{J}$  - множество, состоящее из символа дизъюнкции ( $\vee$ ), символа отрицания ( $\neg$ ), символов арифметических операций, кванторов  $\exists, \exists_\omega$  (где  $\omega \in N$ , выражение  $\exists_\omega x$  соответствует утверждению о существовании множества, состоящего из  $\omega$  элементов  $x$  таких, что для каждого  $x$  выполняется  $P(x)$ ), символы "=", " $\neq$ ", "[ ]", " $\bullet$ " (суперпозиция функций), "." (точка), "," (запятая);  $type$  - множество имен типов формул  $G$  (в  $G$  выделяются пять типов формул: модификаторы, объекты, ситуации, действия, процессы);  $K$  - множество имен категорий элементов из  $FUXUC$ ;  $R$  - множество правил вывода (множество функций  $\pi \in \Omega$ , поставленных в соответствие правилу  $a \in R$ , будем обозначать через  $\pi_a$ ).

На множестве  $F \cup \{\alpha, \delta, \xi, \rho\}$  определено отношение порядка " $\prec$ " с максимальным элементом  $\alpha$  (имя неопределенного процесса). Среди элементов множеств  $F_1$  и  $F_2$  максимальными являются соответственно  $\rho$  и  $\alpha$ ;  $\delta \prec \alpha$ ,  $\xi \prec \alpha$ ,  $\rho \prec \xi$ . Для любых  $a \in F_1$  и  $b \in F_2$  выполняются  $a \prec \rho$ ,  $b \prec \delta$ . Содержательно отношение " $\prec$ " может отражать соотношения "часть-целое", "род-вид" и т.д.

Множество  $\Omega$  состоит из функций, использующихся при определении условий применимости правил вывода. Сюда входят: 1) функция  $\tau$ , ставящая в соответствие каждой паре  $(f, \hat{f})$  (где  $f \in FUXUC$ ,

$\hat{f}$  - прообраз  $f$  во входном тексте) некоторое подмножество  $\gamma_f$  множества  $K$ , называемое дескриптором символа  $f$ ; 2) функция  $\pi_i$ , строящая дескриптор формулы  $\varphi = \varphi_1 \varphi_2$  по заданным дескрипторам  $\gamma_{\varphi_1}$ ,  $\gamma_{\varphi_2}$  двух ее подформул; 3) предикат  $k_\varepsilon(\varphi)$ , проверяющий принадлежность формулы  $\varphi$  категории с именем  $\varepsilon$ ; 4) функции  $\pi_i$ , ставящие в соответствие каждой паре  $(\varphi_1, \varphi_2)$  формул из  $D$  множество  $k_i = K_i \cap \gamma_{\varphi_1} \cap \gamma_{\varphi_2}$ , где  $k_i \subseteq K$ . Если  $\pi_i(\varphi_1, \varphi_2) \neq \emptyset$ , то выражения  $\varphi_1$  и  $\varphi_2$  называются согласованными по множеству категорий  $k_i \subseteq K_i$ . Через  $\pi$  обозначим множество, состоящее из всех функций  $\pi_i$ . Например, если условие состоит в проверке согласованности категорий падежа, поставленных в соответствие формулам  $\varphi_1$  и  $\varphi_2$ , то  $K_i$  - множество, состоящее из всех возможных имен категорий, обозначающих падежи в  $K$ ,  $k_i$  - те из них, которые присутствуют и в  $\gamma_{\varphi_1}$ , и в  $\gamma_{\varphi_2}$ , т.е. те, которые одинаковы у прообразов формул  $\varphi_1$  и  $\varphi_2$ .

Прежде, чем определять множество правил вывода исчисления  $G$ , приведем один пример из языка  $D_0$ . Текст  $t_2$ ="синий квадрат передвигается вправо" будет преобразован в формулу  $\varphi_2 = \exists x [x = (\text{синий} \bullet \text{квадрат}) (\rho)] \cdot (\text{вправо} \bullet \text{передвигать}) (x) \rightarrow s$ , предписывающей выбрать из множества синих квадратов, получае - мого в результате применения формулы  $(\text{синий} \bullet \text{квадрат}) (\rho)$ , некоторый один квадрат; обозначить его через  $x$ , выбрать некоторое допустимое расстояние  $u$  и передвинуть  $x$  вправо на величину  $u$ . Результатом преобразования мира  $B_0$ , соответствующего тексту  $t_2$ , является получение в  $B_0$  некоторой ситуации  $s$ , которая во входном тексте явно не определена. Функциональные символы *синий* и *вправо* входят в  $M^0$  и  $M^d$  соответственно.

Символы константных объектов, ситуаций, процессов именуют соответственно конкретные объекты  $B$  (например, символ *квадрат* №5 в  $B_0$ ); фиксированные конфигурации объектов (это могут быть, например, специальные имена ситуаций: *критическая си-*

туация, начальная ситуация); стандартные процессы (не требующие для своего определения задания ситуации, в которой они осуществляются, и действий, из которых они состоят, например, исходный процесс).

Прообразами символов неопределенных объектов, ситуаций, действий, процессов могут быть соответственно тексты "нечто", "множество объектов", "сделать что-нибудь в ситуации s" (где s - ситуация, определенная в предыдущей части текста), "сделать что-нибудь".

Правильные формулы языка G подразделяются на пять типов: множество правильных имен модификаторов (Mod), множество правильных имен объектов (O), множество правильных имен ситуаций (S), множество правильных имен действий (H), множество правильных имен процессов (W). Множество правильных формул G задается следующими правилами.

Множество  $R_{Mod}$  правил построения имен модификаторов языка G:

- 1)  $Mod_1$ :  $Mod = Mod^O \cup Mod^d$ ,  $M^O \subseteq Mod^O$ ,  $M^d \subseteq Mod^d$ ;
  - 2)  $Mod_2$ : для любых  $m_1, \dots, m_n \in Mod^O$  и  $l_1, \dots, l_n \in Mod^d$  ( $n > 0$ ) выражения  $\varphi = m_1 \bullet \dots \bullet m_n$  и  $\psi = l_1 \bullet \dots \bullet l_n$  принадлежат  $Mod^O$  и  $Mod^d$  соответственно (выражение, удовлетворяющее п.1 или п.2, назовем элементарным именем модификатора);
  - 3)  $Mod_3$ : для любых элементарных модификаторов  $\varphi \in M^O$  и  $\psi \in M^d$  таких, что функции  $\pi_{Mod_3}$  не равны  $\emptyset$ , выражения  $\neg\varphi$  и  $\neg\psi$  принадлежат  $Mod^O$  и  $Mod^d$  соответственно;
  - 4)  $Mod_4$ : для любых элементарных модификаторов  $\varphi_1 \varphi_2 \in Mod^O$  или  $\psi_1 \psi_2 \in Mod^d$  выражения  $\varphi_1 \vee \varphi_2$ ,  $\psi_1 \vee \psi_2$  принадлежат  $Mod^O$  и  $Mod^d$  соответственно;
  - 5)  $Mod_5$ : других правильных имен модификаторов в Mod нет.
- Множество  $R_O$  правил построения имен объектов языка G:
- 1)  $O_1$ :  $C^O \subseteq O$ ,  $\rho \in O$ ;

2)  $O_2$ : для любых  $a \in T_1$  и  $x \in X \cup \{o\}$  таких, что заданные функции из  $\pi_{O_2}$  не равны  $\emptyset$ , выражение  $a(x)$  принадлежит  $O$ ;

3)  $O_3$ : для любых  $a \in O$ ,  $a \notin C^O$ ,  $m \in \text{Mod}^O$ ,  $x \in X\{p\}$ , таких, что заданные функции из  $\pi_{O_3}$  не равны  $\emptyset$ , выражение  $m(a(x)) = (m \bullet a)(x)$  принадлежит  $O$  (выражение, удовлетворяющее одному из пп. 1-3, назовем элементарным именем объекта);

4)  $O_4$ : для любого элементарного имени объекта  $p \in O$  такого, что  $p \neq r$  и заданные функции из  $\pi_{O_4}$  не равны  $\emptyset$ , выражение  $\neg p$  принадлежит  $O$ ;

5)  $O_5$ : для любых выражений  $p_1, p_2$  (каждое из которых является либо именем элементарного объекта, либо его отрицанием) таких, что заданные функции из  $\pi_{O_5}$  не равны  $\emptyset$ , выражение  $p_1 \vee p_2$  принадлежит  $O$ ;

6)  $O_6$ : других правильных имен объектов в  $O$  нет.

Имена объектов устроены таким образом, что в них не могут содержаться вложенные знаки отрицания. Область действия каждого из знаков отрицания описывается отдельно. Например, применение входного текста "квадрат, не расположенный на круге, не лежащем на треугольнике" из  $T_O$  приведет к построению таких имен объектов: *треугольник*( $p$ ), (*круг*  $\bullet$   $\neg$  *на<sub>x</sub>*)( $p$ ), (*квадрат*  $\bullet$   $\neg$  *на<sub>y</sub>*)( $p$ ), где через  $x$  и  $y$  во второй и третьей формулах обозначены имена объектов, заданные ранее (в данном случае они определяются первой и второй формулой соответственно).

Выражению  $p \in O$  может соответствовать один элемент мира  $B$ , т.е. один внутренний процесс (тогда  $p$  относится к категории "единичный объект") или множество элементов (тогда  $p$  относится к категории "множественный объект"), например, в  $D_O$  может быть "единичный объект" *квадрат* и "множественный объект" *квадраты*.

Множество  $R_S$  правил построения имен ситуаций языка  $G$ :

1)  $S_1$ :  $\xi \in S$ ,  $C^C \subseteq S$ ;



2)  $S_2$ : для любых  $x \in X$ ,  $p \in 0$ ,  $\omega \in N$  выражение  $s = \exists_{\omega} x [x=p]$  принадлежит  $S$  (если индекс  $\omega = 1$ , то он опускается);

3)  $S_3$ : для любых  $s_1 = \exists_{\omega_1} x_1 [x_1=p_1], \dots, s_n = \exists_{\omega_n} x_n [x_n=p_n] \in S$  выражение  $s(x_1, \dots, x_n)$ , равное  $\exists_{\omega_1} x_1 \dots \exists_{\omega_n} x_n [x_1=p_1, \dots, x_n=p_n]$ , принадлежит  $S$  (формулу, удовлетворяющую одному из пп.1-3, будем называть элементарным именем ситуации; если элементарное имя ситуации  $s(x_1, \dots, x_n)$  входит в некоторое  $s' \in S$ , то будем писать, что  $x_1 \in s', \dots, x_n \in s'$ );

4)  $S_4$ : для любой формулы, являющейся элементарным именем ситуации ( $s \neq \xi$ ) такой, что заданные функции  $\pi_{S_4}$  не равны  $\emptyset$ , выражения  $\neg s(x_1, \dots, x_n)$  и  $Ex(s)$  принадлежат  $S$ ;

5)  $S_5$ : для любых выражений  $s_1, s_2$  (каждое из которых является либо элементарным именем ситуации, либо его отрицанием) таких, что заданные функции  $\pi_{S_5}$  не равны  $\emptyset$ , выражения  $s_1 \vee s_2$  и  $s_1 \bullet s_2$  принадлежат  $S$ ;

6)  $S_6$ : других правильных имен ситуаций в  $S$  нет.

Правильное имя ситуации не содержит свободных переменных.

Множество  $R_H$  правил построения имен действий языка  $G$ :

1)  $H_1$ :  $C^A \in H$ ,  $\delta(s_1) \in H$ , где  $s_1 \in S$ ;

2)  $H_2$ : для любых  $d \in F_2$  ( $d$  -  $n$ -местная функция) и  $x_1, \dots, x_n \in X$  таких, что заданные функции  $\pi_{H_2}$  не равны  $\emptyset$ , выражение  $d(x_1, \dots, x_n)$  принадлежит  $H$ ; если действие  $d$  применяется к ситуации  $s$ , содержащей  $x_1, \dots, x_n$  в качестве переменных, связанных квантором  $\exists$  или  $\exists_{\omega}$ , то будем писать  $d(s)$ ;

3)  $H_3$ : для любых  $g \in H$ ,  $m \in \text{Mod}^d$  таких, что заданные функции  $\pi_{H_3}$  не равны  $\emptyset$ , выражение  $m \bullet g$  принадлежит  $H$  (формулу, удовлетворяющую пп.1-3, назовем элементарным именем действия);

4)  $H_4$ : для любой формулы  $g \in H$ , являющейся элементарным именем действия ( $g \neq \delta$ ), такой, что заданные функции  $\pi_{H_4}$  не равны  $\emptyset$ , выражения  $\neg g$  и  $Ex(g)$  принадлежат  $H$ ;

5)  $H_5$ : для любых формул  $g_1, g_2 \in H$  (каждая из которых является либо элементарным именем действия, либо его отрицанием) таких, что заданные функции  $\pi$  не равны  $\emptyset$ , выражения  $g_1 \vee g_2$ ,  $g_1 \odot g_2$ ,  $g_1 \rightarrow g_2$  принадлежат  $H$ ;

6)  $H_6$ : других правильных имен действий в  $H$  нет.

Действие  $g \in H$  определяет схему (или план) изменения  $B$ . Для того чтобы преобразовать эту схему в описание реального изменения  $B$ , необходимо задать ситуацию осуществления действия  $g$ .

Множество  $R_W$  правил построения имен процессов языка  $G$ :

1)  $W_1$ :  $\alpha \in W$ ,  $C^N \subseteq W$ ;

2)  $W_2$ : для любых  $g \in H$ ,  $s_1 s_2 \in S$  таких, что, если  $g \neq \delta$  и  $s_1 \neq \xi$ , то для любой свободной переменной  $x$  выражения  $g$  верно  $x \subseteq s_1$ , и заданные функции из  $\pi_{W_2}$  не равны  $\emptyset$ , выражение  $s_1 \cdot f(s_1) \rightarrow s_2$  принадлежит  $W$ ;  $s_1$  назовем исходной ситуацией,  $s_2$  - результирующей ситуацией (формулу, удовлетворяющую п.1 или п.2, назовем элементарным именем процесса);

3)  $W_3$ : для любого элементарного имени процесса  $g \in W$  ( $g \neq \alpha$ ) такого, что заданные функции  $\pi_{W_3}$  не равны  $\emptyset$ , выражения  $\neg g$  и  $Ex(g)$  принадлежат  $W$ ;

4)  $W_4$ : для любых формул  $g_1, g_2 \in W$  (каждая из которых является либо элементарным именем процесса, либо его отрицанием) таких, что заданные функции  $\pi_{W_4}$  не равны  $\emptyset$ , выражения  $g_1 \vee g_2$ ,  $g_1 \odot g_2$ ,  $g_1 \rightarrow g_2$  принадлежат  $W$ ;

5)  $W_5$ : других правильных имен процессов в  $W$  нет.

Если рассматриваются только непрерывные процессы, то все выражения  $\varphi \in W$  сводятся к префиксной форме, в которой располагаются сначала все описания объектов, а затем - все описания

действий. Это объясняется тем, что не найдя все объекты, участвующие в процессе, и не построив все необходимые ситуации, бессмысленно говорить о выполнении процесса.

Полным описанием процесса  $\varphi$  (или замыканием  $\varphi$ ) называется формула, получающаяся в результате дополнения вывода формулы  $\varphi$  всеми последовательностями правил вывода из  $G$ , применимыми к  $\varphi$  (т.е. дополнение всех преобразований мира  $B$  их следствиями).

Формулы  $g_1 \rightarrow g_2$  ( $g_1, g_2 \in W$ ) определяют причинно-следственные отношения на множестве преобразований  $B$ . Использование правила  $g_1 \rightarrow g_2$  при выводе некоторой формулы  $\psi \in D$  не детерминировано входным текстом (т.е. не содержится в задаваемой им стратегии вывода). Применение правила  $g_1 \rightarrow g_2$ , где  $g_1, g_2 \in W$ , детерминировано входным текстом (входит в определение стратегии) и соответствует условному оператору языка  $D$ , требующему выполнения  $g_2$  в случае выполнения  $g_1$ .

Пусть формула  $\varphi \in G$  получена непосредственно из формул  $\varphi_1, \dots, \varphi_n$  в результате применения к ним одного из перечисленных выше правил построения имен объектов, ситуаций, действий или процессов. Выберем только те  $\varphi_i$ ,  $0 < i < n+1$ , которые не входят в  $X \cup \{\phi\}$ . Если таких формул окажется не меньше двух, назовем их непосредственными составляющими формулы  $\varphi$ .

При построении семантики формулы из  $G$ , содержащие символы отрицания, преобразуются следующим образом. Если областью действия символа отрицания является некоторая формула  $\varphi \in D$ , имеющая непосредственные составляющие  $\varphi_1, \dots, \varphi_n$ , то выбирается любая составляющая  $\varphi_i$ ,  $1 \leq i \leq n$ , знак отрицания переносится на нее. Затем все  $\varphi_j$ ,  $1 \leq j \leq i$ , изменяются следующим образом. Если  $\varphi_j$  - имя модификатора, то оно опускается, если  $\varphi_j$  - имя объекта, ситуации, действия или процесса, то оно заменяется на  $\phi, \xi, \delta, \alpha$  соответственно. Например, если  $\varphi = (\neg m \bullet a)(x)$ , то выражение  $\neg \varphi$  может быть преобразовано либо к виду  $((\neg m \bullet a)(x))$ , ли-

бо к виду  $(\neg a)(x)$ . Такая процедура переноса символов отрицания на составляющие заданных формул продолжается до тех пор, пока каждый символ отрицания не будет иметь в качестве области своего действия символ из  $F \cup C$ . Выбор составляющей, на которую переносится символ отрицания, осуществляется согласно условиям, определенным в В.

## 2. Семантика языка действий

Определим *базовый мир* В как множество внутренних процессов  $U = \Theta \cup E \cup Q \cup Z \cup J$ . Внутренние процессы В имеют вид:  $s_1.f(s_1) \rightarrow \xi$ , где  $\xi$  - неопределенная ситуация, поэтому они будут обозначаться как функции вида  $f(s_1)$  и (чтобы отличать их от одного из типов формул языка D) выделяться угловыми скобками  $\langle \rangle$ .

Выделяются следующие типы внутренних процессов:

Z - процессы *преобразования* категорий в В (изменение категориальной принадлежности имеющихся внутренних процессов и создание новых внутренних процессов);

J - процессы *модификации*;

$\Theta$  - процессы *поиска (создания)* элементов Z (имена которых совпадают с  $F \cup \{\alpha, \delta, \xi, \rho\}$ );

E - процессы *построения* последовательностей, состоящих из элементов Z, возможно, модифицированных посредством процессов из J (т.е. внутренние процессы построения последовательностей внутренних процессов);

V - процессы *выбора* представителя из заданного множества внутренних процессов или значения переменной из заданного допустимого множества значений.

Процессы из  $\Theta$  могут содержать внутри себя другие процессы из  $\Theta$ , V, Z или обращения к процессам из  $\Theta$ , V, Z. Процессы модификации применяются к элементам из  $\Theta$ . Каждый процесс  $m \in J$ ,

будучи применен к  $q \in \Theta$ , изменяет область определения некоторых вызываемых внутри  $q$  процессов выбора. Вследствие этого изменяется возможность реализации некоторого  $z \in Z$ , содержащегося в  $\Theta$  (или вызываемого в  $\Theta$ ). Таким образом, конечным результатом применения  $\pi$  к  $\Theta$  является изменение процесса преобразования  $z$ .

Операция изменения процессом  $z \in Z$  самого себя переводится посредством  $Q$  в некоторую машинную программу, изменяющую мир  $\Phi$ . Например, пусть  $t = \text{"квадрат"}$ . Этот текст будет проинтерпретирован языком действий  $D_0$  как некоторый процесс  $z \in Z$  (порождающий всевозможные квадраты). Выполнение процесса  $z$  состоит в изменении им самого себя путем включения в свое описание имени некоторого квадрата  $q$ , характеризующегося выбранными при выполнении  $z$  значениями параметров цвета, размера и расположения (теперь этот квадрат  $q$  будет фигурировать в  $z$  как элемент, имеющий индивидуальное имя, которое можно использовать в других модулях  $B$  и к которому можно обращаться через входные тексты). Следствием изменения процессом  $z$  самого себя будет являться машинная программа (сопоставляемая посредством  $Q$ ) построения на экране квадрата, характеризующегося выбранными значениями параметров.

Интерпретацию формулы  $\phi$  на  $B$  будем обозначать через  $|\phi|_B$ . Интерпретация  $\phi$  включает доопределение стратегии, заданной формулой  $\phi$ , до описания конкретного способа преобразования  $B$ . Пусть  $d(\phi)$  - множество всех возможных доопределений формулы  $\phi$  в  $B$  (получающееся в результате разных способов выполнения внутренних процессов выбора, содержащихся в  $|\phi|_B$ ), тогда  $|d(\phi)|_B = \{|\psi|_B / \psi \in d(\phi)\}$  - множество всех возможных интерпретаций  $\phi$  на  $B$  (если выбор стратегии в  $B$  случаен, то интерпретатор  $I$  может поставить в соответствие формуле  $\phi$  любое из этих доопределений),  $r(|\phi|_B)$  - выбранная интерпретация,  $\phi$  - процесс преобразования  $B$ , полученный в результате выполнения процессов выбора, вызываемых в  $|\phi|_B$ .

Интерпретация формулы  $\varphi \in G$  строится посредством применения последовательности операций. Например, если  $\varphi \in W$ , то первоначально формуле  $\varphi$  ставится в соответствие внутренний процесс, соответствующий неопределенному имени процесса  $\alpha$ . Затем к этому внутреннему процессу последовательно применяются операторы, входящие в  $\varphi$ . Например, если  $\varphi = \exists x [x = \text{синий} \bullet \text{квадрат}(\rho)] . \text{передвинуть}(x)$  ("передвинуть синий квадрат"), то первоначально  $\varphi$  ставится в соответствие процесс  $\alpha$ , внутри которого есть обращения к процессам  $\xi, \rho$  и  $\delta$ . Затем к  $\alpha$  применяется оператор *квадрат*, переводящий процесс порождения произвольной ситуации в процесс порождения произвольного квадрата. К полученному таким образом процессу применяется оператор *синий*, переводящий процесс порождения произвольного квадрата в процесс порождения синего квадрата. В обоих случаях изменяется область определения некоторого процесса выбора: в первом случае область, состоящая из всевозможных ситуаций, переводится в область всех квадратов, во втором случае область, состоящая из квадратов всех цветов, переводится в область, состоящую из всех синих квадратов. Таким образом, если мы обозначим множество областей определения функций выбора, вызываемых во внутреннем процессе  $p$ , через  $\Delta_p$ , то интерпретация  $f_Y$  функции  $f \in F \cup \{\alpha, \xi, \delta, \rho\}$  может быть представлена как функция вида  $f_Y(\Delta_\alpha) = \Delta_{f_Y} \in \Delta_\alpha$ . Таким же образом представляется и интерпретация процесса модификации.

Семантикой  $u = \|\varphi\|_B$  некоторой формулы  $\varphi \in D$  такой, что  $u \notin W$ , является последовательность внутренних процессов. Выполнение некоторых из этих процессов может задерживаться. Сигналом для реализации процессов, входящих в  $\varphi$ , может быть, например, построение по входному тексту некоторой формулы  $\psi \in W$ , включающей  $\varphi$ . Обозначим через  $\|u\|$ , где  $u$  - семантика формулы  $\varphi$  в  $B$ , *операцию реализации* всех внутренних процессов, вызовы которых входят в  $u$ . Например, тексту  $\tau = \text{"найти квадрат №1"}$

и построить треугольник<sup>11</sup> будет соответствовать в D формула:  
 $\varphi = \exists x \exists y [x = \text{квадрат } \#1, y = \text{треугольник}(p)] . \text{найти}(x) \& \text{пост-}$   
 $\text{роить}(y) \rightarrow s$ . В результате интерпретации этой формулы констан -  
 те *квадрат #1* будет поставлен в соответствие внутренний про-  
 цесс из Z, включающий нужный объект в текущую ситуацию. Имя  
 объекта *треугольник* интерпретируется посредством процесса g  
 из  $\Theta$ , порождающего произвольный треугольник (внутри процесса g  
 имеются вызовы внутренних процессов выбора из Q, определяющих  
 цвет, величины сторон и углов, расположение треугольника). Име-  
 нам действий *найти* и *построить* ставятся в соответствие по -  
 следовательности внутренних процессов q из  $\Theta$ . Имя процесса  $\varphi$   
 интерпретируется как применение операции реализации ко всем пе-  
 речисленным выше внутренним процессам, входящим в построенные  
 интерпретации подформул формулы  $\varphi$ .

Мир В может функционировать в автономном режиме (без обра-  
 щения на естественном языке), реализуя допустимые в нем внут -  
 ренние процессы произвольным образом.

Семантика языка G определяется на модели  $V = \langle B, Y \rangle$ , где  
 B - базовый мир;  $Y = \{Y^c, Y^o, Y^d, Y^m\}$  - отображения, ставящие в  
 соответствие значения элементам из  $F \cup C$  (обозначим  $Y(a)$  через  
 $a_Y$ ):  $Y^c$  ставит в соответствие каждой константе  $c \in C^O$  процесс  
 из Z; каждой константе  $c \in C^C$  - фиксированную последовательно-  
 ность процессов из Z, каждой константе  $c \in C^P$  - операцию реали-  
 зации фиксированной последовательности процессов поиска из  $\Theta$ ;  
 $Y^o, Y^d$  ставят в соответствие каждому  $f \in F_1$  и каждому  $g \in F_2$   
 процесс поиска (создания) из Z;  $Y^m$  ставит в соответствие каж-  
 дому  $m \in \text{Mod}$  внутренний процесс из J, определенный на процессе  
 поиска (создания) из Z и изменяющий некоторый вызываемый внут-  
 ри него процесс выбора.

На множестве  $\Theta$  зададим отношение порядка  $\prec_B$ . Отношение  
 $a \prec_B b$  означает, что в описании внутреннего процесса b содер-  
 жится обращение к внутреннему процессу a. Мир В устроен таким

образом, что отношение  $\prec_B$  изоморфно отношению  $\prec$ , определенному на  $F \cup \{\alpha, \delta, \xi, \rho\}$ .

Обозначим через  $\Delta_q$  область определения процесса выбора  $q \in Q$ . На каждой области  $\Delta_q$  зададим набор внутренних процессов  $\eta_q$ , ставящих в соответствие области  $\Delta_q$  некоторую ее под-область  $\Delta'_q$ .

Интерпретация имен модификаторов:

$|m|_B = \langle m_Y \rangle \in \{\eta_q / q \in Q\}$ , т.е. интерпретацией имени модификатора является внутренний процесс, сужающий область определения  $\Delta_q$  некоторого процесса выбора  $q$  до области  $\Delta'_q$ ;

$|\neg m|_B$  - внутренний процесс, сужающий область определения  $\Delta_q$  некоторого процесса выбора  $q$  до области  $\Delta_q \setminus \Delta'_q$ .

Пусть элементам  $\alpha, \rho, \delta, \xi$  из  $G$  поставлены в соответствие в  $B$  внутренние процессы  $\alpha_B, \rho_B, \delta_B, \xi_B$ . Элемент  $\alpha_B$  является максимальным по отношению  $\prec_B$ ;  $\rho_B, \delta_B$  - максимальные элементы (по отношению  $\prec_B$ ) для внутренних процессов, поставленных в соответствие элементам из  $F_1, F_2$  отображением  $Y$ ;  $\rho_B \prec_B \xi_B, \delta_B \prec_B \alpha_B, \xi_B \prec_B \alpha_B$ .

Интерпретация имен объектов:

$|x|_{B, s_1} = r(|p|_B)$ , если  $x \sqsubset s_1$  для некоторого  $s_1 \in S$  и  $x=p$  в  $s_1$  (семантика переменной определяется только относительно ситуации, в которой она присутствует);

$|c|_B = \langle c_Y \rangle$  - внутренние процессы, поставленные в соответствие константе  $c \in C^0$ ;

$|\rho|_B = \langle \rho_B \rangle$ ;

$|a(u)|_B = \langle a_Y([u]_B) \rangle$  для  $a \in F_1, u \in O$ ;

$|\neg p|_B = \langle \rho'_B \rangle$  для  $p \in O$ ; процесс  $\rho'_B$  получается из  $\rho_B$  заменой области определения  $\Delta_a$  некоторого процесса выбора  $a$  на область  $\Delta'_a$ , где  $a \prec_B \rho_B$  (т.е. внутри процесса  $\rho_B$  имеется обращение к процессу  $a$ ),  $\Delta_a$  включает процесс  $p$ , область  $\Delta'_a = \Delta_a \setminus \{p\}$ ;

$|p_1 \vee p_2|_B = \langle q(\{|p_1|_B, |p_2|_B\}) \rangle$  для  $p_1, p_2 \in O$ .



### Интерпретация имен ситуаций:

$|c|_B = \langle c_Y \rangle$  – последовательность процессов поиска (создания) объектов, входящих в константную ситуацию  $c \in C^C$ ;

$|\xi|_B = \langle q(|\rho|_B \wedge |\xi|_B, |\xi|_B = |\rho|_B) \rangle$  для некоторого  $q \in Q$ ;

$|\exists_w x[x=p]|_B = \langle |p_1|_B \wedge |p_2|_B \wedge \dots \wedge |p_n|_B \rangle$ , где  $p_1 = p$ ,  $p_i$  получается из  $p_{i-1}$  путем применения функции  $y(\Delta_{p_{i-1}}) = \Delta_{p_{i-1}} \setminus |p_{i-1}|_B$  для  $i = 2, \dots, \omega$ , т.е. семантикой элементарной ситуации является *цикл*, состоящий из последовательного использования процессов выбора;

$$|s_1 \bullet s_2|_B = \langle |s_1|_B \bullet |s_2|_B \rangle \text{ для } s_1, s_2 \in S;$$

$$|s_1 \vee s_2|_B = \langle q(|s_1|_B, |s_2|_B) \rangle \text{ для } s_1, s_2 \in S;$$

$$Ex(s) = \begin{cases} \varepsilon, & \text{если } |s|_B \neq \eta, \\ \eta & - \text{ в противном случае.} \end{cases}$$

### Интерпретация имен действий:

$$|c|_B = c_Y, \quad c \in C^D;$$

$$|\delta|_B = \langle \delta_B \rangle;$$

$$|g|_B = \langle g_Y(\delta_B) \rangle \text{ для } g \in H;$$

$$|g_1 \vee g_2|_B = \langle q(|g_1|_B, |g_2|_B) \rangle;$$

$$|g_1 \wedge g_2|_B = \langle |g_1|_B \bullet |g_2|_B \rangle;$$

$|\neg g_1|_B = \langle \delta'_B \rangle$ , где  $\delta'_B$  получается из  $\delta_B$  путем замены области определения  $\Delta_\alpha$  некоторого процесса выбора  $a$  на область  $\Delta'_a$ , где  $a \prec_B \delta_B$ ,  $\Delta_a$  включает процесс  $g$ ,  $\Delta'_a = \Delta_a \setminus \{g\}$ ;

$$|g_1 \rightarrow g_2|_B = \langle |g|_B \bullet |g|_B \rangle,$$

$$|Ex(g)|_B = \begin{cases} \varepsilon, & \text{если } |g|_B \neq \eta, \\ \eta & - \text{ в противном случае.} \end{cases}$$

Интерпретации выражений  $|g_1 \bullet g_2|_{B,s}$  и  $|g_1 \rightarrow g_2|_{B,s}$  совпадают. Различие между ними - в применявшихся правилах вывода. Правило вывода, использовавшееся при получении первого, входит в описание стратегии построения формулы, соответствующей всему входному тексту. Правило вывода, использовавшееся при получении второго выражения, не входит в описание стратегии и добавляется при построении ее замыкания.

Интерпретация процессов:

$$|c|_B = \|\langle c_Y \rangle\| \text{ для } c \in C^N;$$

$$|\alpha|_B = \|\langle \alpha_B \rangle\|;$$

$$|s_1 \cdot f(s_1) \rightarrow \xi|_B = \|\langle |s_1|_B \bullet |f(r(|s_1|_B))|_B \rangle\| \text{ для } s_1 \in S, f \in H;$$

$$|s_1 \cdot f(s_1) \rightarrow s_2|_B = \|\langle |s_1|_B \bullet q(d(f(s_1))) \bullet |Ex(s_2)|_B \rangle\| \text{ для}$$

$s_1, s_2 \neq \xi$  и для некоторого  $q \in Q$ ;

$|\neg g|_B = \|\eta\|$  ( $\eta$  - символ невыполненного, в частности, невыполнимого процесса). Тексты "сделать не  $f$ " и "не делать  $f$ " интерпретируются как  $\neg f$ , где  $f$  в первом случае относится к типу "действие", а во втором - к типу "процесс";

$$|Ex(g)|_B = \begin{cases} \epsilon, & \text{если } |g|_B \neq \|\eta\|, \\ \eta & - \text{в противном случае.} \end{cases}$$

### 3. Алгоритмическая реализация

Выделим в описании  $G$  три структурно и функционально разделенные части: словарь  $Vok$ , грамматику категорий  $\Gamma^K$ , грамматику правил  $\Gamma^r$ . Каждую часть организуем как автономную единицу, которую можно строить, корректировать или использовать в справочно-информационном режиме независимо от остальных частей и системы в целом.

Словарь  $Vok$  будем представлять как списковую структуру, в которой содержатся слова входного естественного языка. Элементы  $W$ , имеющие одинаковое начало, "склеиваются" друг с дру-

гом таким образом, чтобы это общее начало записывалось в  $W$  только один раз. Например, общему началу слов "круг" и "красный" соответствует один и тот же набор строк списка. Каждому слову  $w$  в словаре  $Vok$  поставлена в соответствие определенная последовательность  $k(w)$  категорий. Словарь не может содержать слов без указания их категорной принадлежности. Уничтожение всех категорий, приписанных некоторому слову  $w \in Vok$ , равносильно исключению слова  $w$  из словаря.

Вторая часть системы - *грамматика категорий* ( $\Gamma^K$ ). Словарь определяет состав каждой категории из  $K$  (множество относящихся к ней слов). Назначение грамматики категорий - описание отношений на множестве имен категорий. Это могут быть отношения *вложенности* (родовидовые, часть-целое) и отношения *альтернативности* (между частями одного целого, видами одного рода), отношения *применимости* функции к аргументу, отношения *допустимости суперпозиции* функций и т.д. Перечень отношений не является универсальным и зависит от предметной области. Грамматика категорий определяет закономерности сочетаний категорий во входных текстах, т.е. задает *данные*, использующиеся для определения условий применимости правил в грамматике правил. Множество категорий должно быть сформировано таким образом, чтобы оно позволяло формально описать (грамматические, синтаксические, семантические) условия применимости правил.

*Грамматика правил* ( $\Gamma^r$ ), описывающая множество  $R$  правил исчисления  $G$ , определяет допустимые преобразования текста вместе с условиями их применимости, задавая и те, и другие посредством последовательностей имен категорий. Грамматика правил используется для построения стратегии вывода формулы  $G$ , заданной входным (естественно-языковым) текстом.

*Базовый мир*  $B$  задается совокупностью модулей, определяющей способы порождения различных состояний мира  $B$  или

способы перехода из одного состояния  $B$  в другое. Все модули организованы как *генераторы*, т.е. предназначены для порождения интерпретаций объектов, ситуаций, действий, процессов языка  $G$ .

Типы модулей внутренних процессов соответствуют типам формул языка  $G$  и подразделяются на: модули поиска (создания) объектов и действий (имена которых совпадают с именами из  $F_1 \cup F_2 \cup C^O \cup C^P$ ), модули построения, модули выбора, модули преобразования, модули модификации.

*Модули поиска (создания)* подразделяются на *детерминированные* (соответствующие поиску объектов и действий, именуемых константами из  $C^O$  и  $C^P$ ) и *недетерминированные* (которым соответствуют в языке  $D$  элементы из  $F_1$  и  $F_2$ ).

Недетерминированность модуля  $a$  является следствием содержащихся в его описании обращений к *модулям выбора*. В каждом модуле выбора  $f$  с областью определения  $\Delta_f$  могут быть установлены свои вероятности выбора элементов из  $\Delta_f$ . Результатом работы каждого модуля выбора является приписывание значения некоторой переменной (путем случайного выбора из заданной области в соответствии с заданным распределением вероятностей). Некоторые из этих переменных могут фигурировать в качестве параметров модуля  $a$ . Такие переменные будем называть *внешними*. Кроме них, могут существовать *внутренние* переменные, не являющиеся параметрами данного модуля. Если  $x$  - внешняя переменная модуля  $a$ , то область ее значений может сужаться в результате применения к  $a$  некоторого модуля модификации.

Например, внутренний процесс с именем "стол" может уточняться модификаторами, определяющими значения переменных "цвет", "форма", "размер" и заданными как внешние (т.е. доступные извне, из входного языка) переменные. Модификаторы могут или задавать конкретное значение переменной (например, "красный"), или сужать значение переменной (например, "темный"). Внут-

ренными переменными могут быть, например, переменные типа "место расположения ножек у стола" и т.д. Таким образом, внутренние возможности вариации переменных могут быть шире, чем внешние. Вопрос о разделении переменных на внешние и внутренние связан с вопросом о выразительности входного языка и может меняться при ее увеличении (внутренние переменные могут переводиться во внешние). Аналогичным образом определяются внутренние и внешние переменные для действий.

*Модуль поиска (создания)* задает множество объектов и действий генеративно, т.е. процедурой порождения элементов этого множества (которая в общем случае может порождать бесконечное число разных объектов или действий).

Каждый модуль поиска (создания) содержит две части:

- 1) последовательность модулей выбора (или обращений к таким модулям), порождающих значения внешних и внутренних переменных,
- 2) последовательность модулей преобразования (или обращений к ним), осуществляющих преобразования мира  $B$ .

*Модули построения* служат для формирования последовательностей имен, вызываемых модулей поиска (создания), модификации, выбора.

*Модули преобразования* осуществляют изменение мира  $B$ , т.е. изменение категорной соотнесенности элементов, входящих в описанную ситуацию.

Возможны изменения состава модулей. Например, можно ввести новое имя для последовательности модулей, соответствующей некоторой формуле  $\varphi$  языка  $D$ , т.е. ввести некоторый *сценарий* — стандартную последовательность процессов в  $G$ . Недоопределенность преобразований  $B$  в пределах данного сценария обеспечивается тем, что функциональным символам из  $F_1 \cup F_2 \cup \text{Mod}$  в семантике  $G$  соответствуют недетерминированные процедуры генерации.

#### 4. Способы языкового взаимодействия в рамках языка D

Как указывалось в [1], язык D может использоваться в двух режимах: *управления* (приведенное выше описание D) и *обучения*.

Исходным для обучения является мир B, состоящий из двух понятий: о неопределенном объекте  $\rho$  и неопределенном действии  $\delta$  мира B. Описание каждого из понятий  $\rho_0$  и  $\delta_0$  включает набор *внутренних переменных*  $p_1, \dots, p_n$  и  $p_{n+1}, \dots, p_{n+1}$  соответственно. Каждая переменная  $p_i$  определяется (внутри некоторого модуля выбора) именем  $\pi_i$ , областью значений  $\omega_i$  и множеством вероятностей этих значений  $z_i$  (в частности, можно предполагать, что все значения равновероятны). Каждый модуль выбора задается как генератор, порождающий некоторые значения включенных в него переменных. Процессы, производимые в таком B, являются следствиями выполнения процессов преобразования, содержащихся в  $\rho_0$  и  $\delta_0$ , использующих случайные комбинации значений переменных, порожденные модулями выбора. В этом случае мир B изменяется максимально недетерминированно и входной (естественный) язык может инициировать в нем только преобразования вида "сделать что-нибудь".

Дальнейшее обучение может состоять в: добавлении к  $\rho_0$  и  $\delta_0$  новых процессов выбора, расширении областей определения уже имеющихся процессов выбора или конструирования новых внутренних процессов (поиска, создания, модификации, построения) путем комбинирования процессов выбора и преобразования, входящих в  $\rho_0$  и  $\delta_0$ . В результате такого дообучения оказывается возможным осуществление более детерминированных преобразований B. Обучение (дообучение) может производиться в двух режимах: в *режиме управления* G и в *режиме автономного функционирования* G. При первом режиме все производимые изменения G описываются явным образом. В обоих случаях "управляемость" мира B повышается за счет введения новых средств *наименования*.

При втором режиме могут (но не обязательно) задаваться только имена интересующих пользователя объектов или действий (т.е. тех, внутри которых необходимо выделить подпонятия или переопределить их каким-либо другим образом). Мир В преобразуется случайным образом (определяемым случайным выбором значений переменных), без обращений на входном языке. В ходе последовательной генерации таких преобразований могут появляться объекты (модификаторы, действия), которые пользователь сочтет нужным *назвать*, т.е. выделить и использовать далее в качестве самостоятельной конструктивной единицы мира В (при этом пользователь может заранее точно и не знать, что именно он будет выделять в качестве конструктивных единиц В).

В статье описан один из способов самоизменения D. Возможны и другие способы. Можно распространять область действия внутренних процессов построения, поиска (создания), преобразования, модификации, выбора не только на область семантики (т.е. базовый мир В), как это было предложено в данной статье, но и на область синтаксиса G. Если в процессе функционирования язык D может изменять не только свою семантику, но и свой синтаксис, то входной язык может быть обогащен таким образом, чтобы он допускал тексты, состоящие из нескольких связанных между собой предложений, и некоторые случаи метафор. Рассмотрение этого - предмет отдельной публикации.

#### Л и т е р а т у р а

1. ТИМОФЕЕВА М.К. Язык действий //Методологические проблемы науки. - Новосибирск, 1991. - Вып. 138: Вычислительные системы. - С. 61-77.
2. ГОНЧАРОВ С.С., СВИРИДЕНКО Д.И. Σ-программирование //Логико-математические основы проблемы МОЗ. - Новосибирск, 1985. - Вып. 101: Вычислительные системы. - С. 3-29.

Поступила в редакцию

23 ноября 1994 г.