

УДК 519.766

## ЯЗЫКИ ДЕЙСТВИЙ

М.К.Тимофеева

*Языками действий* [1] назовем класс формальных языков, основное назначение которых - управление сложными динамическими системами произвольной природы. Будем полагать, что тексты, написанные на некотором языке действий  $D$ , внешне выглядят как тексты предметно и функционально ограниченного естественного языка.

Наиболее ранние из средств формализации естественного языка определяли его как систему, рассматриваемую независимо от области своего использования. Более поздние - как средство описания ситуаций (например, ситуационная семантика [2]). Языки действий формализуют естественный язык, описывая его как *средство преобразования* действительности (которая может включать и "говорящего", и "слушающего", и сам естественный язык), ориентированное на определенный класс "языковых преобразований". В этом - основа специфики языков действий по сравнению с другими средствами формализации естественного языка. Языки действий предполагается использовать в системе  $\Sigma$ -программирования [3].

1. Пример языка действий. Рассмотрим следующий пример. Пусть управляемая динамическая система  $\Phi_0$  - это экран дисплея, на котором могут строиться, уничтожаться или перемещаться различные геометрические фигуры (квадраты, круги и т.д.), характеризующиеся определенными цветами, размерами и расположением.

Система  $\Phi_0$  управляется посредством текстов функционально и предметно ограниченного естественного языка (которые дальше будут называться "входными текстами"). Механизм порождения этих текстов считается неизвестным. Обозначим через  $T_0$  множество осмысленных входных текстов, т.е. текстов, задающих осуществимые преобразования  $\Phi_0$ . Язык действий  $D_0$  определим как инструмент для преобразования системы  $\Phi_0$  посредством текстов из  $T_0$ . Не касаясь пока подробностей устройства языков действий, рассмотрим, как язык  $D_0$  осуществляет эту свою функцию.

Определим естественный язык как систему со следующими свойствами.

А. Язык есть средство преобразования действительно - сти  $M$  (в данном случае мира, состоящего из  $\Phi_0$ ,  $T_0$  и  $D_0$ ).

В. Любое применение языка является его *самоизменением*.

С. Все выражения языка интерпретируются как *императивы*: (Это не значит, что мы требуем, чтобы все выражения входного языка имели форму императивов в смысле традиционной грамматики русского языка. Но все выражения этого языка, независимо от их грамматической формы, мы будем *понимать* как императивы.)

Поясним такое определение языка на примере  $D_0$ .

Назовем сначала те языки, о которых будет идти речь в статье: входной естественный язык (в нашем примере - язык, на котором написаны тексты  $T_0$ ); язык действий (осуществляющий преобразования  $\Phi_0$  посредством текстов  $T_0$ ) и язык описания преобразований (язык  $G_0$ , посредством которого задается сам язык действий). Перечисленные три свойства считаются присущими только первым двум из названных языков. Механизм реализации этих свойств во входном языке в данной работе не обсуждается, хотя способ представления их в языках действий и может рассматриваться как сильно упрощенный формальный аналог такой реализации.

Язык действий будем задавать: 1) стратегиями своего собственного преобразования (описанными в нашем примере на языке  $G_0$ ); 2) средством перевода описаний этих стратегий с входного языка на  $G_0$ ; 3) средством дополнения стратегий до описаний конкретных преобразований; 4) средствами реализации описанных преобразований в  $M$  (сюда включается и средство взаимосвязи между  $D_0$  и  $\Phi_0$ ).

Будем считать, что каждый входной текст  $t \in T_0$  задает некоторую *стратегию* преобразования  $M$ . Например, тексту  $t_1 =$  "построить синий квадрат" поставим в соответствие стратегию, описываемую некоторым выражением  $\varphi_1$  языка  $G_0$  и предписывающую изменить  $\Phi_0$ , построив в нем синий квадрат произвольного размера и расположения. Рассмотрим, как устроена эта стратегия. Любое преобразование мира  $M$ , происходящее в результате применения  $D_0$ , начинается с изменения языком  $D_0$  самого себя. Это изменение может касаться всех составных частей  $D_0$  (кроме средства реализации преобразований), но при использовании  $D_0$  в режиме управления  $\Phi_0$  оно чаще всего касается формального представления о системе  $\Phi_0$ , встроенного в  $D_0$ . Например, текст  $t_1$  изменит область определения функции *квадрат* языка  $G_0$ , добавив к ней еще один элемент - имя построенного синего квадрата.

Дополнение стратегии, заданной формулой  $\varphi_1$ , определим как выбор конкретных характеристик размера и расположения квадрата и нахождение всех следствий построения такой фигуры (например, некоторые из уже имеющихся на экране фигур могут перестать быть "видимыми"). Для перевода описания стратегии с входного языка на  $G_0$  будем использовать грамматику (построенную по принципу категориальных грамматик) с условиями на применимость правил. Будем считать, что средство реализации преобразований в  $M$  транслирует доопределенную формулу  $\varphi_1$  на некоторый язык программирования и выполняет полученную машинную программу.

Преобразование языка  $D_0$ , описанное доопределенной формулой  $\Phi_1^1$ , реально осуществляется посредством этой программы. Преобразования системы  $\Phi_0$  (изменение изображения на экране) определим как следствия преобразований  $D_0$ , осуществляемые средствами взаимосвязи между  $D_0$  и  $\Phi_0$  (в данном случае - тоже машинной программой).

Преобразование языка  $D_0$  может иметь следствием и изменение набора осмысленных входных текстов  $T_0$ . Например, если в  $D_0$  не определено понятие времени, то применение текста  $t_2 =$  "уничтожить все синие квадраты" исключает из  $T_0$  все выражения, предписывающие какие-либо перемещения синих квадратов. В результате применения текста  $t_1$  в  $T_0$  появятся тексты, в которых используются выражения "этот квадрат", "построенный квадрат", "синий квадрат" и т.д.

Все выражения языка интерпретируются как императивы (не только имена действий и процессов, но и имена объектов и ситуаций понимаются как императивы). Императивы подразделяются на императивы поиска (создания), построения, выбора, преобразования. Например, тексту "синий квадрат" будет соответствовать в  $D_0$  формула вида (синий  $\bullet$  квадрат) ( $\rho$ ), где  $\rho$  - символ неопределенного объекта, " $\bullet$ " - знак суперпозиции функций. Семантика выражений  $\rho$ , квадрат( $\rho$ ), синий( $x$ ) неформально может быть передана соответственно предложениями: "построить произвольную геометрическую фигуру" (назовем эту семантику "процесс  $\rho$ "), "преобразовать процесс  $\rho$  таким образом, чтобы он порождал только квадраты" ("процесс  $x$ "), "преобразовать процесс  $x$  таким образом, чтобы он порождал только синие квадраты" (в языке  $D_0$  эти интерпретации задаются посредством последовательностей императивов),

2. Строение языков действий. Пусть  $\Phi$  - управляемая динамическая система,  $T$  - множество осмысленных текстов, посредством которых можно управлять  $\Phi$ ,  $D$  - язык действий, осуществляющий такое управление. Язык действий  $D$  определим как тройку

$\langle L, M, Q \rangle$ , где  $L$  - средство построения стратегий преобразования  $D$ ;  $M$  - формальное описание  $\Phi$  (которое будет называться "мир  $M$ ");  $Q$  - средство построения программы, реализующей нужные преобразования  $M$ .

Мир  $M$  будем описывать посредством внутреннего языка  $G$ , задаваемого типовым исчислением предикатов первого порядка с условиями на применимость правил вывода;  $B$  - базовый мир - многосортная модель той же сигнатуры, что и  $G$ ;  $I$  - средство построения интерпретаций формул  $G$  на  $B$ .

В языке  $G$  выделим следующие типы формул: объекты, действия, модификаторы, ситуации, процессы. Каждый тип может подразделяться на категории, функциональная нагрузка которых состоит в определении условий применимости правил вывода в  $G$ .

Множество категорий должно, по возможности, обеспечивать однозначное решение вопроса о том, какое правило вывода выбрать на каждом данном этапе вывода формулы в  $G$ .

Категории будем подразделять на: семантические и грамматические (по которым можно выяснить соответственно семантическую или грамматическую согласованность двух заданных формул); ситуационные, предназначенные для установления места построенных объектов в ситуации реализации действия, например, подстановка фактических параметров вместо формальных или отождествление разных (встречающихся в разных частях входного текста) наименований одного и того же объекта (модификатора, ситуации, действия, процесса); программные (указывает значения параметров и точку входа в программу, строящую семантику формул  $D$ ).

Например, для того, чтобы в  $G$  было выводимо выражение (синий  $\bullet$  квадрат) ( $\rho$ ), соответствующее тексту "синий квадрат"  $\in T_0$ , необходимо, чтобы семантические и грамматические категории, приписанные элементам *синий* и *квадрат*,

были согласованы друг с другом (должны совпадать категории падежа, числа, рода и т.д.).

При каждом применении язык  $D$  изменяет самого себя. Процедурь самоизменения языка  $D$  могут касаться всех его составных частей, кроме  $Q$ , и задаются посредством императивов указанных четырех видов. В случае изменения  $B$  эти императивы определим на множествах объектов и действий  $B$ , в случае изменения  $G$  - на множествах стратегий вывода формул  $G$ , в случаях изменения  $L$  и  $I$  - на множествах правил анализа входного текста или интерпретации формул  $G$  соответственно. Например, в случае изменения  $G$  под выполнением императива поиска может пониматься поиск последовательностей правил вывода  $G$ , применимых к заданной формуле  $\varphi$ , а под выполнением императива создания - добавление к  $G$  новых правил вывода, позволяющих вывести из заданной последовательности формул  $G$  некоторую формулу, ранее не выводимую в  $G$  (множество выводимых в  $G$  формул в первом случае не меняется, во втором - расширяется).

Для иллюстрации таких преобразований приведем два примера более сложных способов самоизменения  $D$ .

1. Рассмотрим текст  $t_3 = \text{"построить синий квадрат. Этот квадрат передвинуть влево"}$ . Функциональный символ *этот* интерпретируется как императив поиска, определенный на правилах вывода имен объектов. Выполнение этого императива состоит в поиске применимого правила вывода вида  $\text{квадрат}(\rho)$ ,  $x = \varphi \vdash \text{квадрат}(\rho) = x / U$ , где  $\varphi$  - выражение, входящее в уже построенную часть формулы языка  $G_0$ , сопоставляемой тексту  $t_3$ ;  $U$  - условие применимости правила вывода, включающее проверку грамматической и семантической согласованности выражений  $\varphi$  и  $\text{квадрат}(\rho)$ ;  $x$  - имя конкретного квадрата, построенного в результате выполнения семантики выражения (*синий*  $\bullet$  *квадрат*)  $(\rho)$ .

2. Рассмотрим текст  $t_4 = \text{"построить треугольник треугольников"}$ . Пусть выражение "треугольник треугольников" не встреча-

лось ранее в  $T_0$ , и в  $G_0$  не выводима формула, задающая стратегию преобразования, соответствующую этому выражению. Пусть в  $D_0$  формально заданы: 1) понятие треугольника, описываемое следующим процессом: "найти три точки, не лежащие на одной прямой, и соединить их попарно отрезками прямых линий"; 2) категории "единичный объект" и "множественный объект"; 3) средства обнаружения грамматических указателей принадлежности одного объекта (объектов) другому объекту (например, распознавание родительного падежа в выражении "множество точек"). В ответ на использование текста  $t_3$  в качестве входного будет сделана попытка добавить его к  $T_0$ , т.е. проанализировать как осмысленный. Можно построить такой вариант формальной реализации языков действий, который позволит "понять" выражение "треугольник треугольников" по аналогии с выражением "множество точек", вследствие чего будет построен треугольник, содержащий внутри себя другие треугольники. В результате преобразования  $D_0$  посредством  $t_3$  в  $D_0$  будет создана новая стратегия, описываемая посредством формулы  $\varphi$ , ранее не выводимой в  $D_0$ . Иначе можно сказать так: если раньше в  $D_0$  была выводима формула  $\varphi_0 = \text{треугольник}(\rho)$ , где  $\rho$  - символ неопределенного объекта (относящийся к категории "единичный объект"), а *треугольник* - функциональный символ в  $G_0$  (язык описания  $D_0$ ), то теперь будет выводима еще и формула  $\hat{\varphi}_1 = \text{треугольник}(\mu)$ , где  $\mu$  - символ неопределенного множества объектов (относящийся к категории "множественный объект").

Сложность входных текстов языка действий  $D$  зависит от сложности "самоизменения" этого языка в ходе его использования. Один из более простых вариантов "самоизменения" затрагивает только  $B$ . Входными текстами в этом случае являются простые полные предложения, не связанные между собой. Интересно было бы рассмотреть более сложные варианты "самоизменения"  $D$ , включающие, помимо преобразований  $B$ , преобразования  $G$  (входные

тексты при этом могут содержать связанные между собой предложения, метафоры, которые трактуются в D соответственно как императивы поиска вывода или императивы создания вывода), однако это сильно усложнило бы изложение, вызвав необходимость введения метауровня. Поэтому ниже описывается более простой вариант языка действий (формальная реализация которого также вызывает немало сложностей), в котором предполагается, что 1) "самоизменение" D затрагивает только B; 2) формулы G удовлетворяют условию композициональности, т.е. для любых формул  $\phi$  и  $\psi$  таких, что  $\phi$  входит в  $\psi$ , семантика  $\phi$  не зависит от  $\psi$ . Например, в случае применения текста  $t_1$  интерпретация формулы  $\phi = \text{квадрат}(\rho)$  определяется независимо от формулы  $\psi = (\text{синий} \bullet \text{квадрат})(\rho)$ . Затем, в результате применения функционального символа *синий* к формуле  $\phi$ , эта интерпретация изменяется.

Язык действий D будем строить как систему, включающую две изолированные части: G и B.

Множество формул, выводимых в G, определяет допустимые изменения B путем построения *расширений* базового мира B. Мир B можно рассматривать как множество понятий (заданных генеративно) объектов и действий, а расширение  $B_1$  - построение в B некоторой ситуации, состоящей из имен, присвоенных конкретным объектам и действиям, подходящим под некоторые из уже имеющих понятий B. Например, формула, соответствующая тексту  $t_1$ , интерпретируется как процедура порождения произвольного синего квадрата. Расширение  $B_1$ , получаемое в результате применения этого текста, заключается в добавлении к B ситуации s, состоящей из единственного имени x, присвоенного построенному квадрату в результате выполнения семантики формулы  $(\text{синий} \bullet \text{квадрат})(\rho)$ . Каждое преобразование B изменяет не все текущее состояние  $B_1$  полностью, а его фрагмент - некоторую ситуацию - последовательность описаний объектов. Мир B заранее не рас -



членен на ситуации применения действий. Эти ситуации строятся в результате интерпретации стратегий вывода формул в  $G$ , задаваемых текстом на входном языке.

Каждое конкретное расширение  $B_i$  строится в результате интерпретации вывода некоторой формулы языка  $D$  и называется состоянием мира  $B$ . Входной текст  $T$  (на естественном языке) не задает однозначно преобразование  $B$ , которое надо выполнить. Назначение входного языка - назвать процесс, т.е. задать стратегию вывода в  $G$ , определяющую стратегию преобразования  $B$  - перевода  $B$  в некоторое состояние  $B_i$ . Назначение  $B$  - доопределить это название, т.е. доказать выполнимость/невыполнимость такого процесса в  $B$ . Доопределение стратегии до описания конкретного способа перехода  $B$  в  $B_i$  производится средствами самого  $B$  без обращения к тексту  $T$  (т.е.  $B$  обладает некоторой самостоятельностью относительно  $D$ ).

Базовый мир  $B$  задан *генеративно* и описывает, как система  $\Phi$  могла бы развиваться сама по себе, не будь к ней никаких обращений на входном языке. В этом случае мир  $B$  развивается недетерминированно. Использование входного текста дает возможность задавать (через формулы  $G$ ) определенные стратегии развития  $B$ , которые затем реализуются в  $B$  уже по его собственным законам развития, независимо от входного текста (выбор способа реализации заданной стратегии может быть, в том числе, и случайным). Например, в языке  $G_0$  строится формула  $\phi = (сидит \bullet квадрат)(p)$ , а выбор конкретного квадрата, который будет строиться, осуществляется базовым миром  $B_0$  независимо от  $G_0$ . Автономное развитие  $B_0$  (не обусловленное применением входных текстов) состоит в произвольном построении, удалении и перемещении геометрических фигур согласно возможностям, определенным в  $B_0$  и независимо от  $G_0$ .

Автономное развитие  $B$  используется в режиме обучения языку  $D$ , развитие  $B$  через формулы  $G$  - в режиме управления посредством  $D$ .

Вследствие указанной самостоятельности  $B$  относительно  $G$  семантика формул  $\varphi \in G$  двухступенчата:  $I$  формирует последовательность императивов, соответствующих  $\varphi$ ,  $B$  дополняет эту последовательность указанием конкретного способа и порядка выполнения этих императивов (решение этих вопросов не зависит от  $I$  и  $G$ ). Из-за такой двухступенчатости семантики  $G$  каждый из типов формул "объект", "ситуация", "действие", "модификатор", "процесс" означает и императив, и преобразование, определенное на  $\mathcal{M}$ .

Например, имя объекта  $\alpha$  - одноместная функция, интерпретируемая посредством составного императива, состоящего из: императива поиска (создания) всех тех и только тех объектов из  $B$ , которые соответствуют указанному имени; императива построения множества, состоящего из найденных объектов; императива выбора представителя из этого множества; императива включения выбранного представителя в текущую ситуацию. (Так как объекты в  $B$  заданы генеративно, то выполнению императива поиска соответствует нахождение способа построения процедуры генерации объектов с заданным именем; выполнению императива построения - построение этой процедуры генерации; выполнению императива выбора - выбор конкретного способа генерации.) Например, выражение  $\varphi$  языка  $D$  может содержать функцию  $y = \text{стол}(x)$ , интерпретирующуюся так: "найти объекты, входящие в множество объектов  $x$  и называемые "столами"; построить множество, состоящее из всех таких и только таких объектов; выбрать из полученного множества один объект, назвать этот объект "у" и включить в текущую ситуацию."

В  $D$  выделяются специальные имена для обозначения неопределенных объектов, ситуаций, действий, процессов, в результате интерпретации которых в  $B$  могут быть построены любой объект, любая ситуация, любое действие, любой процесс соответственно. Например, семантикой символа неопределенного объекта  $\rho$  языка  $G_0$

является процесс построения произвольной геометрической фигуры.

Среди функциональных символов языка  $G$  выделяется особый класс - модификаторы - функции, использующиеся для конструирования сложных имен объектов, ситуаций, действий и процессов (во входном языке модификаторы могут отображаться, например, посредством прилагательных, союзов, предлогов, наречий: "на", "слева", "справа", "после", "большой", "осторожно", "быстро", "и", "или").

В  $G$  все многоместные модификаторы сводятся к одноместным, например, функция на ( $x, y$ ) заменяется набором одноместных функций вида на <sub>$y$</sub>  ( $x$ ) для фиксированного  $y$ . При таком способе представления выражению естественного языка "книги, лежащие на столе" будет соответствовать следующая формула языка  $G$ : ( $x = \text{стол}(P)$ ) & ( $y = (\text{на}_x \bullet \text{книга})(P)$ ), где " $\bullet$ " - знак суперпозиции функций,  $P$  - понятие "объект" в  $B$ .

Мир  $B$  организован как система модулей, каждый из которых представляет собой недетерминированный процесс порождения новых состояний  $B$ . Эти процессы называются дальше внутренними процессами (чтобы отличать их от названия одного из типов формул языка  $G$ ). Например, имени объекта *квадрат* из языка  $G_0$  в  $B_0$  сопоставляется внутренний процесс, порождения произвольного квадрата. Ситуация - последовательность внутренних процессов, соответствующих именам объектов  $G$ .

Текст на естественном языке, посредством которого мы хотим что-то изменить в  $m$ , описывает в общем случае три вещи: ситуацию  $s_1$  текущего состояния  $B_i$ , ситуацию  $s_2$  нового состояния  $B_j$ , способ перехода от  $s_1$  к  $s_2$ . Процесс, соответствующий входному тексту, определяется посредством последовательности формул вида  $s_1 \cdot f(s_1) \rightarrow s_2$  ("построить ситуацию  $s_1$ , применить к ней преобразование  $f$ , в результате получить ситуацию

$s_2$ ). Неопределенными могут быть любые компоненты этого выражения ( $s_1$ ,  $s_2$ ,  $f$ ) в любых сочетаниях.

Одной из основных процедур при анализе текста является процедура выбора (основывающаяся на некоторых заданных отношениях предпочтения). Первоначально, до анализа текста  $t$ , считается, что  $t$  задает полностью неопределенный процесс  $\alpha$ , в котором ни одна из компонент  $s_1, s_2, f$  не задана. (Любая допустимая последовательность преобразований в  $\mathcal{M}$  может быть получена как результат построения семантики  $\alpha$ .) По мере анализа текста  $t$ , это множество сужается за счет использования функциональных символов  $G$ , состав и порядок применения которых задаются самим текстом  $t$ , и за счет доопределения получаемых формул.

3. Анализ входного текста. Средство анализа входного текста (понимаемое в данной работе как средство построения стратегий вывода) разработано без использования других методов формализации естественного языка, хотя по самым общим характеристикам его можно соотнести со следующими тремя направлениями: категориальными грамматиками [4], унификационными грамматиками [5], Gapping Grammars [5]. Ниже описаны те характеристики указанных направлений, которые свойственны также и  $L$ , но формально реализованы в  $L$  иначе, чем в перечисленных работах.

Три основных свойства категориальных грамматик.

А. *Структура типа "функция-аргумент"* как средство описания текста. Каждое слово входного текста понимается как функция, преобразующая построенное к тому времени описание семантики уже прочитанной части текста в некоторое другое описание.

В. *Функциональные основания выбора системы категорий.* Любой языковый анализатор (если он не является прямым одношаговым транслятором с естественного языка на формальный, не структурирующим входной текст), содержит два основных ком -

понента: систему категорий и систему правил переписывания. Существует альтернатива: где сконцентрировать основную комбинаторную сложность - в правилах переписывания или в системе категорий, в терминах которых эти правила строятся. Второй способ удобнее, так как система категорий легче обозрима, чем система правил, кроме того, она имеет ясное содержательное основание: категории формируются по принципу общности функциональных свойств относящихся к ним элементов.

С. Композициональность процедур перевода текста на логический язык важна для того, чтобы обойтись без механизма возврата в цепочке вывода формулы D и без перестройки этого вывода (что вызывает серьезные вычислительные сложности).

Композициональность оказывается возможной благодаря интерпретации формул D как *процедур сужения выбора*. Семантика большой формулы есть сужение множеств процессов, из которых предписывается производить выбор семантиками входящих в нее подформул.

Граматики типа Gapping Grammars допускают объединение в одну формулу частей текста, не являющихся смежными. Это удобно в силу несовпадения линейной структуры текста с его грамматической структурой. Это несовпадение приводит к тому, что грамматически связанные части текста удалены друг от друга пространственно. Семантика такого "разорванного" текста строится как результат унификации семантик, сопоставленных каждой из частей. Унификация здесь понимается как задача выбора общей части заданных конечных множеств, реализуемая посредством переборных процедур, определенных на графовых структурах.

4. Способы языкового взаимодействия в рамках языка D. Язык D может использоваться в двух режимах: *управления* (приведенное выше описание D есть описание именно этого режима) и *обучения*. Содержательно эти режимы не являются четко разделенными,

так как согласно предлагаемой концепции языка, всякое использование языка, помимо преобразования мира  $\Phi$  (которое не обязательно имеет место), включает в качестве обязательной составной части изменение самого языка. Поэтому всякое управление объектом  $\Phi$  есть одновременно и обучение (точнее, переобучение) языку  $D$ . С другой стороны, всякое обучение  $D$  (средствами, специально предназначенными только для обучения) сказывается в дальнейшем на возможностях управления объектом  $\Phi$ .

Исходным для обучения (если язык  $D$  еще не начинал строиться для данной предметной области) является базовый мир  $B$ , состоящий из двух частей:

- 1) понятия о неопределенном объекте  $\rho$  мира  $B$ ,
- 2) понятия о неопределенном процессе  $\delta$  мира  $B$ ,

где  $\rho$  порождает произвольный объект мира  $B$  в соответствии с некоторым отношением предпочтения, определяющим выбор такого элемента в  $B$  (например, в соответствии с вероятностями объектов и процессов в  $B$ ),  $\delta$  порождает произвольный процесс мира  $B$ . Преобразования, происходящие в таком  $B$ , определяются произвольными последовательностями результатов выполнения внутренних процессов, соответствующих  $\rho$  и  $\delta$ . В этом случае мир  $B$  изменяется максимально недетерминированно, и входной (естественный) язык может управлять им только посредством текстов вида "сделать что-нибудь".

Дальнейшее обучение состоит во введении средств, позволяющих производить более детерминированные преобразования. Это осуществляется путем структурирования понятий  $\rho$  и  $\delta$  (последовательного введения новых, более узких, подпонятий) посредством изменения синтаксиса и семантики языка  $D$  в диалоге с пользователем. Диалог может быть в двух вариантах: в режиме управления  $\Phi$  и в режиме автономного функционирования  $\Phi$ . Повышение "управляемости" мира  $B$  в обоих режимах идет за счет введения

новых средств *наименования*. Осуществляется это в указанных режимах по-разному.

Первый режим требует явного определения вносимых изменений путем формирования соответствующих текстов из Т.

При втором режиме могут (но не обязательно) задаваться только имена интересующих пользователя объектов или действий (тех объектов или действий, внутри которых необходимо выделить подпонятия или переопределить их каким-либо другим образом). Мир В развивается автономно, независимо от входного текста. В ходе последовательной генерации таких преобразований могут появляться объекты (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). Возможность такого названия предусмотрена в диалоге, соответствующем режиму автономного функционирования.

Режим обучения языку действий D фактически является таким вариантом языка, при котором управляемая система Ф совпадает с самим языком D. Возможен и случай, когда Ф совпадает с Т. Однако оба эти варианта языков действий (совпадение Ф с D или с Т) будут обладать своими особенностями. Описание формального способа реализации языков действий, ориентированных на управление внешним объектом Ф, составит предмет последующей публикации.

## Л и т е р а т у р а

1. ТИМОФЕЕВА М.К. Проект построения языка действий //Методы обработки символьных последовательностей и сигналов. - Новосибирск, 1990. - Вып. 132: Вычислительные системы. - С. 115-133.
2. BARWISE J. Situations and Attitudes. - Cambridge: MIT press. - 1983. - 352 p.

3. ГОНЧАРОВ С.С., СВИРИДЕНКО Д.И.  $\Sigma$ -программирование //Логико-математические основы проблемы МОЗ. - Новосибирск, 1985. - Вып. 101: Вычислительные системы. - С. 3-29.

4. МОНТЕГЮ Р. Универсальная грамматика //Семантика и информатика. 1985.-№ 26. - С. 105-136.

5. SAINT-DIZIER P. An Approach to Natural Language Semantics in Logic Programming //The Journal of Logic Programming.- 1986. - Vol. 3, N 4. - P. 329-356.

Поступила в ред.-изд.отд.

26 июня 1991 года