

ПРОЕКТ ПОСТРОЕНИЯ ЯЗЫКА ДЕЙСТВИЙ

М.К. Тимофеева

Моделирование естественного языкового взаимодействия с ЭВМ для решения задач управления сложным динамическим объектом вызывает необходимость построения определенного класса языков - языков действий. Тексты языка действий по внешней форме совпадают с текстами предметно-ограниченного естественного языка, синтаксис и семантика его формальны. Почему строится свой язык, а не используются уже имеющиеся? Здесь - три основные причины.

1. Языка, в точности ориентированного на решение нужных задач, нет.

2. Имеющиеся языки слишком "привязаны" к традиционным структурным моделям естественного языка. Во многих работах истинный интерес сосредоточен либо на формализации естественного языка, либо на рассмотрении принципов его строения как источника для постановки новых задач в области формальной логики. В данной работе естественный язык рассматривается не как самоцель, а как средство решения другой задачи - управления сложным динамическим объектом.

3. Выбор подходящей модели из числа существующих [1] трудно осуществим главным образом из-за отсутствия метатеории, позволяющей проводить сравнение качества функционирования разных моделей.

Построение языка действий, хотя непосредственно и не использует какой-либо из существующих подходов к формализации естественного языкового взаимодействия, но опирается на некоторые общие принципы таких направлений, как ситуационная семантика [2], ситуационное исчисление [3], категориальные грамматики [4], Gapping Grammars [5]. В плане реализации предлагаемый подход ориентирован на язык Σ -программирования [6]. Основные принципы анализа текста иллюстрируются примером в конце статьи.

1. Строение языка действий

Язык действий предназначен для управления динамическим объектом Φ (изменение состояния Φ или формального представления модели мира M для Φ) посредством применения императивных предложений естественного языка. Входной текст может содержать только одно предложение в повелительном наклонении. В этом предложении могут использоваться определения, дополнения, причастные и деепричастные обороты, сочинительные придаточные предложения. Могут использоваться местоимения. Лексика определяется предметной областью. Семантика языка действий задается формальной моделью мира M , описывающей допустимые способы изменения Φ как расширения некоторого базового мира m_0 . Каждое конкретное расширение называется состоянием мира M . В любой момент мир M находится в одном из своих состояний.

Основные понятия: M -объекты, действия, процессы. Действия переводят M из одного состояния в другое (одно расширение мира m_0 в другое). Объекты M рассматриваются только относительно действий, и наоборот: объектом M является только то, что может подвергнуться какому-либо действию; любое действие должно быть применимо хотя бы к одному объекту M .

В ходе языкового управления понятия об одних объектах могут исчезать, о других - появляться, но предполагается, что существует некоторое базовое элементарное множество объектов, которое остается неизменным и в терминах которого можно описать любой объект M . То же верно и для действий. Базовые множества объектов и действий содержатся в \mathcal{M}_0 . Появление в M понятий о новых объектах или действиях, не описываемых в терминах соответствующих базовых множеств, возможно только в результате возникновения новых объектов или действий в мире Φ и осуществляется вне системы естественного языкового взаимодействия. В \mathcal{M}_0 входят описания тех объектов и действий, каждое из которых именуется в естественном языке одним словом. Расширение \mathcal{M}_1 мира \mathcal{M}_0 состоит из \mathcal{M}_0 , дополненного описаниями объектов и действий, имеющих во входных естественных языковых текстах сложные наименования, построенные с использованием определений, дополнений и т.д. Каждое действие изменяет не все текущее состояние \mathcal{M}_1 полностью, а только его фрагмент - некоторую ситуацию, являющуюся подмножеством \mathcal{M}_1 . Мир \mathcal{M}_1 заранее не расчленен на ситуации применения действий, эти ситуации задаются входным текстом на естественном языке.

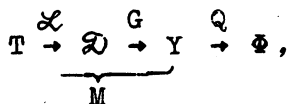
Входным текстом не определяется однозначно последовательность действий, которую нужно выполнить на \mathcal{M}_1 , но описываются только процессы - недоопределенные последовательности действий (т.е. множества последовательностей действий), которые, реализуясь в \mathcal{M}_1 , доопределяются уже по законам мира M , а не по входному тексту на естественном языке. Недоопределенность может касаться разных моментов. Может быть задана только цель (конечный результат) процесса, могут быть только начальные действия, может быть частично задана стратегия, может быть задана локация (пространственная или временная).

Текст на естественном языке, посредством которого мы хотим что-то изменить в \mathcal{M}_1 или Φ , описывает в общем слу-

чае три вещи: ситуацию S_1 текущего состояния мира \mathcal{M}_1 , ситуацию S_2 нового состояния \mathcal{M}_2 , способ перехода f от S_1 к S_2 . Процесс, соответствующий входному тексту, описывается формулой вида $S_1 f(S_1) \rightarrow S_2$ ("построить ситуацию S_1 , применить к ней преобразование f , в результате получить ситуацию S_2 "). Недоопределенными могут быть любые компоненты этого выражения (S_1 , S_2 и f) в любых сочетаниях.

Объекты, действия, процессы, именуемые текстом на естественном языке, есть множества. Объект - множество элементов из \mathcal{M}_0 , действие - множество способов перехода из одного состояния M в другое, процесс - множество последовательностей действий. Каждое из этих множеств содержит элементы, тождественные относительно необходимого изменения M или Φ . Реализация описанного входным текстом процесса всегда включает выбор представителей множеств объектов и действий. Этот выбор обусловливается уже не входным текстом, а производится по законам развития M . Для этого предполагается специальный аппарат выбора, опирающийся на оценки различных вариантов с позиции некоторого отношения предпочтения (причинно-следственные связи между действиями, сложность реализации действий и т.д.) на M .

Схема управления посредством языка действий такова:



где T - входной текст на естественном языке; \mathcal{D} - язык описания процессов; \mathcal{L} - текстовый анализатор, переводящий входной текст в выражение из \mathcal{D} ; Y - язык описания действий; G - преобразователь, доопределяющий процессы на \mathcal{M}_1 , т.е. переводящий их в последовательности действий (если таковое возможно); Φ - управляемый объект; Q - базовая система программ, задающая базовые объекты и действия.

Возможны четыре уровня "понимания" входного текста (обрывающие указанную цепочку преобразований в разных местах): 1) проверить, выполним ли описанный текстом T процесс G на \mathcal{M}_1 (т.е. доопределим ли он на \mathcal{M}_1); 2) доопределить процесс G на \mathcal{M}_1 ; 3) выполнить G на \mathcal{M}_1 ; 4) выполнить G на Φ .

Анализ входного текста основан на двух классах операций: 1) описания допустимых множеств, 2) выбора представителей из каждого множества. Операции описания и выбора чередуются в ходе анализа. Каждый из использующихся уровней представления $T(T, \mathcal{D}, Y)$ интерпретируется двумя способами: 1) декларативно - как описание множеств объектов и действий (в T, \mathcal{D}, Y); 2) процедурно - посредством реализации процедур выбора (\mathcal{X} выбирает соответствующую T стратегию вывода в \mathcal{D} , приводящую к некоторой формуле Φ , которая описывает множества объектов и действий в \mathcal{D} ; G выбирает для Φ соответствующую формулу Ψ из Y ; Q выбирает программную реализацию Ψ).

2. Язык описания процессов

Естественный язык не задает однозначно последовательность действий на \mathcal{M}_1 , которую надо выполнить. Назначение языка - назвать процесс. Назначение M - доопределить этот процесс, т.е. доказать выполнимость такого процесса в \mathcal{M}_1 и, проинтерпретировав доопределенную логическую формулу процесса процедурно, построить алгоритм его реализации в терминах базовых объектов и действий. Назначение Q - выполнить процедуру, обозначенные именами базовых действий, так, как это описано в формуле доопределенного процесса.

Анализ входного текста основан на принципе выбора. Первоначально, до анализа текста T , считается, что текстом T задается полностью неопределенный процесс σ_T , в котором ни од-

на из компонент S_1, S_2, f не задана. Множество последовательностей действий \mathcal{A} включает любые допустимые в M по следовательности действий. Затем, по мере анализа текста T , это множество сужается за счет использования анализатором \mathcal{L} специальных функций - модификаторов, состав и последовательность применения которых задается самим текстом T .

Язык описания процессов \mathcal{D} строится как исчисление формул, описывающееся в следующей сигнатуре: $\langle P_0, D_0, Mod, X, C, N, \mathcal{A}, \delta, \rho, s, \sigma \rangle$, где $P_0 \cup D_0 \cup Mod$ - множество функциональных символов, $Mod = Mod_0 \cup Mod_d$; P_0, Mod - символы одноместных функций, обозначающих соответственно имена базовых объектов и модификаторы объектов (Mod_0) или действий (Mod_d); D_0 - имена базовых действий; X - множество символов переменных; C - множество константных символов, включающее: 1) индивидуальные имена объектов Φ , не встречающиеся при наименовании других объектов Φ , например, "куб № 1" (такие имена могут быть не у всех объектов); 2) имя P множества всех объектов Φ ; N - множество натуральных чисел; $\mathcal{A}, \delta, \rho, s$ - соответственно множества символов неопределенных процессов, действий, объектов, ситуаций; σ - множество символов пропозициональных связок, кванторы и символы "=", " \neg ", " \rightarrow ", " \leftrightarrow ", " \forall ", " \exists " (суперпозиция функций). Множество Mod_0 замкнуто относительно операции " \circ ".

Множество \mathcal{P} объектов языка \mathcal{D} :

- 1) $C \subseteq \mathcal{P}$, $\rho \in \mathcal{P}$;
- 2) для любых $a \in P_0$ и $x \in XU\{P\}$ выражение $a(x)$ принадлежит \mathcal{P} ;
- 3) для любых $a \in P_0$, $x \in XU\{P\}$ и $\pi \in Mod_0$ выражение $(\pi \circ a)(x)$ принадлежит \mathcal{P} ;
- 4) для любых $P_1, P_2 \in \mathcal{P}$ выражения $P_1 \vee P_2$, $P_1 \& P_2$, принадлежат \mathcal{P} ;

5) для любых $a \in P_0$, $m_1, \dots, m_u \in \text{Mod}_0$ и $x \in XU\{P\}$ выражение $(\Box a)(x)$ принадлежит \mathcal{P} ($m = m_1 \circ \dots \circ m_u$);

6) для любого $p \in \mathcal{P}$ выражение $\Box p$ принадлежит \mathcal{P} ;

7) других объектов в \mathcal{P} нет.

Обозначим через $V(p)$, где $p \in \mathcal{P}$, множество всех символов из P_0 , встречающихся в выражении p . Верхний индекс при имени функции будет означать ее арность. Выражение $O(x_1, \dots, x_n) = ((x_1 = p_1) \dots (x_n = p_n))$ обозначает ситуацию в M , $x_i \in X$, $p_i \in \mathcal{P}$, $1 \leq i \leq n$.

Множество процессов языка \mathcal{D} :

1) для любых $d^n \in D_0$ и $p_1, \dots, p_n \in \mathcal{P}$ выражение $d^n(p_1, \dots, p_n)$ принадлежит \mathcal{D}_0 ;

2) для любых $g^n \in \mathcal{D}_0$ и $m \in \text{Mod}_d$ выражение $m(g^n)$ принадлежит \mathcal{D}_0 ;

3) для любых $g_1^{n_1}, g_2^{n_2} \in \mathcal{D}_0$ выражения $g_1^{n_1} \vee g_2^{n_2}$, $g_1^{n_1} \& g_2^{n_2}$ принадлежат \mathcal{D}_0 ;

4) для любого $g^n \in \mathcal{D}_0$ выражение $\Box g^n$ принадлежит \mathcal{D}_0 ;

5) $\delta \subseteq \mathcal{D}_0$;

6) для любых $g^n \in \mathcal{D}_0$, $p_1, \dots, p_n, p'_1, \dots, p'_v \in \mathcal{P}$ и $x_1, \dots, x_n \in X$ следующее выражение принадлежит \mathcal{D} :

$$O_1(x_1, \dots, x_n) \\ \varphi = \exists x_1, \dots, x_n \left[\overbrace{((x_1 = p_1) \& \dots \& (x_n = p_n))}^{O_1(x_1, \dots, x_n)} \right] \theta_{t_1} x_{t_1}, \dots \\ \dots, \theta_{t_k} x_{t_k} g^n(x_{i_1}, \dots, x_{i_r}) \rightarrow$$

$$O_2(x_{j_1}, \dots, x_{j_V}) \\ \rightarrow \overbrace{(x_{j_1} = p'_1) \& \dots \& (x_{j_V} = p'_V)},$$

где $\{x_{i_1}, \dots, x_{i_r}\}, \{x_{t_1}, \dots, x_{t_k}\}, \{x_{j_1}, \dots, x_{j_V}\}$ - подмножества множества $\{x_1, \dots, x_n\}$, θ_i - либо квантор всеобщности, либо натуральное число, указывающее число элементов, которые нужно выбрать из множества x_i (если должен быть выбран только один элемент, то соответствующая пара $\theta_i x_i$ в Φ отсутствует); $r \leq n$, так как некоторые объекты из p_1, \dots, p_n могут не использоваться действием g^n , а служить только для нахождения других объектов из p_1, \dots, p_n ;

7) $ox \in \mathcal{D}$;

8) других процессов в \mathcal{D} нет.

Любая формула из \mathcal{P} - определение объекта, любая формула из \mathcal{D} - определение процесса. Расширение \mathcal{M}_1 базового мира \mathcal{M}_0 состоит в добавлении к \mathcal{M}_0 любого подмножества формул из \mathcal{P} или \mathcal{D} . Фиксированность базового мира \mathcal{M}_0 означает, что сигнатура языка описания процессов \mathcal{D} не меняется в ходе естественного языкового взаимодействия.

Среди функциональных символов языка \mathcal{D} выделяется особый класс - модификаторы - имена функций, переводящих одно множество в другое и позволяющих конструировать новые объекты (Mod_o - модификаторы объектов) или новые действия (Mod_a - модификаторы действий) соответственно из базовых объектов или действий (или символов неопределенных объектов и действий). Различаются два класса модификаторов - модификаторы смещения (например, рядом, на, слева, справа, близко от, вскоре, после), строящие новое множество, элементы которого связаны с элементами старого множества (на уровне интерпретации) смежностью в пространстве или во времени; модификаторы сужения (например, большой, белый, осторожно, быстро), для которых новое множество является подмножеством старого. В языке

\mathcal{D} все модификаторы сужения рассматриваются как функции, при переходе к программной интерпретации посредством Q трактовка модификаторов объектов остается прежней, модификаторы же действий начинают трактоваться как параметры функций. Эта разница в интерпретации модификаторов объясняется тем, что на уровне Q объекты из P_0 задаются перечислением, а действия из D_0 - параметрическими процедурами.

Переход к процедурной интерпретации формулы $p \in \mathcal{D}$ меняет семантику пропозициональных операторов и кванторов, которые начинают пониматься как операции над множествами: $\exists A$ - "найти такое A , что ...", $\forall A$ - "построить множество всех таких A , что ...", $A \vee B$ - "построить объединение множеств A и B ", $A \& B$ - "построить пересечение множеств A и B ", $\neg A$ - "построить дополнение к множеству A ".

Имена объектов понимаются как одноместные функции, например, стол (x) - функция вида $g(x) = y$: "построить множество y всех объектов из x , являющихся столами". Символы переменных трактуются как множества объектов из P . Многоместные модификаторы сводятся к одноместным, например, на (x, y) - "построить множество A таких объектов из x , каждый из которых располагается на каком-либо объекте из y ". Эта функция заменяется набором одноместных функций вида на_x (для фиксированного $x \in XU\{P\}$), определенных на любых множествах $y \in XU\{P\}$. При таком способе описания, например, выражению естественного языка вида "множество книг, лежащих на столе" будет соответствовать формула: $(x = \text{стол}(P)) \& (y = (\text{на}_x \text{ о книга})(P))$, где "о" - знак суперпозиции функций. Это выражение требует построения двух множеств: x - "множество объектов из P , являющихся столами", y - "множество объектов из P , являющихся книгами и таких, что каждый из них располагается на каком-либо объекте из x ".

3. Текстовый анализатор \mathcal{L}

Введем обозначения. Пусть W - словарь языка действий - множество словоформ всех слов, использующихся в этом языке; W_R - R -граф [7], ставящий в соответствие каждому элементу $w \in W$ набор его грамматико-семантических признаков ξ_w . К грамматическим признакам относятся типы словоизменительных парадигм и типы управления. Семантические признаки отражают семантическую сочетаемость слов, например, для имен объектов это - типы применимых к ним действий и модификаторов, для имен действий - типы действий и типы применимых к ним модификаторов.

Через Σ обозначим объединение множества всех объектов из \mathcal{O} и множества всех действий из \mathcal{A} , gap - вспомогательный символ, который может обозначать любую цепочку символов из $(W \cup \Sigma)^*$; K - множество категорий элементов из $(W \cup \Sigma)^*$. Все словоформы из W принадлежат одной и той же категории \tilde{W} . Основные категории, выделяемые из Σ , - это объекты $(\tilde{\mathcal{O}})$, модификаторы объектов $(\tilde{\text{Mod}}_0)$, действия $(\tilde{\mathcal{A}})$, модификаторы действий $(\tilde{\text{Mod}}_A)$. Каждая из этих категорий имеет подкатегории, соответствующие типам объектов, действий и модификаторов. Одноместная функция k ставит в соответствие формуле $\varphi \in \Sigma$ имя наименьшей категории, к которой относится φ , $k_\alpha(\varphi)$ - одноместный предикат, определенный на элементах Σ и принимающий значение, равное 1, если φ относится к категории с именем α .

Одноместные функции возврата к тексту $\theta_1, \dots, \theta_5$ ставят в соответствие заданному элементу $\varphi \in \Sigma$ грамматико-семантические признаки словоформы из W , обозначающей соответственно последний из применявшихся в φ объект, модификатор объекта, процесс, инструмент, субъект процесса, названного формулой φ . Если некоторые из этих словоформ при наименовании процесса не использовались, то соответствующее значение θ_i равно

пустому множеству. Двухместный предикат π_j определен на множестве $W \cup \Theta$, где $\Theta = \{\theta_i(x) | 1 \leq i \leq 5, x \in \Sigma\}$. Предикат $\pi_j(w_1, w_2)$ предназначен для проверки связанности словоформ w_1 и w_2 . Каждому выделяемому типу связанности соответствует свой предикат π_j .

Текстовый анализатор есть совокупность переписывающих правил, восстанавливающих по тексту T стратегию вывода формулы процесса, который необходимо выполнить. Правила основного анализатора имеют вид: $\Phi \xrightarrow{U_1, U_2} \Psi$, где левая часть представля-

ет собой цепочку в алфавите $W \cup \Sigma \cup \{\text{gap}\}$, правая часть Ψ - цепочку в алфавите $\Sigma \cup \{\text{gap}\}$. Каждое правило сопровождается: 1) условиями применимости U_1 , определяемыми значениями предикатов из π_j на элементах $W \cup \Sigma$ и значениями предиката k_α на элементах Σ , входящих в Φ ; 2) действиями U_2 , состоящими в определении значения функции k на Ψ . Помимо правил указанного типа, \mathcal{L} содержит метаправила, применяющиеся в том случае, когда дальнейший анализ текста показал, что построенный к данному моменту вывод формулы в чем-то неверен и его нужно скорректировать. Метаправила реализуют механизм возврата в цепочке вывода и имеют вид:

$$\Phi \xrightarrow{U_1, U_2} \Psi, \text{ где } \Phi, \Psi \in \Sigma.$$

Применение правила $\Gamma: \Phi \xrightarrow{U_1, U_2} \Psi$ основного анализатора соответствует введению шага вывода в $\mathcal{D}: T, \Phi_1, \dots, \Phi_t \vdash_{\Gamma} \Psi$, где Φ_1, \dots, Φ_t - все элементы Σ , использовавшиеся в Φ .

4. Язык описания действий

В G по полученному описанию процесса строится описание последовательности действий из D_0 , т.е. производится доопределение формулы языка \mathcal{D} . Для этого могут использоваться:

- 1) отношения порядка, заданные на множествах P_0 и D_0 (родо-видовые отношения и отношения "часть-целое");
- 2) причинно-следственные отношения на D_0 ;
- 3) условия выполнимости действий;
- 4) прочие отношения предпочтения на множествах объектов и действий (например, удобство реализации действия и т.д.).

Формула $\varphi \in \mathcal{D}$, построенная по тексту T анализатором \mathcal{A} , переводится посредством G в последовательность Ψ действий на \mathcal{M}_1 . Последовательность $\Psi = \langle \Psi_1, \Psi_2 \rangle$, где Ψ_1 почти идентична φ . Различие между ними заключается в следующем: φ содержит кванторы существования, определяющие, из каких множеств объектов и действий нужно выбирать представителей; в Ψ_1 это выбор уже сделан, и Ψ_1 не содержит кванторов существования; если в φ использовались символы из \mathcal{O} , ρ , δ , ε , то в Ψ_1 вместо них должны быть подставлены соответствующие доопределения. Формула Ψ_2 задает множество следствий из действий, описанных посредством Ψ_1 .

5. Пример языка действий

Рассмотрим мир \mathcal{M} , состоящий из блоков и шаров, которые можно перемещать в пространстве и которые могут различаться цветом и размерами. В \mathcal{M} допустимы пять типов действий. Каждому действию ставятся в соответствие условия применения и следствия, задаваемые логическими формулами. Имена процессов в данном языке совпадают с именами действий.

Условия или следствия действий, общие для группы действий \mathcal{M} , описываются отдельно в виде законов развития \mathcal{M} . Они могут быть, например, такими: 1) ни один объект не может находиться более чем в одном месте; 2) ни один объект не может быть поставлен на круглый объект; 3) один объект можно поставить на другой только в том случае, если на первом ничего не

стоит; 4) при перемещении объектов с ними не происходит никаких изменений, кроме изменения пространственного расположения (не меняется ни цвет, ни форма); 5) если $A_1, \dots, A_n \subseteq A \subseteq P$, где A_1, \dots, A_n - имена всех объектов, связанных с объектом A родо-видовым отношением, то выбор элемента из A , требуемый в формуле процесса \mathcal{E} , аналогичен выбору элемента в одном из множеств A_1, \dots, A_n ; 6) действия могут выполняться только последовательно; 7) могут выполняться только те действия, которые были описаны входным текстом или являются следствиями таковых; 8) все действия, не являющиеся неопределимыми или невыполнимыми, выполнимы. Такие законы являются метавысказываниями об M и определяются в виде логических формул, например, первым двум из них соответствуют такие формулы:

$$1) \forall x \in P, \forall y_1, y_2 \in \text{Loc}[y_1, y_2 \in \text{loc}(x)) \supset (y_1 = y_2)],$$

где Loc - базовый объект - множество координат объектов из P , $\text{loc}(x)$ - базовая процедура, задающая координаты объекта $x \in P$;

$$2) \forall x \in \text{круглый}(P) \quad \exists y[y \in \text{на}(P)].$$

Каждый тип действия в Q определяется некоторым алгоритмом, машинной программой, содержащей параметры (недоопределенности), т.е. тип действия - процесс, недоопределенное действие (или, что то же самое, множество допустимых действий). Доопределение может производиться разными путями: 1) названием (выбором) конкретного имени действия этого типа (например, если речь идет о действиях типа "перемещение", то выбор действия "поднять" привносит параметр "характер перемещения - вертикальное"); 2) применением модификаторов, например, "поднять на высоту 1 метра"; 3) спецификацией используемых объектов; 4) использованием некоторых стандартных для мира M способов выполнения действий (подстановка параметров по законам M - по умолчанию).

Возможны такие состояния действия: выполнено, выполнимо, планируемо, неизбежно, неопределимо, невыполнимо. Последнее означает, что действие невыполнимо при текущем состоянии M_1 , а не вообще в M ; вообще невыполнимых действий в M нет. Действие d невыполнимо, если либо а) не выполнены условия его применения, либо б) оно запрещено по одному из законов M . Если не имеет места ни "а", ни "б" и действие d определимо в терминах m_0 , то оно выполнимо; если d не может быть доопределено в терминах m_0 , то оно неопределимо. Действие d планируемо, если путем анализа входного текста построена логическая формула, описывающая d , и показана выполнимость d в m_1 . Действие d неизбежно, если d является следствием процесса, описанного входным текстом; d выполнено, если реализовано и само d , и все его следствия.

Типы действий в M .

Действия первого уровня: 1) поиск: найти₁, искать, поискать; 2) пространственные перемещения: а) поставить: поставить, положить, бросить, побросать, повыбрасывать, выбросить, вытряхнуть; б) взять: взять, достать, поднять, снять, убрать₁, выбрать₁; в) взять и поставить: переставить, перенести, переложить, убрать₂.

Действия второго уровня (при выполнении которых используются действия первого уровня): 3) изменение состояния: закрыть, открыть, освободить, заполнить; 4) сравнение: сравнить, выбрать₂, сопоставить; 5) измерение: измерить, найти₂.

В M выделяются два типа объектов: 1) конкретные: куб, шар; 2) абстрактные: предмет, вещь, что-то, нечто, это, то.

Типы модификаторов в M .

Модификаторы первого уровня: 1) качественные модификаторы объектов: большой, маленький, круглый, кубический, красный, синий, средний; 2) кванторные модификаторы объектов: какой-нибудь, некий, любой, произвольный, каждый, всякий; 3) модифика-

торы состояния (употребляются в сочетании с пространственными модификаторами): располагаться (на), стоять (у), находиться (под) и т.д.; 4) пространственные модификаторы объектов: на, у, под, над, за, перед, сзади, близко, далеко, рядом, недалеко; 5) временные модификаторы объектов: вчера, сегодня, завтра и т.д. (временные формы относятся к модификаторам действий только в том случае, когда они указывают последовательность выполнения или период повторяемости действий, все прочие случаи относятся к временным модификаторам объектов, например, "каждый день выполнять ..." - временной модификатор действия, "который каждый день выполняет ..." - временной модификатор объекта); 6) качественные модификаторы действий: осторожно, быстро, медленно, вертикально, горизонтально, навстречу; 7) временные модификаторы действий: сразу, после, до, скоро, потом; 8) пространственные модификаторы действий: здесь, там; 9) циклические модификаторы действий: всегда, каждый раз, всякий раз.

Модификаторы второго уровня: и, или, либо, который, тот ... который, тот ... что, там ... где.

Один и тот же элемент $w \in W$ может относиться к нескольким типам модификаторов. Какой именно из этих типов имеет место, выясняется в процессе анализа текста. В результате этого элементу w ставится в соответствие элемент множества Mod , этому элементу приписывается имя категории, соответствующей типу, установленному для w . Аналогично каждому $w \in W$, относящемуся к одному из перечисленных типов объектов или действий, ставятся в соответствие элементы \mathcal{P} или \mathcal{D} , каждому из которых приписано имя категории, соответствующей типу w .

Примеры анализа текстов.

Следующим фразам естественного языка $(\varphi_1 - \varphi_3)$ поставлены в соответствие формулы $\Psi_1 - \Psi_3$ языка \mathcal{D} :

φ_1 : взять блок со стола;

Ψ_1 : $\exists x \exists y \exists z [(x = \text{стол}(P)) \ \& \ (y = (\text{блок о на}(P)) \ \& \ (z = p_1)]$ взять $(y, z) \Rightarrow s$, где z - инструмент действия; взять $p_1 \in p$. При переходе от \mathcal{Q} к Y будет: 1) выбрано по одному представителю из каждого множества X , Y и Z ; 2) изменено значение функции лос на элементе, выбранном из Y ; 3) символ неопределенной ситуации s заменен полностью определенным описанием ситуации, например, выражением $Y = \underline{v}(Z)$, где \underline{v} соответствует предлогу "в";

Φ_2 : взять большой красный блок;

Ψ_2 : $\exists x \exists y [(x = (\text{большой о красный о блок}(P)) \ \& \ (y = p_1)]$ взять $(x, y) \Rightarrow s$.

Модификатор большой на уровне Q может отображаться, например, таким образом: большой $(A) = \{x | (x \in A) \ \& \ (v(x) > v_{cp}(A))\}$, где $A \subseteq P$, $v(x)$, $v_{cp}(A)$ - функции, определяющие соответственно объем объекта x и средний объем элементов из A ;

Φ_3 : взять блок, который стоит на красном блоке;

Ψ_3 : $\exists x \exists y \exists z [(x = (\text{красный о блок}(P)) \ \& \ (y = (\text{на } x \text{ о блок}(P)) \ \& \ (z = p_1)]$ взять $(y, z) \Rightarrow s$.

При преобразовании выражения Φ_3 в формулу Ψ_3 анализатор \mathcal{A} использует следующие правила:

$$\tau_1: w \xrightarrow{U_1^1, U_2^1} \bar{w}(P),$$

$$U_1^1: k_{\tilde{w}}(w) = 1, \quad \pi_c(w, s) = 1; \quad U_2^1: k(\bar{w}(P)) = \tilde{\mathcal{P}}_1,$$

где π_c проверяет по графу W_R , совпадают ли грамматические признаки словоформы w со стандартными признаками существительного $(\pi_c(w, s))$, прилагательного $(\pi_c(w, A))$, глагола $(\pi_c(w, V_e))$, e - тип модели управления глагола; $\tilde{\mathcal{P}}_1$ - категория конкретных объектов, \bar{w} - элемент $P_0 \cup D_0 \cup$

Mod , поставленный в соответствие элементу w графом w_R ;

$$r_2: w \ a(P) \xrightarrow{U_1^2, U_2^2} (\bar{w} \circ a)(P),$$

$$U_1^2: k_{\tilde{w}}(w) = 1, \quad \pi_c(w, A) = 1, \quad k_{\tilde{\mathcal{P}}_1}(a(P)) = 1,$$

$$\pi_r(w, \vartheta_1(a(P))) = 1,$$

$$U_2^2: k((\bar{w} \circ a)(P)) = \tilde{\mathcal{P}}_1,$$

где π_r проверяет грамматическую согласованность указанных в качестве аргументов словоформ;

$$r_3: w \ u \ p \xrightarrow{U_1^3, U_2^3} p \ \& \ \bar{u}_p(P),$$

$$U_1^3: k_{\tilde{w}}(w) = 1, \quad \pi_c(w, V_1) = 1, \quad k_{\widetilde{\text{Mod}}_0^c}(\bar{w}) = 1,$$

$$\pi_r(u, \vartheta_1(p)) = 1,$$

$$U_2^3: k(p \ \& \ \bar{u}_p(P)) = \widetilde{\text{Mod}}_0^n,$$

где $\widetilde{\text{Mod}}_0^c$ - категория модификаторов состояния, $\widetilde{\text{Mod}}_0^n$ - категория пространственных модификаторов объектов;

$$r_4: a(P), \text{ который } p \ \& \ b_p(P) \xrightarrow{U_1^4, U_2^4} p \ \& \ (b_p \circ a)(P),$$

$$U_1^4: k_{\tilde{\mathcal{P}}_1}(a(P)) = 1, \quad k_{\tilde{\mathcal{P}}_1}(p) = 1, \quad k_{\widetilde{\text{Mod}}_0^n}(p \ \& \ b_p(P)) = 1,$$

$$\pi_r(\vartheta_1(a(P)), \text{ который }) = 1, \quad \pi_r(\vartheta_1(a(P)), \vartheta_2(b_p(P))) = 1,$$

$$U_2^4: k(p \& (b_p \circ a)(P)) = \tilde{\mathcal{P}}_1,$$

$$r_5: wp_1 \& \dots \& p_n \xrightarrow{U_1^5 U_2^5} \exists x_1 \dots \exists x_n \exists x_{n+1} \dots$$

$$\dots \exists x_{n+m} [(x_1 = p_1) \& \dots \& (x_n = p_n) \& (x_{n+1} = p_1) \& \dots \\ \dots \& (x_{n+m} = p_m)] \bar{w}(x_{i_1}, \dots, x_{i_t}) \Rightarrow s,$$

где $t \leq n+m, m \geq 0, p_1, \dots, p_m \in \rho, x_{n+1}, \dots, x_{n+m}$ - неопределенные во входном тексте параметры функции \bar{w} ;

$$U_1^5: k_{\tilde{w}}(w) = 1, k_{\tilde{\mathcal{P}}_1}(p_1 \dots p_n) = 1, \pi_c(w, v_2) = 1,$$

если $\vartheta_1(p_1 \& \dots \& p_n)$ определено, то $\pi(w, \vartheta_1(p_1 \& \dots \& p_n)) = 1,$

$U_2^5: k(F) = \tilde{\mathcal{D}}_v$, где F - правая часть правила r_5 , $\tilde{\mathcal{D}}_v$ - категория действий типа "взять".

Указанной последовательности правил $r_1 - r_5$ соответствует следующий вывод формулы Ψ_3 в \mathcal{D} :

$$T \vdash_{r_1} p_1 = \underline{\text{блок}}(P),$$

$$T \vdash_{r_1} p_2 = \underline{\text{блок}}(P),$$

$$T, p_2 \vdash_{r_2} p_3 = (\underline{\text{красный}} \circ \underline{\text{блок}})(P),$$

$$T, p_3 \vdash_{r_3} p_4 = p_3 \& \underline{\text{на}}_{p_3}(P),$$

$$T, p_1, p_4 \vdash_{r_4} p_5 = p_3 \& ((\underline{\text{на}}_{p_3} \circ \underline{\text{блок}})(P)),$$

$$T, p_3, p_5 \vdash_{r_5} \Psi_3.$$

Л и т е р а т у р а

1. ТИМОФЕЕВА М.К. Взаимодействие с ЭВМ на естественном языке //Теория алгоритмов и ее приложения. - Новосибирск, 1989. - Вып. 129: Вычислительные системы. - С. 96-127.
2. BARWISE J. Situations and Attitudes. - Cambridge: MIT press. - 1983. - 352 p.
3. ПЕЛЛАВИН Р., АЛЛЕН Дж.Ф. Формальная логика планов в задачах, богатых временными связями //ТИИЭР. - 1986. - Т.74, №10. - С. 79-100.
4. BARTSCH R. Tenses, Aspects, and Other Scopes in Discourse //ITLI Republication Series for Logic, Semantics and Philosophy of Language LP-88-07. - Amsterdam. - 1988. - 32 p.
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.
6. ГОНЧАРОВ С.С., СВИРИДЕНКО Д.И. Σ-программирование //Логико-математические основы проблемы МОЗ. - Новосибирск. -1985. - Вып. 101: Вычислительные системы. -С. 3-29.
7. ВЕЛЬБИЦКИЙ И.В., ХОДАКОВСКИЙ В.Н., ШОЛМОВ Л.И. Техно - логический комплекс производства программ на машинах ЕС ЭВМ и БЭСМ-6. -М.: Статистика. - 1980. - 263 с.

Поступила в ред.-изд.отд.
13 сентября 1989 года