Доклады Академии наук СССР
1975. Том 223, № 6

УДК 681.3.06:51

КИБЕРНЕТИКА И ТЕОРИЯ РЕГУЛИРОВАНИЯ

И. В. ВЕЛЬБИЦКИЙ

НОРМАЛЬНОЕ ЗАДАНИЕ СЕМАНТИК ЯЗЫКОВ СОВРЕМЕННЫХ СИСТЕМ ПРОГРАММИРОВАНИЯ

(Представлено академиком В. М. Глушковым 24 III 1975)

Из двух классов объектов будем условно один называть исходным, определяемым, а другой — известным, эталонным. Формальное описание отображения исходного класса объектов в эталонный определяет некоторым образом семантику исходного класса объектов ([1]). В общем случае под семантикой будем понимать множество цепочек последовательных отображений одного класса объектов, который называется исходным, в другой или другие известные, которые называются эталонными. Из многочисленных цепочек последовательных отображений будем рассматривать те, которые объектам исходного языка, как множеству его предложений, понятных и установившихся в некоторой сфере человеческой деятельности, ставят в соответствие объекты другого известного языка, являющегося языком команд автоматическому устройству или вычислительной машине. Язык для задания указанных отображений будем называть семантическим метаязыком.

Семантический метаязык, являясь формальной системой, должен удовлетворять следующим требованиям: во-первых, он должен обеспечивать описание исходного языка наиболее простым и естественным образом с использованием небольшого числа правил (аксиом), легко понимаемых и быстро усваиваемых; во-вторых, он не должен иметь обязательных смысловых категорий таких, употребление которых навязывалось бы его собственными правилами (наподобие кванторов в языке логики предикатов) или правилами описываемого им исходного или эталонного языка; в-третьих, семантический метаязык должен обеспечивать достаточно точное и полное (но не избыточное) описание исходного языка, которое можно использовать в автоматической системе, обслуживающей этот метаязык.

В работе приведено определение семантического R-метаязыка, удовлетворяющего перечисленным требованиям и эффективного для задания языков современных систем программирования. Семантический R-метаязык получен как развитие метаязыка магазинных автоматов ([2]) и синтаксического R-метаязыка ([3, 4]) в направлении дальнейшего обобщения понятия памяти. В отличие от известных способов задания семантики ([5, 6]), основанных на использовании некоторой эталонной машины или языка, доопределяющего синтаксические правила соответствующих грамматик, предлагаемый подход основан на использовании различных абстрактных памятей. Операции над этими памятями включены в R-метаязык и используются для описания одновременно синтаксиса и семантики исходного языка. Для формального описания синтаксиса и семантики большинства языков достаточно задать 4 вида абстрактных памятей: счетчиковую, регистровую, табличную и вагонную, из которых первая известна, а остальные вводятся впервые в данной работе.

При задании каждого вида абстрактной памяти R-метаязыка определяются допустимые над ними операции: $W_m^{\pi}(\alpha_1, \ldots, \alpha_n)$, где π — имя абстрактной памяти, n — число элементов в ячейке памяти π, m — способ обращения (запись, чтение, поиск и т. п.) ко всем элементам доступной

ячейки, $m=0, 1, \ldots$; α_j — цепочка символов в алфавите Г памятей R-метаязыка либо некоторая функция, область значений которой есть множество Г*, $1 \leqslant j \leqslant n$. Шесть значений параметра m, $m=0, 1, 2, 21, 3, 31$, фиксированы для всех видов абстрактных памятей и означают соответственно: ничего не делать, запись, чтение с разрушением и без разрушения, поиск по сравнению и несравнению. Запись аргументов $\alpha_1, \ldots, \alpha_n$ осуществляется безусловно в соответствующие элементы доступной ячейки памяти π, чтение — при условии, что все аргументы, которые не обозначены символами ε (пустая цепочка), либо ϕ (пустое множество), совпали с записями в соответствующих элементах доступной ячейки. При несовпадении операция считается невыполнимой. При $\alpha_j=\varepsilon$, над j-м элементом доступной ячейки любого вида памяти никаких операций не производится: $W_m{}^{\pi}(\varepsilon, \ldots, \varepsilon)=W_0$. При $\alpha_j=\phi$ (для $n=1$ $W_m{}^{\pi}(\phi)=W_m{}^{\pi}$) над j-м элементом доступной ячейки указанные операции выполняются особым образом, оговоренным при определении каждого вида памяти. Например, для рассмотренных приложений осуществляется чтение любой записи из соответствующего элемента, а запись и поиск — текущего символа входного текста.

Под регистровой памятью ($\pi=P$, $n=1$) понимается неограниченная вправо лента с безадресным принципом обращения. Запись в P-память цепочки символов α осуществляется приписыванием справа к содержимому ленты цепочки α. При чтении с разрушением и без производится выборка всех записанных в P-память символов, совокупность которых рассматривается как некоторое единое слово, не ограничиваемое по длине.

Под счетчиковой памятью ($\pi=C$, $n=1$) понимается неограниченная влево лента, в которой по операции $W_1{}^{C}$ содержимое увеличивается на $+1$, а по $W_{21}{}^{C}$ считывается без разрушения как некоторое единое слово.

Под табличной памятью ($\pi=T$) понимается бесконечная лента, в которой каждая ячейка может содержать n записей, не ограничиваемых по длине. Число записей, их длина и содержание, вообще говоря, различны в различных ячейках T-памяти. Запись и чтение в (из) T-памяти производится по определенным выше правилам в (из) ячейки, которая называется доступной. Определение доступной ячейки производится заранее по операции поиска по сравнению или несравнению. По операции $W_3{}^{T}(\alpha_1, \ldots, \alpha_n)$ все аргументы, у которых $\alpha_j \neq \varepsilon$, сравниваются с аналогичными (по j) записями в каждой ячейке T-памяти. При сравнении соответствующая ячейка объявляется доступной для любого числа любых последующих операций, а при несравнении указанная операция считается невыполнимой. По операции $W_{31}{}^{T}(\alpha_1, \ldots, \alpha_n)$ при несравнении доступной объявляется любая свободная ячейка T-памяти, а при сравнении операция считается невыполнимой.

Под вагонной ($\pi=B$, $n=1$) понимается такая комбинация магазинной и бобслей-памятей ([1, 4]), у которой для чтения и записи доступны оба конца. Вагонную память удобно представлять в виде бесконечной ленты с безадресным принципом обращения. К любому слову на ленте по операции записи слева ($m=1$) или справа ($m=10$) может быть приписана цепочка символов α: первый символ цепочки записывается в первую свободную ячейку соответственно до или после слова на ленте, второй — во вторую и т. д. Чтение возможно только самого левого ($m=2$, 21) или правого ($m=20$, 210) символа слова на ленте. Ограничений на порядок чтения (записи) с любого конца B-памяти, так же как и на число обращений к ней, не накладывается.

Определение. Грамматикой G в семантическом R-метаязыке называется упорядоченная последовательность из 10 элементов $G=(T, S, F, K, \Pi, \Gamma, P, R, r_0, r_{\varnothing})$, где:

1) T, S — конечные алфавиты, соответственно символов языка (термов) и символов грамматики языка (синтермов ([4]));

2) F — бинарное отношение на множестве $T \times S$, задающее соответствие термов синтермам;

3) K — непустое конечное множество имен комплексов правил, $K=\{r_0, r_1, \ldots, r_h\}$;

4) Π — конечное множество имен памятей R-метаязыка;

5) Γ — алфавит памятей Π;

6) P — непустое конечное множество правил вида

$$a \xrightarrow{\prod_{i=1}^{\sigma} W_{mi}^{\pi i}(\alpha_1, \ldots, \alpha_{ni})} r, \quad a \in S^*,$$

$\prod_{i=1}^{\sigma} W_{mi}^{\pi i}(\alpha_1, \ldots, \alpha_{ni})$ — отображение, задающее σ последовательных операций над памятями из Π; r — имя комплекса правил-преемников, $r \in K \cup \{r_\varnothing\} \cup \{r_\pi\}$, r_π — выделенное имя, которое при выполнении соответствующего правила заменяется на имя комплекса правил-преемников, считываемое с выхода памяти π, $\pi \in \Pi$,

$$[r_\pi] \in K \cup \{r_\varnothing\} \cup \{r_\pi\}, \quad \Gamma \cap (K \cup \{r_\varnothing\} \cup \{r_\pi\}) \neq \phi;$$

7) R — взаимно однозначное отображение множества K в непустое множество подмножеств множества P;

8) r_0 — аксиома грамматики;

9) $r_\varnothing$ — выделенное имя, обозначающее конечное правило грамматики.

Рассмотрим, как элементы грамматики G в семантическом R-метаязыке используются для задания исходного языка L, если эталонным является $L_э$.

Под конфигурацией k грамматики G понимается кортеж следующих элементов: $k=(l\#r\#\Omega)$, где $l \in T^*$, $r \in K \cup \{r_\varnothing\} \cup \{r_\pi\}$, Ω — состояние памятей R-метаязыка, $\Omega \in \mathfrak{P}(\mathfrak{P}(\mathfrak{P}(\Gamma^*)))$; Γ^*, $\mathfrak{P}(\Gamma^*)$, $\mathfrak{P}(\mathfrak{P}(\Gamma^*))$, $\mathfrak{P}(\mathfrak{P}(\mathfrak{P}(\Gamma^*)))$ — области значений состояний соответственно элементов, ячеек, памяти π и памятей Π R-метаязыка. Последовательность $k_1 \Rightarrow k_2 \Rightarrow \ldots \Rightarrow k_q$ (сокращенно $k_1 \overset{*}{\Rightarrow} k_q$) называется распознающим выводом, если элементы конфигурации $k_{i+1}=(l\#r_{v_{i+1}}\#\Omega_{i+1})$ определяются через элементы конфигурации $k_i=(xl\#r_{v_i}\#\Omega_i)$ соответствующим правилом из r_{v_i}, где $i=1, \ldots, q-1$; $l, x, xl \in T^*$; $r_{v_i}, r_{v_{i+1}} \in K \cup \{r_\pi\}$, $r_{v_q} \in K \cup \{r_\varnothing\} \cup \{r_\pi\}$; $\Omega_i, \Omega_{i+1} \in \mathfrak{P}(\mathfrak{P}(\mathfrak{P}(\Gamma^*)))$. Правило из r_{v_i} применимо к элементам конфигурации k_i, если цепочка x соответствует цепочке a правила, т. е. $x=F(a)$ и выполнимы все указанные в нем операции над памятями R-метаязыка. Назовем языком, задаваемым грамматикой G, множество $L(G)=\{l \in T^* \mid (l\#r_0\#\Omega_\varepsilon) \overset{*}{\Rightarrow} (\varepsilon\#r_\varnothing\#\Omega_{\text{ВЫХ}} \cup \Omega_{\Pi \setminus \{\text{ВЫХ}\}}), \Omega_{\text{ВЫХ}}=l_э \in L_э\}$, где $\Omega_\varepsilon \in \mathfrak{P}(\mathfrak{P}(\mathfrak{P}(\phi)))$, $\Omega_{\text{ВЫХ}}$ — состояние вагонной памяти ВЫХ, в которой формируется предложение $l_э$, соответствующее предложению l исходного языка, $\Omega_{\text{ВЫХ}} \in \mathfrak{P}(\Gamma)$, $\Omega_{\Pi \setminus \{\text{ВЫХ}\}} \in \mathfrak{P}(\mathfrak{P}(\mathfrak{P}(\Gamma^*)))$.

Пример. Рассмотрим следующий фрагмент языка АЛГОЛ-60 ([7]):

⟨язык⟩::=$\{$⟨тип⟩⟨список типа⟩;$\}_1\{$⟨идентификатор⟩:=⟨простое арифметическое выражение⟩$\}_1$

⟨тип⟩::= ***real*** | ***integer***

⟨первичное выражение⟩::=⟨идентификатор⟩ | (⟨простое арифметическое выражение⟩).

Определения остальных метапеременных приведены в ([7]), а в итерационных скобках заключены конструкции языка, которые повторяются не менее одного раза. Примером предложения указанного языка является: ***real*** a, b, c;***integer*** $a1, b1, c1$; $a1:=b1\times(c1+a1)/b1$; $a:=b+a\times(c-b)$;

Если в качестве эталонного языка принять польскую систему обозначений ([1]), то формальное описание синтаксиса и семантики приведенного исходного языка задается следующей R-грамматикой: $G=(T, S, F, K, \Pi, \Gamma, P, R, r_0, r_\varnothing)$, где

$T=\{A|B|\ldots|Z|a|b|\ldots|z|0|1|\ldots|9|+|-|\times|/|\div|)\,|\,(\,|\,$ ***real*** $|$ ***integer*** $|:=|,|;\,|\,\}$, $S=\{$ *б* $|$ *бц* $|rt|\langle+\rangle|\langle\times\rangle|)\,|\,(|,|;\,\}$, $F=\{$*б*:: $\cong A|B|\ldots|Z|a|b|\ldots|z|$, *бц*:: $\cong A|B|\ldots|Z|a|b|\ldots|z|0|1|\ldots|9$, ⟨+⟩:: $\cong +|-$, ⟨×⟩:: $\cong \times|/|\div$,

$rt::=$***real|integer***$\}$, $K=\{r_0, r_1, \ldots, r_{15}\}$, $\Pi=\{$Р, Р1, Р2, ТИ, С, В, ВЫХ$\}$, $\Gamma=\{r_3, r_5, r_9, \ldots, r_{14}, \{\text{С}\}, (T\setminus\{,|\}\,|\,(|;))\}$,

R: $r_0 \sim \{rt \xrightarrow{W_1^B(r_3)W_1^{P1}} r_1\}$, $r_1 \sim \{б \xrightarrow{W_1^P} r_2\}$, $r_2 \sim \{бц \xrightarrow{W_1^P} r_2,\ \varepsilon \xrightarrow{W_0} r_\text{В}\}$,

$r_3 \sim \{, \xrightarrow{W_{31}^{\text{ТИ}}(W_{21}^{\text{Р}})W_1^{\text{ТИ}}(W_2^{\text{Р}}, W_{21}^{\text{Р1}}, W_{21}^{\text{С}})W_1^{\text{С}}W_1^B(r_3)} r_1,$

$; \xrightarrow{W_{31}^{\text{ТИ}}(W_{21}^{\text{Р}})W_1^{\text{ТИ}}(W_2^{\text{Р}}, W_{21}^{\text{Р1}}, W_{21}^{\text{С}})W_1^{\text{С}}} r_4\}$,

$r_4 \sim \{rt \xrightarrow{W_1^B(r_3)W_1^{P1}} r_1,\ \varepsilon \xrightarrow{W_1^B(r_{13}r_5)} r_1\}$,

$r_5 \sim \{: = \xrightarrow{W_3^{\text{ТИ}}(W_2^{\text{Р}})\ W_1^{\text{Р1}}(W_{21}^{\text{ТИ}}(\varepsilon,\ \varnothing))W_1^{\text{Р2}}(W_{21}^{\text{ТИ}}(\varepsilon,\ \varepsilon,\ \varnothing))} r_6\}$,

$r_6 \sim \{\varepsilon \xrightarrow{W_1^B(r_{10})} r_7\}$, $r_7 \sim \{\varepsilon \xrightarrow{W_1^B(r_{11})} r_8\}$, $r_8 \sim \{(\xrightarrow{W_1^B(r_{12})} r_6,\ \varepsilon \xrightarrow{W_1^B(r_9)} r_1\}$,

$r_9 \sim \{\varepsilon \xrightarrow{W_3^{\text{ТИ}}(W_2^{\text{Р}})W_1^{\text{ВЫХ}}(W_{21}^{\text{ТИ}}(\varepsilon,\ \varnothing))W_1^{\text{ВЫХ}}(W_{21}^{\text{ТИ}}(\varepsilon,\ \varepsilon,\ \varnothing))} r_\text{В}\}$,

$r_{10} \sim \{\langle + \rangle \xrightarrow{W_1^BW_1^B(r_{13})} r_6,\ \varepsilon \xrightarrow{W_0} r_\text{В}\}$, $r_{11}\{\langle \times \rangle \xrightarrow{W_1^BW_1^B(r_{13})} r_7,\ \varepsilon \xrightarrow{W_0} r_\text{В}\}$,

$r_{12} \sim \{) \xrightarrow{W_0} r_\text{В}\}$,

$r_{13} \sim \{\varepsilon \xrightarrow{W_1^{\text{ВЫХ}}(W_2^B)} r_\text{В}\}$, $r_{14} \sim \{; \xrightarrow{W_1^{\text{ВЫХ}}(:=W_2^{\text{Р1}}W_2^{\text{Р2}})} r_{15}\}$,

$r_{15} \sim \{б \xrightarrow{W_1^B(r_{14}r_5)W_1^P} r_2,\ \varepsilon \xrightarrow{W_0} r_\varnothing\}$.

Работу R-грамматики удобно проследить в процессе распознающего вывода предложения исходного языка, приведенного в начале примера. В результате в памяти ВЫХ будет получена следующая последовательность символов эталонного языка:

$$\begin{aligned}&\textit{integer}(b1)\textit{integer}(c1)\textit{integer}(a1)+\textit{integer}(b1)/\times:=\\&=\textit{integer}(a1)\textit{real}(b)\textit{real}(a)\ \textit{real}(c)\textit{real}(b)-\times+:=\textit{real}(a),\end{aligned}$$

где в скобках обозначены адреса соответствующих переменных.

В заключение отметим, что описание семантики языка неотделимо от описания его синтаксиса и не намного его увеличивает. Например, описание синтаксиса приведенного языка содержит 17 правил вместо 23 в семантическом R-метаязыке. Описание синтаксиса и семантики языка АЛГОЛ-60 содержит около 350 правил семантического R-метаязыка, что приблизительно лишь в 1,5 раза больше правил, задающих только синтаксис этого языка ([4]). При описании языка АЛГОЛ-60 в семантическом R-метаязыке было использовано около 30 памятей 6 различных видов. Важно отметить, что в качестве эталонного языка можно использовать язык микроопераций ЭВМ, что позволяет использовать семантический R-метаязык для построения вычислительных машин по новым принципам ([3]).

Институт кибернетики
Академии наук УССР
Киев

Поступило
20 I 1975

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] Энциклопедия кибернетики, Киев, 1974. [2] *В. М. Глушков,* Кибернетика, № 5 (1968). [3] *В. М. Глушков и др.,* Кибернетика, № 3 (1972). [4] *И. В. Вельбицкий,* ДАН, т. 213, № 5 (1973); Кибернетика, № 3 (1973); Сб. Системное и теоретическое программирование, т. 1, Кишинев, 1974, стр. 260. [5] *Дж. Фельдман, Д. Грайс,* Сб. Алгоритмы и алгоритмические языки, в. 5, М., 1971, стр. 105. [6] *C. A. R. Heare, P. E. Lauer,* ACTA Inform., 3 (1974). [7] *P. Naur,* Ed., Comm. ACM, v. 6, № 1 (1963); перевод: Алгоритмический язык АЛГОЛ-60, М., 1965.