Доклады Академии наук СССР
1973. Том 208, № 6

УДК 681.3.06:51

КИБЕРНЕТИКА И ТЕОРИЯ РЕГУЛИРОВАНИЯ

И. В. ВЕЛЬБИЦКИЙ

К ВОПРОСУ ПОСТРОЕНИЯ ГЕНЕРАТОРОВ ПРАВИЛЬНОЙ ИНФОРМАЦИИ

(Представлено академиком В. М. Глушковым 29 V 1972)

Увеличение потока информации для обработки на современных вычислительных машинах (ВМ) и привлечение большого числа не специалистов в сферу применения вычислительной техники выдвигает как одну из актуальных задач разработку эффективных средств и методов подготовки правильной информации ([1]). Настоящая работа посвящена описанию принципов построения устройства, впервые разработанного в Институте кибернетики АН УССР и предназначенного для независимой от ВМ подготовки синтаксически правильных программ и данных в языках, грамматики которых описаны в специальном, разработанном для этих целей R-метаязыке и хранятся в постоянной памяти устройства. Эти принципы могут быть эффективно использованы также для схемного воплощения стандартизованных частей математического обеспечения в архитектуре машин четвертого поколения, а также для построения тренажеров, ориентированных на эффективное обучение языкам программирования.

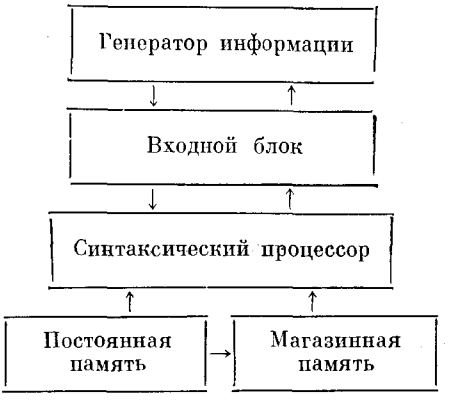

Рис. 1. Блок-схема СИНТАКСОРа — устройства генерации синтаксически правильной информации

Сущность предлагаемого подхода заключается в том, что к устройству генерации информации (печатающей машинке, телетайпу, устройству подготовки перфокарт, дисплею и т. д.) подключается некоторое устройство контроля, в постоянной памяти которого записана грамматика языка, в котором осуществляется генерация информации. Полученное в результате устройство подготовки синтаксически правильных программ и данных (сокращенно СИНТАКСОР) работает таким образом, что блокируется генерация (печать, перфорация и пр.) неправильной информации.

СИНТАКСОР состоит (см. рис. 1) из входного блока, предназначенного для формирования текущего символа подготавливаемой информации; постоянной памяти для хранения грамматик используемых языков; синтаксического процессора, осуществляющего проверку совместимости текущего символа с правилами грамматики, и магазинной памяти для организации проверки сложных синтаксических конструкций. Работа устройства осуществляется следующим образом. Текущий символ, сформированный генератором информации и входным блоком, сравнивается синтаксическим процессором с текущим комплексом правил грамматики, записанной в постоянной памяти. Вначале текущим комплексом является аксиома грамматики соответствующего языка. Если текущий символ соответствует некоторому правилу грамматики, то по ней определяется комплекс правил для проверки следующего символа, а схемы устройства вырабатывают соответствую-

щему генератору информации сигнал, разрешающий печать или перфорацию текущего символа.

Если текущий символ не соответствует текущему комплексу правил грамматики, то в устройстве вырабатывается сигнал синтаксической ошибки, по которому блокируется генерация этого символа на соответствующий носитель (бумагу, перфокарту, перфоленту и т. д.). Оператор, работающий на СИНТАКСОРе, должен с помощью клавиатуры сформировать другой текущий символ и, если он соответствует текущему комплексу правил грамматики, продолжить работу по подготовке синтаксически правильной информации. В противном случае, после неоднократной неудачной попытки напечатать или отперфорировать текущий символ информации, оператор должен либо вернуться назад и повторить подготовку определенной группы символов, либо, если его квалификации недостаточно, включить устройство на автоматическое «исправление» ошибки. В последнем случае устройство переключает цвет печати (либо идентифицирует начало ошибки любым другим способом) и снимает блокировку клавиатуры. Оператор продолжает генерировать (в частности, печатать) исходный текст. При этом проверка генерируемой информации не производится — происходит лишь перенос информации с одного носителя на другой и коррекция состояния схем СИНТАКСОРа. Через несколько символов (для реальных языков программирования через 2—3 символа) устройство скорректирует состояние своих схем (исправит ошибку), автоматически переключит цвет печати и перейдет на обычный режим работы.

Такое устройство может работать в режиме упреждающего контроля. В этом случае на клавиатуре соответствующего генератора информации выделяются, например, подсветкой те клавиши, которые, согласно грамматике языка, соответствуют правильным символам. После нажатия одной из выделенных клавиш подсвечиваются другие клавиши, соответствующие символам, которые могут быть за нажатым и т. д. Такое устройство эффективно для обучения и в значительной степени может ускорить работу оператора за клавиатурой.

К входному блоку СИНТАКСОРа может быть подключено несколько генераторов информации таким образом, что каждый из них может работать в режиме разделения времени на любом языке, грамматика которого предварительно записана в постоянной памяти. В постоянной памяти СИНТАКСОРа может быть записано несколько грамматик используемых языков. Настройка на нужную грамматику языка производится с соответствующего пульта управления в начале работы. Ориентация СИНТАКСОРа на новый язык, отличный от ранее записанных в его постоянную память, сводится к добавлению или замене одного блока постоянной памяти другим, в котором записана грамматика нового языка.

Грамматика для устройства задается в R-метаязыке. Этот метаязык является некоторым системным способом задания языков, объединяющим положительные черты как аппарата формальных грамматик, так и способов задания языков с помощью автоматных преобразователей. Отличительной особенностью R-метаязыка является сочетание возможности компактности записи реальных языков с большой скоростью распознавания их предложений, что позволяет стандартизировать соответствующий алгоритм контроля и эффективно реализовать его схемно в описанном выше устройстве.

Грамматика в R-метаязыке задается с помощью некоторой системы правил. Эти правила объединяются в комплексы, и выбор правила из комплекса связан с выполнением дополнительной операции над памятью, состоящей из n магазинов ($n \geqslant 0$). Каждое правило имеет следующий вид:

$$a \xrightarrow{W_m(\alpha_1, \ldots, \alpha_n)} r, \qquad (1)$$

где a — символ терминального алфавита T, который может содержать пустой символ ϕ; $W_m(\alpha_1, \ldots, \alpha_n)$ определяет операцию с магазинами; $m =$

$= 0, 1, 2$; α_i — *слово магазинного алфавита* $\Gamma \subset (R \cup \mathfrak{A})$, $1 \leqslant i \leqslant n$; R — множество комплексов правил, $\mathfrak{A}$ — вспомогательный алфавит: $\phi \in \mathfrak{A}$, $\mathfrak{A} \cap R = \varnothing$, $\Gamma \setminus \overline{\mathfrak{A}} = \mathfrak{A}$; r — комплекс правил-преемников, к которым осуществляется переход, если $W_m(\alpha_1, \ldots, \alpha_n)$ выполнимо, $r \in R$. При $m = 0$ никаких операций над магазинами не производится. При $m = 1$ $(m = 2)$ в магазин с номером i для всех $i = 1, \ldots, n$ записывается (стирается) слово $\alpha_i \neq \phi$. Запись слова в магазин означает последовательную слева направо по одному запись символов слова α_i в соответствующий i-й магазин. Запись слова α_i в соответствующий магазин осуществляется безусловно, стирание — при условии, что $\alpha_i \neq \phi$ в правиле грамматики и в верхних ячейках i-го магазина совпадают. При несовпадении соответствующее правило грамматики не применимо и стирание из магазинов не производится. Если $\alpha_i = \phi$, то содержимое i-го магазина не изменяется. По определению, в начале и конце процесса распознавания магазинные памяти должны быть пусты.

По определению, комплекс правил r_i может содержать имя другого комплекса r_j правил. Это означает, что все правила комплекса r_j входят в комплекс r_i. На практике достаточно, чтобы в комплекс r_i входило не больше одного имени комплекса других правил. Это позволяет за счет упорядочения правил в r_i-комплексе обеспечить компактность записи при сохранении беступиковыми схемы распознавания. При распознавании вначале просматриваются правила комплекса r_i, а затем r_j.

Множество комплексов R содержит три выделенных комплекса: r_0, $r_\varnothing$ и r_M. Первое обозначает аксиому грамматики; второе — имя пустого комплекса правил. Если в правиле (1) $r = r_\varnothing$, то a является последним символом предложения языка. Третье — r_M — обобщенное имя комплекса, равное имени, записанному на вершине магазинной памяти.

Таким образом, R-грамматика G задается тройкой

$$G = (\hat{C}, R, \hat{r}),$$

где $\hat{C}$ — алфавиты R-грамматики: T — терминальный алфавит, $\mathfrak{A}$ — вспомогательный алфавит, Γ — магазинный алфавит; R — множество комплексов правил вида (1); $\hat{r}$ — выделенные комплексы правил R-грамматики: r_0 — аксиома грамматики, $r_\varnothing$ — имя заключительного комплекса правил, r_M — обобщенное имя комплекса правил, записанного на вершине магазинной памяти.

Метаязык R-грамматик эффективен для распознавания. При распознавании исходная последовательность символов принадлежит языку, если ее первый символ встречается в левой части одного из правил r_0, каждый следующий соответствует символу, встречающемуся в левой части одного из правил-преемников примененного правила, а последний символ определен правилом с $r_\varnothing$ в правой части. Применение каждого правила сопровождается действиями с магазинной памятью.

Теорема. *Класс языков, распознаваемых R-грамматиками, совпадает с классом рекурсивно-перечислимых множеств.*

Пример. Простое арифметическое выражение языка АЛГОЛ-60, всегда оканчивающееся на «;» и в определении ([2]) которого

⟨первичное выражение⟩ : : = ⟨целое⟩ | ⟨простая переменная⟩ |

(⟨простое арифметическое выражение⟩),

задается следующей R-грамматикой из 11 правил:

$$G = (T, \Gamma, R, r_0, r_\varnothing, r_M),$$

$$T = \{б, ц, ;, +, \times,), (\}, \quad \Gamma = \{r_5, r_6, r_7\},$$

$$R = \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_\varnothing, r_M\},$$

$$r_0 = \{\phi \xrightarrow{W_1(r_7 r_8)} r_1\},$$

$$r_1 = \{+ \xrightarrow{W_0} r_2,\ r_2\},$$

$$r_2 = \{б \xrightarrow{W_0} r_3,\ ц \xrightarrow{W_0} r_4,\ (\xrightarrow{W_1(r_6 r_8)} r_1\},$$

$$r_3 = \{б \xrightarrow{W_0} r_3,\ ц \xrightarrow{W_0} r_3,\ r_M\},$$

$$r_4 = \{ц \xrightarrow{W_0} r_4,\ r_M\},$$

$$r_5 = \{\times \xrightarrow{W_1(r_8)} r_2,\ r_M\},$$

$$r_6 = \{\,) \xrightarrow{W_0} r_M\},$$

$$r_7 = \{\,; \xrightarrow{W_0} r_\varnothing\};$$

здесь $б$ — обозначение любой буквы; $ц$ — обозначение любой цифры; $+$ — обозначение любого из следующих символов языка: $+ \mid - \mid ;$ $\times$ — обозначение любого из следующих символов языка: $+ \mid - \mid \times \mid / \mid \uparrow \mid \div$.

R-грамматики реальных языков программирования отличаются компактностью представления и большой скоростью распознавания. Например. R-грамматика языка АЛГОЛ-60 задается приблизительно 200 правилами, которые занимают около 5000 бит постоянной памяти.

Построенное в Институте кибернетики АН УССР и описанное в настоящей работе устройство является основой для разработки в будущем генераторов (например, печатающих машинок) только правильной информации. Генерация неправильной информации будет блокироваться. Причем процессу контроля будут подвергаться не только синтаксические, но и формально задаваемые семантические конструкции языка. Очевидно, метаязык R-грамматик позволяет производить такой контроль при наличии компактных и дешевых памятей машин четвертого поколения.

Институт кибернетики
Академии наук УССР
Киев

Поступило
17 IV 1972

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] В. М. Глушков, И. В. Вельбицкий, А. А. Стогний, Кибернетика, № 3 (1972). [2] Алгоритмический язык АЛГОЛ-60 (пересмотренное сообщение), М., 1965.