

УДК 681.142.2

Р-ЛЯПАС — БАЗОВЫЙ ЯЗЫК МОДЕЛИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ

В.А. Воробьев

1. Предварительные замечания

1.1. Одним из недостатков большинства существующих в настоящее время языков моделирования цифровых устройств является их ориентация на определенный уровень модели: структурный, функциональный или логический (см., например, [1]). Это обстоятельство приводит к разноязычию на последовательности этапов проектирования устройства и резко снижает эффективность моделирования.

Одной из попыток преодоления указанной трудности является создание системы АВТОКОД М [2], входной язык которой обладает достаточно широким набором изобразительных средств. Однако и в этом случае фиксация языка иногда становится трудно преодолимым барьером. Дело в том, что знания входного языка системы АВТОКОД М недостаточно для расширения возможностей проектировщика-программиста, и, следовательно, всякое нововведение требует глубокого пересмотра всей системы трансляции.

Разрешение этого противоречия следует искать в построении *растущей* системы моделирования, в которой новые операции вводятся и описываются на базе входного языка и его сред-

ствами. Сама система при этом строится методом с а м о п р о г р а м м и р о в а н и я , где каждый достигнутый уровень является подпрограммой для вновь создаваемого. Достаточно подробная разработка и последовательная реализация этой идеи содержится в работах по развитию программирующей системы ПС ЛЯПАС [3,4].

1.2. Предметом настоящей работы является построение базового языка растущей системы моделирования цифровых устройств. Входной язык системы строится путем расширения первого уровня языка ЛЯПАС и соответственно называется "Расширенным ЛЯПАСом", или Р-ЛЯПАСом. Этот путь выбран по следующим причинам.

Во-первых, алгоритмы, моделирующие функционирование цифровой вычислительной машины (ЦВМ), относятся к классу логических, на представление которых и ориентирован ЛЯПАС.

Во-вторых, самопрограммирование органически присуще ПС ЛЯПАС и может быть полностью использовано без специальных разработок.

В-третьих, второй уровень ЛЯПАСа почти автоматически переносится на Р-ЛЯПАС и обеспечивает гибкость моделирующей системы. По мере необходимости любые новые средства вводятся на втором уровне, и, таким образом, их набор фактически не ограничен.

В-четвертых, система автоматического проектирования ЦВМ [5,6], с развитием которой непосредственно связана эта работа, существенно использует ПС ЛЯПАС. Описанный ниже язык является частью этой системы, а именно языком алгоритмического моделирования [5]. Вдохновляющим примером, кроме того, является система АЛЕРТ [7] на базе языка, родственного ЛЯПАСу.

Существенным недостатком ЛЯПАСа, с точки зрения потребностей моделирования, является фиксированная размерность операндов. На преодолении этого ограничения и будет сосредоточено в дальнейшем наше внимание. Все остальные сведения о языке ЛЯПАС предполагаются известными читателю.

1.3. Для описания Р-ЛЯПАСа мы воспользуемся обозначениями, предложенными А.Д. Закревским [3], добавив к ним следующие.

Произвольную конечную цепочку из символов $\tau \in \{\tau_1, \tau_2, \dots, \tau_m\} = M$ будем обозначать $\tilde{\tau}$, т.е.

$$\tilde{\tau} \in \{M, M \times M, \dots, M \times M \times \dots \times M\},$$

где символ $\times$ означает декартово произведение. Если цепочка $\tilde{\tau}$ не превышает длины n , то она обозначается выражением $\tilde{\tau}^n$.

Договоримся также, что запись

$$\{\tau \setminus \tau_1, \tau_2, \dots, \tau_n\}$$

означает удаление из множества значений τ его подмножества

$$\tau_1, \tau_2, \dots, \tau_n, \quad n \leq m.$$

1.4. Набор основных символов Р-ЛЯПАСа получается заменой $'$, $"$, $"$, $"$, $'$, $'$, $'$, $'$ на $[,] , \neg , \vdash , ;$ соответственно в наборе символов языка ЛЯПАС. Кодирование при этом сохраняется. Договоримся также, что освободившийся символ $'$ используется для обозначения дополнительных переменных и индексов, как то: α - основная переменная, α' - соответствующая дополнительная. Все прочие условности, принятые в ЛЯПАСе, сохраняются в расширении.

Это обстоятельство позволит нам в дальнейшем построить систему программирования, единую для ЛЯПАСа и его расширения, а иногда воспользоваться готовыми блоками ПС ЛЯПАС, например, трансляторами первого уровня в машинных языках.

2. Первый уровень Р-ЛЯПАСа

2.1. Операнды

Аналогично [3] вводятся следующие операнды Р-ЛЯПАСа.

Переменная $\pi_1 \in \{\alpha, \beta, \dots, \gamma, \alpha', \beta', \dots, \gamma'\}$.

Индекс $\pi_2 \in \{\alpha, \beta, \dots, \gamma, \alpha', \beta', \dots, \gamma'\}$.

Натуральная восьмеричная константа $\pi_3 \in \{0, 1, 2, \dots, 177\}$;

ричная константа $\pi_4 \in \{1, 2, \dots, 177\}$.

Комплекс $\pi_5 \in \{A, B, C, \dots, Y\}$.

Специальный комплекс $\pi_6 \in \{A, B, C, \dots, Q\}$;

$\pi_7 \in \{A, B, C, \dots, P\}$.

Индексирующий операнд $\pi_6 \in \{\pi_2, \pi_3\}$.

Операнд произвольной размерности $\pi_7 \in \{\pi_1, \pi_2, \pi_4, \pi_6\}$.

Операнд фиксированной размерности $\pi_8 \in \{\pi_3, \pi_5, \pi_6\}$.

В отличие от ЛЯПАСа, в расширенном варианте допускается использование любых фиксированных частей операндов произвольной размерности. Это достигается с помощью конструкции $\pi_7 [\pi_6, \pi_3]$, где π_6 и π_3 соответственно — первый разряд и размерность используемой части π_7 . Допустимо также использование любого разряда π_7 с помощью конструкции $\pi_7 [\pi_6]$. В общем виде операнд произвольной размерности определяется так:

$$\pi_9 \in \{\pi_7, \pi_7 [\pi_6], \pi_7 [\pi_6, \pi_3]\}.$$

Двоичные константы произвольной размерности вводятся в расширение следующей конструкцией:

$$\pi_{10} \in \{\tilde{\pi}_3 [\pi_3']\},$$

где $\tilde{\pi}_3 [$ — содержимое константы, а π_3' — ее размерность. Запись в скобках совпадает с восьмеричным кодом константы, представление же ее символами π_3' — произвольно. Например, константа $] 756025 [30$ может быть представлена как $] 75,60,25 [30$ или $] 7,56,0,25 [30$ и т.п. (Неверно представление $] 7,56,02,5 [30$, так как символ 02 не есть π_3).

Значение разряды константы располагаются в правом конце, а остальные заполнены нулями. Символ ",", конкретизирующий разделение кода на части, не кодируется.

В Р-ЛЯПАСе, таким образом, определены:

произвольный операнд — $\pi \in \{\pi_6, \pi_9, \pi_{10}\}$;

операнд, допускающий

присвоение — $\pi' \in \{\pi_9, \pi_5', \pi_6\}$.

Кроме того, в дальнейшем потребуются

$$\pi_{11} \in \{\pi_4, \pi_5\};$$

$$\pi_{12} \in \{A\pi_3, B\pi_3, \pi_1, \pi_2, \pi_4, \pi_5\};$$

$$\pi_{13} \in \{\pi_1, \pi_2\};$$

$$\pi_{14} \in \{\pi_1, \pi_2, \pi_4\}.$$

В заключение описания операндов договоримся о следующем.

2.1.1. Значение разряды всех операндов расширения находятся в правом конце операнда языка ЛЯПАС.

2.1.2. Размерность операнда π_8 совпадает с "ляпасной".

2.1.3. Собственная переменная $\tau [3,4]$ имеет плавающую размерность, определяемую семантикой операторов.

2.1.4. Нумерация разрядов операнда происходит слева направо и начинается с нуля.

2.1.5. Специальный комплекс Q отводится для хранения констант. Элемент Q_i для $i = 0, 1, \dots, \kappa$, где κ — размерность "ляпасного" операнда, содержит i единиц в левом конце; для $i = \kappa + 1, \kappa + 2, \dots, \kappa + j, \dots, \kappa + \kappa - 1$; j единиц в правом. Смысл остальных специальных комплексов не меняется.

2.1.6. Натуральные константы имеют размерность от 1 до 7, равную ближайшему сверху целому от $\lg [\pi_3]$.

2.2. Операторы

В расширении сохраняют прежний смысл все операторы ЛЯПАСа. Кроме того, вводятся два оператора присоединения $\vdash$ и $\models$ и оператор согласования размерностей $\otimes \Rightarrow \otimes$.

Запись $\xi_i \vdash \xi_j$ означает, что из операндов ξ_i и ξ_j формируется новый операнд ξ_{κ} с размерностью, равной сумме размерностей ξ_i и ξ_j . Значение ξ_{κ} получается размещением в его левой части значения ξ_i , а в правой — ξ_j .

Запись $\xi_i \models \xi_j$ отличается от присоединения справа лишь тем, что значение ξ_j записывается в левой части ξ_{κ} , а ξ_i — в правой.

$$\begin{aligned} \text{Пример: } 011010 &= 101 = 011010101 \\ 011010 &\models 101 = 101010101 \end{aligned}$$

В отличие от принятого в большинстве языков способа задания размерностей операндов в описаниях, в Р-ЛЯПАСе вводится более естественная для иерархического программирования форма, а именно операция согласования размерностей

$$\otimes \tilde{\pi} \Rightarrow \tilde{\pi}_{14} \otimes.$$

Смысл ее состоит в том, что при дальнейшем чтении программы слева направо операнды из $\tilde{\pi}_{14}$ имеют размерность, равную сумме размерностей операндов из $\tilde{\pi}$. Если при этом $\pi = \pi_3$, то

в сумму входит не размерность, а само значение π_s . Собственная переменная τ при выполнении операции согласования сохраняется.

Запись $\odot \pi \odot$ эквивалентна натуральной константе со значением, равным сумме размерностей операндов из π .

Все операторы расширения работают с операндами плавающей размерности.

2.2.1. При выполнении одноместных операторов $\circ, \bar{\circ}, \Delta, \bar{\Delta}, \vdash, \vdash, \uparrow$ результат получается с размерностью исходного операнда и (для $\Delta, \bar{\Delta}$) по соответствующему модулю.

2.2.2. Операторы $\pm, \Sigma, *, \bar{*}, \circ, \bar{\circ}, /, \bar{/}, \uparrow, \uparrow$ выравнивают τ с размерностью ячейки ЦВМ.

2.2.3. Операторы $\nabla, \vdash, \bar{\nabla}, \bar{\vdash}$ дают результат с минимальной размерностью, необходимой для хранения числа k , где k - размерность исходного операнда.

2.2.4. Двуместные операторы $<, >, \leftarrow, \rightarrow$ сохраняют размерность левого операнда.

2.2.5. Двуместные операторы $+, -, \times, \bar{\times}, :, \wedge, \vee, \oplus$ выполняются после выравнивания правых концов обоих операндов; при этом операнд минимальной размерности дополняется нулями слева. Размерность результата при этом равна:

$$\begin{aligned} k_{\min} & \quad - \text{ для } :, \wedge, \vee, \oplus; \\ (k_{\max} + 1) \bmod k & \quad - \text{ для } +, -; \\ (k_1 + k_2 - 1) \bmod k & \quad - \text{ для } \times, \bar{\times}; \end{aligned}$$

где k - размерность лямбда-операнда;

k_1, k_2 - размерности операндов;

$$k_{\min} = \min \{k_1, k_2\};$$

$$k_{\max} = \max \{k_1, k_2\}.$$

2.2.6. При выполнении операторов $\Rightarrow$ и $\Leftarrow$ пересылаемый операнд заносится в правый конец воспринимающего, а остальные разряды последнего, если они есть, заполняются нулями. Собственная переменная принимает размерность правого операнда.

2.2.7. Операторы переходов $\rightarrow, \circ \rightarrow, \vdash \rightarrow, \vdash \rightarrow, \uparrow$,

! передают управление в один из полюсов алгоритма, образуемый символами $\hat{\circ}, \hat{\circ}, \uparrow, \vdash$. При проходе через полюс собственная переменная τ принимает размерность лямбда-операнда, а информация о том, где находятся ее существенные разряды, теряется. Таким образом, слежение за правильностью так называемых "плохих" предложений типа $\hat{\circ} \pi'_s \Rightarrow$ возлагается на программиста.

2.3. О п е р а ц и и

Оператор Р-ЛПАСа имеет смысл лишь в определенной комбинации символов, которая ниже называется операцией. Некоторые операторы могут быть интерпретированы лишь в связи с операцией, в которой они используются.

Положим, что:

$$\rho_1 \in \{\nabla, \vdash, \vdash, \vdash, \uparrow\};$$

$$\rho_2 \in \{\nabla, \vdash, \pm, <, >, \leftarrow, \rightarrow, \times, \bar{\times}, +, -, \wedge, \vee, \oplus\};$$

$$\rho_3 \in \{\rightarrow, \circ \rightarrow, \vdash \rightarrow\};$$

$$\rho_4 \in \{\circ, \bar{\circ}, \Delta, \bar{\Delta}\};$$

$$\rho \in \{\rho_1, \rho_2, \rho_3, \rho_4, \Rightarrow, \Leftarrow, \vdash, \bar{\vdash}, \uparrow, \bar{\uparrow}, \Sigma, \circ, *, \bar{*}, /, \bar{/}\}.$$

Операцией называется составной символ α , где

$$\alpha \in \{\vdash, \pi, \rho_1, \rho_2 \pi, \Sigma \pi_{13} \pi_{13}, \quad (I)$$

$$\rho_4 \pi', \quad (2)$$

$$\Rightarrow \pi', \Rightarrow (\pi'), \pi_4 \Rightarrow (\pi_{13}), (\pi_{13}) \Rightarrow \pi_4, \pi' \Leftarrow \pi', \quad (3)$$

$$\pi \bar{\times} \pi'_3 \pi'_2, \pi \bar{\times} \pi'_3 \pi'_2, \quad (4)$$

$$\rho_3 \pi'_3, \rho_3 [\pi'_3], \uparrow, \quad (5)$$

$$\vdash \pi'_3, \vdash [\pi'_3], \uparrow \pi_{13}, \quad (6)$$

$$\hat{\circ} \pi'_3, \hat{\circ} \pi_{13}, \quad (7)$$

$$\uparrow \pi_{11}, \uparrow \vdash, \uparrow \pi_{12}, \uparrow \pi_{12}, \quad (8)$$

$$*, \bar{*}, \pi_{11} *, \pi_{11} \bar{*}, \pi_{11} \circ, \pi_{11} \bar{\circ}, \quad (9)$$

$$\uparrow \pi_{13} \pi_{13} \pi_{13}, \uparrow \pi_{13} \pi_{13} \pi_{13}, \quad (10)$$

$$\odot \pi \Rightarrow \pi_{14} \odot, \odot \pi \odot \}. \quad (II)$$

Множество символов α упорядочено так, что в строках

(1), (2) - находятся операции вычисления τ ;

(2) -(4) - записи τ в оперативный комплекс;

(4) -(6) - переходы к полюсам алгоритма;

(6) -(7) - формирование полюсов алгоритма;

(8)-(10) - ввод, вывод и обмен с внешними накопителями, соответственно;

(II) - согласование размерностей.

Семантика большинства операций описана в [3,4] .

$\Rightarrow(\pi')$ - означает присвоение значения τ всем операндам из списка, ограниченного круглыми скобками.

$\rho_3 [\pi'_3]$ и $\Leftarrow [\pi'_3]$ - коммутаторы, реализующие переход в один из адресов списка $[\pi'_3]$ в зависимости от значения собственной переменной τ .

Для $\rightarrow [\pi'_3]$ и $\Leftarrow [\pi'_3]$ номер адреса в списке совпадает со значениями τ , причем $\Leftarrow [\pi'_3]$ выполняет и все последующие переходы. Для $\rightarrow [\pi'_3]$ и $\Leftarrow [\pi'_3]$ адрес определяется номером левого нуля или левой единицы τ , соответственно. Нумерация элементов списка начинается с нуля и происходит слева направо. $\sum \pi_{13}^0 \pi_{13}'$ - циклическое сложение τ с контрольной суммой элементов оперативного комплекса $\mathcal{K}$, начиная с π_{13}' в количестве π_{13}' .

Из операций строки (8) употребляются те, которые имеют аналог в используемой ПС ЛЯПАС. Так, для языка, описанного в [3] , имеется единственная аналогия $1: \sim 1$.

2.4. Л - программа первого уровня

В системе введенных выше обозначений α - цепочка [3] может быть обозначена одним символом $\tilde{\alpha}$. Символы $\oint \pi_{13}'$ разбивают $\tilde{\alpha}$ на предложения, номерами которых служат значения π_{13} в местах $\oint \pi_{13}'$, стоящих в начале каждого предложения. Построим, как и в [3] , ориентированный граф переходов в $\tilde{\alpha}$ следующим образом.

Вершинам графа поставим в соответствие предложения, а дугам - переходы по операторам $\rightarrow \pi_{13}'$, $\Leftarrow$, $\Leftarrow$ или по непосредственному следованию предложений, если предшествующее предложение не кончается операцией $\rightarrow \pi_{13}'$, или $\rightarrow [\pi'_3]$, или $\Leftarrow$. Участок $\tilde{\alpha}$, предшествующий первому по порядку символу $\oint \pi_{13}'$ (в том числе и пустой), назовем нулевым предложением, или вхо-

дом . Последнее предложение (в том числе и пустое) назовем выходом . В конце его ставится символ $\cdot$, который, с одной стороны, служит символом конца алгоритма, а с другой - означает операцию остановки реализации.

Отметим дуги, соответствующие переходу по оператору $\rightarrow$, и вершины, соответствующие предложениям, содержащим операцию

Пусть $\pi_{13} \in \{000, 001, \dots, 777\}$.

Л-программой первого уровня в Р-ЛЯПАСе считается выражение

$$\# \pi_{15} \pi_{15} \pi_{15} \pi_{15} \oint 0 \tilde{\alpha} \cdot ,$$

если оно удовлетворяет следующим условиям:

1. В операциях переходов встречаются только те значения π_{13}' , которые имеются в выражениях $\oint \pi_{13}'$, т.е. переходы адресуются только к существующим предложениям.

2. В свою очередь, в выражениях $\oint \pi_{13}'$, значения символа π_{13}' не должны повторяться, т.е. каждое предложение имеет свой собственный номер.

3. Вход и выход в графе переходов не должны быть отмеченными.

4. В графе переходов нет контуров, содержащих отмеченные дуги.

5. В подграфе, получаемом из графа переходов выбрасыванием отмеченных дуг, не должно существовать путей от входа к отмеченным вершинам.

6. В том же подграфе должен существовать путь от входа к любой неотмеченной вершине и путь от каждой неотмеченной вершины к выходу.

7. Выражение $\# \pi_{15} \pi_{15} \pi_{15} \pi_{15}$ называется шапкой Л-программы. Нулевой символ $\#$ есть признак начала программы; первый и второй - номер программы; третий - мощность той части оперативного комплекса $\mathcal{K}$, в которой размещается Л-программа; четвертый - максимальный номер предложения Л-программы.

Л-программа плотно упаковывается в оперативный комплекс $\mathcal{K}$, причем каждый символ Р-ЛЯПАСа занимает 9 разрядов ячейки ЦВМ. Для разных типов ЦВМ мощность Л-программы различна. Иногда удобно нарушать плотную упаковку так, чтобы некоторые символы

располагались в левом конце ячейки, оставляя предыдущую незаполненной. Договоримся в этом случае ту часть программы, которая упаковывается с левого конца ячейки, в формальном описании начинать с новой строки. В частности, тело Л-программы первого уровня Р-ЛЯПАСа, начиная с § 0, упаковывается отдельно от шапки, т.е.

$$\# \pi_{15} \pi_{15} \pi_{15} \pi_{15} \\ \hat{e} 0 \hat{e} .$$

3. Второй уровень Р-ЛЯПАСа

Строгое описание второго уровня ЛЯПАСа до сих пор не сделано. Дело в том, что при программировании на втором уровне существенную роль играет смысловое и синтаксическое согласование внешних операндов подпрограммы, входящих в перечень, с употреблением соответствующих символов в теле подпрограммы. Поскольку при этом текст подпрограммы в программе не фигурирует (более того, в программе отсутствуют какие-либо указания о смысле внешних операндов подпрограммы), синтаксический анализ Л-программы второго уровня не дает информации о том, будет ли полученная после компиляции Л-программа первого уровня удовлетворять синтаксису.

Идеи и средства второго уровня Р-ЛЯПАСа в основном совпадают с предложенными в [4]. Исключение составляет лишь способ согласования размерностей операндов. В расширении эта операция делается на первом уровне и не требует специальных средств на втором.

Ниже дано описание синтаксиса второго уровня Р-ЛЯПАСа, которое можно рассматривать как область определения функций компилятора, обеспечивающего переход со второго уровня языка на первый.

Положим, что

$$\pi_3'' \in \{ \pi_3', \# \pi_{15} \pi_{15} \pi_3' \},$$

где $\# \pi_{15} \pi_{15} \pi_3'$ - адрес дополнительного входного полюса

$\hat{e} \pi_3'$ Л-оператора второго уровня с номером $\pi_{15} \pi_{15}$. Операция α , совпадает с операцией α везде, кроме переходов. В последнем случае символ π_3' должен быть замещен на π_3'' . Появляющийся при этом переход в дополнительный полюс Л-оператора называется косвенной формой упоминания о подпрограмме (обращения к подпрограмме).

Пусть

$$\pi_3''' \in \{ \pi_3'', \hat{e} \pi_3' \}$$

$$\rho_5 \in \{ / , \tilde{\pi}_3 [\pi_3 , [\pi_3 , [\pi_6 \pi_3] , \pi_6 \pi_3] , \pi_6] , \pi_3 [\pi_6 \pi_3] , \} ,$$

$$\pi_4 , \Rightarrow \pi_4 , \Leftrightarrow \pi_4' ,] \Leftrightarrow \pi_4' , \pi_6] \Leftrightarrow \pi_4' , \pi_6 \pi_3] \Leftrightarrow \pi_4' ,$$

$$[\pi_6 \pi_3] \Leftrightarrow \pi_4' , \pi_3 [\pi_6 \pi_3] \Leftrightarrow \pi_4' , \hat{e} \pi_3'' \pi_2 , \hat{e} \pi_3''' \pi_2 ,$$

$$\# \pi_{15} \pi_{15} \pi_3' , [\tilde{\pi}_3'''] , \tilde{\pi}_3''' , \pi_{11} ; , \tilde{\pi}_{12} ; , ; , \odot ,$$

$$\odot \odot , \tilde{\pi} \Rightarrow \tilde{\pi}_{14} \odot , \Rightarrow \tilde{\pi}_{14} \odot , \tilde{\pi}_{14} \odot , \odot \} ;$$

$$\rho_6 \in \{ / , \pi_{11} , \pi_7 [, \pi_2 [\pi_6 , \pi_2 [\pi_6 \pi_3 ,] \tilde{\pi}_3 ,] \tilde{\pi}_3 [, \rho_2 ,$$

$$\rho_2 \pi_{11} , \rho_2 \pi_2 [, \rho_2 \pi_2 [\pi_6 , \rho_2 \pi_2 [\pi_6 \pi_3 , \rho_2] \tilde{\pi}_3 , \rho_2] \tilde{\pi}_3 [,$$

$$\rho_2] , \rho_4 , \rho_4 \pi_4 , \rho_4 \pi_5' , \rho_4 \pi_2 [, \rho_4 \pi_2 [\pi_6 , \rho_4 \pi_2 [\pi_6 \pi_3 ,$$

$$\Rightarrow \pi_4 , \Rightarrow \pi_5' , \Rightarrow \pi_2 [, \Rightarrow \pi_2 [\pi_6 , \Rightarrow \pi_2 [\pi_6 \pi_3 , \pi_1' \Leftrightarrow \pi_4 ,$$

$$\pi_1' \Leftrightarrow \pi_2 [, \pi_1' \Leftrightarrow \pi_2 [\pi_6 , \pi_1' \Leftrightarrow \pi_2 [\pi_6 \pi_3 , \pi_1' \Leftrightarrow \pi_5' ,$$

$$\pi_1 \hat{e} \pi_3'' , \pi_1 \hat{e} , \pi_1 \hat{e} \pi_3''' , \pi_1 \hat{e} , \rho_3 , \rho_3 [, \rho_3 [\tilde{\pi}_3'''] , \mapsto , \mapsto [,$$

$$\mapsto [\tilde{\pi}_3''' , \dagger , \hat{e} , \hat{e} , \dagger , \dagger , \dagger \pi_{11} , \dagger \tilde{\pi}_{12} , \dagger \pi_{11} , \dagger \tilde{\pi}_{12} , \dagger , \dagger \pi_{13} ,$$

$$\dagger \pi_{13} \pi_{13} , \dagger , \dagger \pi_{13} , \dagger \pi_{13} \pi_{13} , \odot , \odot \tilde{\pi} , \odot \tilde{\pi} \Rightarrow , \odot \tilde{\pi} \Rightarrow \tilde{\pi}_{14} \} ;$$

$$\alpha_2 \in \{ \alpha , \setminus (\tilde{\pi}_{13}) \Rightarrow \pi_4 , \pi_4 \Rightarrow (\tilde{\pi}_{13}) , \Rightarrow (\tilde{\pi}_{14}) \} ;$$

$$\rho \in \{ / , \rho , \pi_1 , \pi_2 , \pi_3 , \pi_4 , \pi_5 , (\tilde{\pi}_{13}) , (\tilde{\pi}_{14}) , (\pi_3''') , (\rho_5 \hat{e} \rho_6) \} .$$

Прямая форма упоминания о подпрограмме с номером $\pi_{15} \pi_{15}$ выглядит так:

$$\# \pi_{15} \pi_{15} \tilde{\rho}^{16} .$$

Здесь $\tilde{\rho}^{16}$ - перечень внешних операндов. Номер последнего эле-

мента перечня не должен быть меньше максимального номера внешнего операнда π_6 в теле подпрограммы (определение π_{16} см. ниже).

Операция второго уровня теперь есть.

$$\alpha' \in \{ \alpha_1, \# \pi_{15} \pi_{15} \beta^{16} \}.$$

Л-программа второго уровня

$$\# \pi_{15} \pi_{15} \pi_{15} \pi_{15}$$

$$\# \pi_{15} \pi_{15}$$

$$\dots$$

$$\# \pi_{15} \pi_{15}$$

$$\notin \alpha'.$$

Смысл первой строки шапки тот же, что и на первом уровне. Остальные представляют собой список номеров употребляемых Л-операторов, упорядоченный в соответствии с расположением прямых форм упоминания о них в теле Л-программы. На α - цепочки α' и α_2 наложены те же ограничения, что и на α . При этом надо учесть, что множество номеров предложений дополняется выражением $\# \pi_{15} \pi_{15} \pi_3$, а номера некоторых предложений дублируются в β^{16} .

Для описания подпрограммы вводится внешний операнд

$$\pi_{16} \in \{ \alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \eta, \psi, \chi, \lambda, \mu, \nu, \xi, \pi, \rho, \sigma \}.$$

Определим обобщенные операнды и операторы второго уровня следующим образом:

$$\pi_7 \in \{ \pi_1, \pi_{16} \}$$

$$\pi_4 \in \{ \pi_4, \pi_{16} \}$$

$$\pi_2 \in \{ \pi_2, \pi_{16} \}$$

$$\pi_5 \in \{ \pi_5, \pi_{16} \}$$

$$\pi_3 \in \{ \pi_3, \pi_{16} \}$$

$$\pi'_5 \in \{ \pi'_5, \pi_{16} \}$$

$$\pi'_3 \in \{ \pi'_3, \pi_{16} \}$$

$$\pi_6 \in \{ \pi_6, \pi_{16} \}$$

$$\pi''_3 \in \{ \pi''_3, \pi_{16} \}$$

$$\pi_7 \in \{ \pi_7, \pi_2, \pi_4, \pi_6 \}$$

$$\pi'''_3 \in \{ \pi'_3, \pi'_3, \# \pi_{15} \pi_{15} \pi'_3 \}$$

$$\pi_8 \in \{ \pi_3, \pi_5, \pi_6 \}$$

$$\pi_9 \in \{ \pi_1, \pi_2 [\pi_6], \pi_7 [\pi_6 \pi_3] \}$$

$$\pi_{10} \in \{ [\pi_3, \pi'_3] \}$$

$$\rho_1 \in \{ \rho_1, \pi_{16} \}$$

$$\pi_{11} \in \{ \pi_4, \pi_5 \}$$

$$\rho_2 \in \{ \rho_2, \pi_{16} \}$$

$$\pi_{12} \in \{ A \pi_3, B \pi_3, \pi_1, \pi_2, \pi_4, \pi_5 \}$$

$$\rho_3 \in \{ \rho_3, \pi_{16} \}$$

$$\pi_{13} \in \{ \pi_1, \pi_2 \}$$

$$\rho_4 \in \{ \rho_4, \pi_{16} \}$$

$$\pi_{14} \in \{ \pi_1, \pi_2, \pi_4 \}$$

$$\rho \in \{ \rho, \pi_{16} \}$$

$$\pi \in \{ \pi_9, \pi_9, \pi_{10} \}$$

$$\pi' \in \{ \pi_2, \pi'_5, \pi_6, \pi_9 \}$$

Синтаксис подпрограммы Л-оператора получается из синтаксиса Л-программы второго уровня заменой всех операндов и операторов на обобщенные. Соответствующие α - цепочки α' и α_2 должны отвечать уже описанным условиям. При этом необходимо учесть, что множество выходов из подпрограммы дополняется адресами π_{16} или обобщенной операцией π_{16} .

Синтаксис второго уровня Р-ЛЯПАСа можно содержательно интерпретировать следующим образом. Пусть необходимо превратить в подпрограмму некоторую Л-программу первого или второго уровня. При этом разрешается обозначить греческой буквой любую часть последовательности основных символов Р-ЛЯПАСа, представляющей Л-программу, лишь бы обозначаемая цепочка не пересекалась с прямой формой упоминания об Л-операторах. Греческие буквы при этом должны быть использованы подряд, а смысл их содержательно описан в инструкции к подпрограмме.

Из описания ρ_5 и ρ_6 видно, что запрещено разрушать только выражение $\# \pi_{15} \pi_{15} \pi'_3$, которое не может разделяться даже пустым оператором.

При использовании подпрограммы в перечень подставляются все без исключения значения ее внешних операндов в порядке следования последних, причем если значение описывается одним сим-

волом, то разрешается не заключать его в круглые скобки.

Если внешний операнд подпрограммы является адресом дополнительного выхода, то этот факт отмечается подстановкой на соответствующее место полной метки π'_3 или $\# \pi_{15} \pi_{15} \pi'_3$.

Допустимо также использование символа π'_3 , если программист уверен, что дополнительный выход из подпрограммы реализуется на ее собственной глубине, а не в какой-то ее внутренней подпрограмме.

Широкие возможности, открываемые синтаксисом второго уровня, в основном позволяют строить сравнительно простую и быст-родействующую процедуру компиляции (см., например, [4]).

Л И Т Е Р А Т У Р А

1. Применение вычислительных машин для проектирования цифровых устройств. М., "Сов.радио", 1968.
2. А.К. ПОЛЯКОВ. Система автоматизации моделирования ЦВМ. Цифровая вычислительная техника и программирование, М., "Сов.радио", 1968, стр. 18-35.
3. А.Д. ЗАКРЕВСКИЙ. Описание языка ЛЯПАС. - Логический язык для представления алгоритмов синтеза релейных устройств, М., "Наука", 1966, стр. 7-39.
4. А.Д. ЗАКРЕВСКИЙ. Алгоритмический язык ЛЯПАС и автоматизация синтеза дискретных автоматов. Томск, Изд-во ТГУ, 1966.
5. С.В. ПИСКУНОВ, С.Н. СЕРГЕЕВ, Б.А. СИДРИСТЫЙ. Язык для описания алгоритмов функционирования ЦВМ. - "Вычислительные системы", Новосибирск, "Наука" СО, 1969, вып. 34, стр. 5-18.
6. И.В. ИЛОВАЙСКИЙ. Построение функциональной схемы ЦВМ по описанию процесса ее работы. Там же, стр. 38-50.
7. T.D. FRIDMAN, JANG SIH-CHIN. Methods used in an automatic logic design generator (ALERT). - IEEE Trans. Comp., 1969, v. 18, N 17, p. 593-614.
Э.-И. - ВТ, 1970, № 1, реф. 5.

Поступила в редакцию
13. XI. 70 г.