АВТО-
-АНАЛИТИК

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ
Р С Ф С Р

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Е.А.Арайс, Г.В.Сибиряков

АВТО-АНАЛИТИК

Новосибирск 1973

Авто-Аналитик,
Арайс Е.А., Сибиряков Г.В.,
НГУ, 1973, 1-285.

В настоящей работе приводится описание системы программирования задач, связанных с аналитическими выкладками. Излагаются теоретические основы системы, получившей общее название Авто-Аналитик, сведения о реализации Авто-Аналитика на ЭЦВМ БЭСМ-6, а также описание входного языка и библиотеки операторов. Последние составляют основу практического применения Авто-Аналитика.

Книга предназначена как для профессиональных программистов, так и для математиков, работа которых связана с громоздкими алгоритмизируемыми аналитическими выкладками.

П Р Е Д И С Л О В И Е

Возрастающая мощность электронных цифровых вычислительных машин дает возможность применять их в новых областях науки и техники. В последние годы заметно возрос интерес к реализации на ЭЦВМ алгоритмов аналитического и численно-аналитического характера. Настоящая книга посвящена описанию разработанной в НИИ прикладной математики и механики при ТГУ системы программирования Авто-Аналитик для ЭЦВМ БЭСМ-6.

В главе I излагается алгебраическая основа Авто-Аналитика.

В главе II рассматривается реализация Авто-Аналитика на ЭЦВМ БЭСМ-6.

В главе III описывается входной язык.

В главе IУ приводится описание библиотеки операторов Авто-Аналитика.

Авторы выражают глубокую благодарность руководителю настоящей разработки академику Н.Н. Яненко за постоянную поддержку, внимание, определение класса задач и проблем, Шапееву В.П., разработавшему теоретические основы важнейшего приложения Авто-Аналитика-реализации метода Картана и активно содействовавшему развитию самой системы, а также Быкову Н.И., Гельфману Б.Ш., Карначуку В.И., Магазинникову Л.И., Шутенкову А.В., принимавшим участие в разработке и отладке системы Авто-Аналитик и ее приложений.

В В Е Д Е Н И Е

Характерной чертой современной эпохи является широкое применение универсальных цифровых вычислительных машин в науке и практике. Обычно использование вычислительных машин дает определенный экономический эффект, освобождая людей от малопроизводительных ручных расчетов. ЭЦВМ позволяют ставить и решать задачи, которые до их появления считались неразрешимыми. Появление и развитие средств вычислительной техники послужило толчком к развитию новых методов и направлений в математике, связанных с выполнением большого количества арифметических операций. Однако следует признать, что в настоящее время ЭЦВМ используются при решении ограниченного круга задач. Основная трудность при работе с ЭЦВМ заключается в необходимости перевода методов и алгоритмов в коды машинного языка. Естественным способом преодоления этой трудности является развитие систем автоматизации программирования. Большие успехи достигнуты в области автоматизации программирования численного анализа. В последние годы были предприняты многочисленные попытки разработки систем программирования аналитических (или символических) методов, которые играют важную роль в математике и ее приложениях. Однако эти системы не получили широкого распространения, что объясняется, по-видимому, следующими причинами: а) современные вычислительные машины ориентированы на решение задач арифметического характера; б) алгоритмы, связанные с аналитическими выкладками, обычно более сложны, чем алгоритмы чисто вычислительного характера; в) и, наконец, несравненно более сложную структуру, чем числовые массивы, имеет сам объект, подлежащий обработке этими алгоритмами, - аналитические выражения или формулы.

Системы автоматизации программирования задач, связанных с аналитическими выкладками, можно разделить на проблемно-ориентированные системы и системы общего назначения.

К первым относятся программы аналитического дифференцирования [1, 2], система оперирования с полиномами ALPAC [3, 4, 5] и полиномиальный прораб [6]. Полиномиальные и некоторые матричные операции может производить алгебраическо-дифференциальная программа АДП [7]. Операции с рядами Фурье производит Кэмбриджская алгебраическая система (Cambridge

algebra system) [8, 9] , основным целевым назначением которой является перенесение на ЭЦВМ некоторых аналитических методов небесной механики [10] .

Расширениями распространенных алгоритмических языков (Алгола и Фортрана) для решения задач арифметического характера явились языки FORMAC (FORmula MAnipulation Compiler) и формульный Алгол (Formula algol) [11, 12] .

Формульный Алгол есть расширение Алгола, позволяющее программисту включать собственные процедуры обработки символических выражений. FORMAC является расширением языка Fortran - IV и допускает решение задач численно-аналитического характера. Для проведения аналитических выкладок в системе FORMAC предусмотрена возможность приведения подобных в классе арифметических операций, замена одной буквы другой, раскрытия скобок, определения числа членов и так далее.

Возможно программирование аналитических задач с привлечением языков IPL-V [13] , Lisp [14, 15] .Эти языки предназначены для решения информационных задач. Объекты обработки (операнды) имеют списковую структуру, причем числовое значение элементов списка можно не определять заранее. Символические задачи рассматриваются как частный случай более широкого класса информационных задач. Пригодны для проведения аналитических выкладок алгоритмические языки ЭПСИЛОН [16] и ЛЯПАС [17] , так как они также работают с объектами нечисловой природы.

Следует отметить системы обеспечения диалогового режима работы ЭЦВМ с человеком для решения аналитических задач. К таким системам относятся, например, язык "Аналитик", реализованный на специализированной вычислительной машине "Мир" [18] , и система разговорного программирования "Сириус" [19] , предназначенная для ЭЦВМ типа М-20.

Настоящая работа является описанием системы программирования Авто-Аналитик, разработанной для наиболее мощной отечественной вычислительной машины БЭСМ-6. Была предпринята попытка реализации Авто-Аналитика на ЭЦВМ типа М-20, однако, вследствие малой мощности этих машин, система оказалась непригодной для решения сколько-нибудь сложных практических задач.

Описание системы Авто-Аналитик не носит чисто инструктивного характера. Так, например, рассматриваемая в главе I алгебраи-

ческая основа Авто-Аналитика развернута несколько более широко, чем требуется для понимания работы системы. Такое тщательное исследование общих подходов вызваны тем, что авторы отнюдь не считают Авто-Аналитик полностью завершенным универсальным аппаратом. Сложность и разнообразие аналитических методов и алгоритмов таковы, что никакая система программирования не может быть, по мнению авторов, универсальной. Весьма подробно во второй главе описывается административная система Авто-Аналитика. Эта подробность не является излишней, так как при решении новых классов задач может возникнуть (и уже возникала) необходимость изменения самых основных структур Авто-Аналитика, обеспечивающих распределение памяти, организацию программ, хранение информации и так далее.

Наиболее характерной особенностью входного языка Авто-Аналитика (по сравнению с другими языками аналитического программирования) является то, что он был разработан с учетом достаточно большого опыта ручного программирования сложных задач аналитического характера.

Основу практического использования Авто-Аналитика составляет библиотека операторов, представленная в главе IУ. Пользователю для решения небольших задач зачастую оказывается достаточным знакомство только с этой главой. Основное расширение Авто-Аналитика производится путем накопления новых операторов библиотеки. Структура входного языка и транслятора с него таковы, что включение новых программ в библиотеку операторов происходит автоматически.

Практические приложения Авто-Аналитика являются достаточно сложными самостоятельными разработками и поэтому в данной книге либо не рассматриваются вообще, либо рассматриваются на уровне общей информации.

К таким приложениям относятся реализация метода внешних форм Картана, задачи теории цепей, небесной механики и теоретической механики.

ГЛАВА I

АЛГЕБРАИЧЕСКАЯ ОСНОВА АВТО-АНАЛИТИКА

§ I. АЛФАВИТ

Под алфавитом в Авто-Аналитике понимается система обозначений, применяемых для записи аналитических выражений или формул – основного объекта обработки. Мы предполагаем, что алфавит Авто-Аналитика Г состоит из некоторых графических знаков, именуемых элементарными символами. В отличие от других систем программирования мы не можем перечислить полностью набор этих символов, так как он существенно зависит от класса решаемых аналитических задач. Окончательная конкретизация алфавита Г производится математиком, использующим Авто-Аналитик для решения своей конкретной задачи. В общем случае в этом необходимости нет. Значительная часть блоков Авто-Аналитика не использует конкретного вида алфавита Г, а предполагает лишь следующее:

Алфавит Г есть объединение трех попарно непересекающихся множеств: множества специальных символов θ , множества связей Σ , множества букв П. Множество θ содержит четыре символа: граничный символ Δ , пустой символ $\odot$, раскрывающую скобку (и закрывающую скобку). Во множестве Σ выделены два подмножества Σ_a и $\Sigma_к$, элементы которых называются ассоциативными и коммутативными связями соответственно. Множество П есть объединение трех попарно непересекающихся множеств: множества постоянных букв Ω , множества переменных букв Φ , множества чисел R . Весь алфавит Г линейно упорядочен некоторым отношением порядка $\leqslant$, причем $\Delta \leqslant \odot \leqslant$ ($\prec$) и каждая связь в смысле этого порядка сильнее любого специального символа и слабее любой буквы.

Разбиение алфавита Г на множества θ , Σ и П отвечает различной синтаксической роли математических символов из этих множеств. А именно, мы неявно предполагаем, что во множество связей отнесены знаки логических, теоретикомножественных, арифметических, алгебраических и т.п. отношений и операций (например, знаки =, $\in$, +, $\int$). Ассоциативные связи суть знаки ассоциативных операций. Коммутативные связи суть знаки коммутатив-

ных операций и симметричных отношений. Во множество θ включены элементарные символы, играющие вспомогательную роль при записи формул. Граничный символ Δ играет роль пробела между соседними формулами. Пустой символ $\odot$ употребляется для вычеркивания тех символов в формулах, которые становятся ненужными в дальнейшем счете. Раскрывающая и закрывающие скобки имеют обычный смысл, то есть указывают порядок выполнения операций. Во множество букв П должны быть отнесены все "предметные" знаки, то есть обозначения участвующих в решении задачи множеств и их элементов. При этом обозначения конкретных вещественных чисел (допускающих приближенное представление на "машинном" языке ЭЦВМ) образуют множество R. Множество переменных букв Ф состоит из символов, обозначающих так называемые "общие элементы" в правилах преобразования аналитических выражений. Так, правило дифференцирования дроби

$$\frac{\partial}{\partial t}\frac{f}{g} = \left(\frac{\partial f}{\partial t}g - f\frac{\partial g}{\partial t}\right)\cdot g^{-2},$$

которое в Авто-Аналитике можно записать в виде

$$\Delta(\bar{f}/\bar{g})\partial\bar{t} = (\bar{f}\partial\bar{t}\times\bar{g} + \bar{f}\times\bar{g}\partial\bar{t}\times\bar{1})/\bar{g}\uparrow\bar{2}\,\Delta,$$

содержит в себе три переменных буквы $\bar{f}$, $\bar{g}$ и $\bar{t}$, два числа $\bar{1} = -1$ и 2, а также связи =, +, x, /, $\uparrow$, ∂ (знаки /, $\uparrow$, ∂ суть соответственно знаки операций деления, возведения в степень и дифференцирования). Более детальное описание роли переменных букв в Авто-Аналитике см. в § 10. Наконец, множество постоянных букв Ω составляют обозначения всех прочих "объектов" задачи математика. В частности, в Ω должны быть отнесены латинские и греческие буквы, а также такие символы, как ∞, $\emptyset$, истина, функция, множество, топология и т.п., если конечно они участвуют в решении задачи.

Обозначения конкретных функций (например, sin, $\ln$, max, ...) могут быть включены либо во множество связей Σ, либо во множество постоянных букв Ω. В последнем случае функциональная зависимость указывается при помощи особой связи $*$. В Σ особенно удобно включать обозначения функций двух переменных.

Учитывая эти замечания о содержательной интерпретации элементарных символов, мы можем в качестве примера указать один из многих допустимых вариантов конкретной фиксации алфавита Г. А именно, во всех примерах данной главы мы будем считать, что множество Σ состоит из связей

$$\Leftrightarrow, \Rightarrow, \vee, \wedge, \neg, \in, \notin, \subset, \setminus, \cup, \cap, =, \neq, \leq, <, +, \int, \times, /, \log, \dagger, \partial, *, ;,$$

которые соответственно суть знаки эквивалентности, импликации, дизъюнкции, конъюнкции, отрицания, принадлежности, непринадлежности, включения, разности множеств, объединения, пересечения, равенства, неравенства, порядка, строгого порядка, сложения, интеграла (выражения

$$\int f dt, \quad \int_T f dt, \quad \int_a^b f dt$$

можно записывать, например, в виде $f\int t$, $f\int t;T$, $f\int t;a;b$), умножения, деления логарифма (вместо $\log_a b$ мы пишем $a\log b$), возведения в степень (вместо a^2 мы пишем $a\uparrow 2$), дифференцирования (вместо $\partial f/\partial t$ или f'_t мы пишем $f\partial t$), функциональной зависимости (вместо $\sin t$, $f(t)$ и $f(t,S)$ мы пишем $\sin * t$, $f * t$, $f * t; S$) и разделительный знак (играющий вспомогательную роль). Из этих связей естественно объявить ассоциативными связи $\vee$, $\wedge$, $\cup$, $\cap$, $+$, $\times$, $*$, а коммутативными – связи $\Leftrightarrow$, $\vee$, $\wedge$, $\cup$, $\cap$, $=$, $+$, $\times$.

Далее, во множество постоянных букв Ω мы включим символы ∞, $\varnothing$, прописные и строчные буквы латинского и греческого алфавитов и их произвольные конечные последовательности, а во множество переменных букв Φ – строчные латинские буквы с чертой наверху: $\bar{a}$, $\bar{b}$, ...

Отношение линейного порядка в Г определим следующим образом. Первым, вторым, третьим и четвертым элементами алфавита Г объявляем символы Δ, $\odot$, (и). Затем идут связи в том порядке, как они расположены в приведенном выше списке. Первым и вторым элементами множества Π объявляем символы ∞ и $\varnothing$. Латинские и греческие буквы и их произвольные конечные последовательности упорядочиваем естественным лексикографическим способом,

причем каждую латинскую букву будем считать слабее любой греческой буквы и каждую прописную букву – слабее соответствующей строчной. Переменные буквы из Φ и числа из R также располагаются в естественном порядке, причем каждая переменная буква сильнее любой постоянной буквы и слабее любого числа.

§ 2. ФОРМУЛЫ

Прежде чем конструировать алгоритмы для преобразования формул, необходимо составить четкое представление о том, что представляет собой формула. Под этим термином мы будем понимать следующее:

ОПРЕДЕЛЕНИЕ I. Формулой называется всякая конечная последовательность элементарных символов

$$\Delta\ \alpha_1 \alpha_2 \dots \alpha_n\ \Delta\ j \qquad (1)$$

удовлетворяющая четырем условиям:

(a) число n неотрицательно (если $n = 0$, то последовательность (I) имеет вид $\Delta\,\Delta$ и называется пустой формулой) и для каждого $i = 1, 2, \dots, n$ символ α_i отличен от символов Δ и Θ ;

(b) в любом отрезке $\Delta\ \alpha_1 \alpha_2 \dots \alpha_i$, $1 \le i \le n$, раскрывающих скобок не меньше, чем закрывающих, а во всей последовательности (I) раскрывающих скобок столько же, сколько и закрывающих;

(c) в последовательности (I) перед каждой буквой и перед каждой раскрывающей скобкой находится либо связь, либо граничный символ, либо раскрывающая скобка;

(d) в последовательности (I) после каждой буквы и после каждой закрывающей скобки находится либо связь, либо граничный символ, либо закрывающая скобка.

Примерами формул могут служить следующие последовательности элементарных символов (отделенные друг от друга запятой):

$$\Delta\ \cos^*(\alpha+\beta) = \cos^*\alpha \times \cos^*\beta + \sin^*\alpha \times \sin^*\beta \times \bar{1}\ \Delta\ ,$$

$$\Delta\ ((a+b)\times c)\,!\,(a/b)\ \Delta\ , \qquad (2)$$

$\Delta\,(a \cup b) \cap c = a \cap c \cup b \cap c\;\Delta$, $\Delta\; A \subset B \vee B \subset A \Rightarrow A = B\;\Delta$,

$\Delta + + + \Delta$, $\Delta\,(((x;\phi)))\;\Delta$,

$\Delta\,((((+)/)+)x)\;\partial(x(/(+(\Leftrightarrow)=)\phi)+())\,\Delta$.

Напротив, выражения

$\Delta \int_{\tau} f(t)\,dt\;\Delta$, $\Delta \lim_{t \to o} \frac{\sin t}{t} = 1\;\Delta$,

$\Delta\;\Theta\,\Delta$, $\Delta\,(a+b)(a+b)\,\Delta$, $\Delta\,\mathcal{U} + \mathcal{B}\,\Delta$

формулами не являются, так как они либо "двумерны", либо не удовлетворяют одному из условий определения I. Последнее выражение станет формулой, если готические буквы присоединить к алфавиту Г.

Читатель по-видимому уже заметил, что многие формулы Авто-Аналитика "не имеют смысла". Однако это не должно его волновать. Авто-Аналитик не нуждается в том, чтобы все доступные ему формулы имели ту или иную смысловую интерпретацию. Дело в том, что основу Авто-Аналитика составляют правила формального преобразования формул. Эти правила определяются так, что для их применения не обязательно существование смысловой интерпретации преобразуемых формул. Однако в этих правилах и соответствующих им алгоритмах используется каждое из условий (a), (b), (c) и (d) определения I. Следовательно, данное определение описывает максимальное семейство формул, доступных Авто-Аналитику. Какие-либо дополнительные ограничения, апеллирующие к "здравому смыслу" (и требующие, например, исключить такие формулы, как $\Delta + + + \Delta$, $\Delta\,(\;)\,\Delta$ и т.п.) по сути дела не нужны. С другой стороны, не следует забывать, что "бессмысленные" формулы могут получить вполне законное смысловое содержание при иной интерпретации элементарных символов. Так, формула $\Delta\,(\cos + \sin) * t\,\Delta$ бессмысленна с точки зрения школьной тригонометрии, но в рамках общей теории отображений может быть воспринята, как запись значения в точке t оператора, представляющего собой сум-

му операторов cos и sin. Наконец, при решении конкретных задач всегда можно ограничиваться классами "осмысленных" формул. При разумном использовании формального аппарата Авто-Аналитика бессмысленные формулы никогда не появляются в качестве окончательных результатов решения задачи.

Как уже упоминалось, граничный символ Δ играет роль пробела между формулами. В соответствии с этой смысловой нагрузкой символа Δ при записи группы формул между соседними формулами вместо двух граничных символов разрешается писать только один и не рекомендуется применять какие-либо дополнительные средства (пробелы в печати, запятые и т.п.) для разделения этих формул. Например, последовательность

$$\Delta\ 2\times f * t\int t\ \Delta\Delta\ a/b\ \Delta\ a/b/c\ \Delta(a/b)/$$
$$c\ \Delta + \times t\ \Delta\ 25\ \Delta \qquad (3)$$

содержит 7 формул, из которых первая есть $\Delta\ 2\times f * t\int t\ \Delta$, вторая - $\Delta\Delta$, третья - $\Delta\ a/b\ \Delta$ и т.д.

Условимся формулы или иные последовательности элементарных символов для сокращения письма обозначать прописными буквами русского алфавита (с индексами или без них). В связи с этим мы часто будем писать выражения вида

$$Ш = \Delta\ \alpha_1\alpha_2\dots\alpha_n\ \Delta,$$

понимая под этим следующее: Формула или, быть может, просто последовательность $\Delta\ \alpha_1\alpha_2\dots\alpha_n\ \Delta$ обозначена через Ш. Последовательность $\alpha_1\alpha_2\dots\alpha_n$, полученная из формулы Ш отбрасыванием обоих окаймляющих ее граничных символов, всегда обозначается через $\widetilde{Ш}$. Далее, если Ш и Щ суть обозначения последовательностей $\alpha_1\alpha_2\dots\alpha_n$ и $\beta_1\beta_2\dots\beta_m$ соответственно, то выражения

$$Ш\ Щ,\ Ш\ \beta_1\beta_2\dots\beta_m,\ \alpha_1\alpha_2\dots\alpha_n\ Щ$$

представляют собой одну и ту же последовательность

$$\alpha_1\alpha_2\dots\alpha_n\beta_1\beta_2\dots\beta_m.$$

Аналогично следует понимать выражения вида $Ш\ \delta\ Щ$, $\delta\ Щ\ Ш$, $Ш\ Ш\delta\delta\ Щ$, ..., где δ, σ, ... суть элементарные символы.

Используя отношение порядка в алфавите Г, семейство всех формул можно линейно упорядочить обычным лексикографическим

способом:

ОПРЕДЕЛЕНИЕ 2. Будем считать, что первая из формул

$$Ш = \Delta\ \alpha_1 \alpha_2 \ldots \alpha_n \Delta \quad , \quad Щ = \Delta\ \beta_1 \beta_2 \ldots \beta_m \Delta$$

не сильнее второй, если выполнено одно из следующих двух условий:

($а$) Формула $Щ$ представима в виде

$$Щ = \Delta\ \tilde{Ш}\ \beta_{n+1} \beta_{n+2} \ldots \beta_m \Delta ,$$

т.е. $n \le m$ и $\alpha_i = \beta_i$ для всех $i = 1, 2, \ldots, n$.

($б$) Для некоторой последовательности Э элементарных символов

$$Ш = \Delta\ Э\ \alpha_j \alpha_{j+1} \ldots \alpha_n \Delta \quad , \quad Щ = \Delta\ Э\ \beta_j \beta_{j+1} \ldots \beta_m \Delta ,$$

причем символ α_j слабее символа β_j ; иначе говоря, существует такой индекс j , $j \le n$ и $j \le m$, что $\alpha_i = \beta_i$ для всех $i = 1, 2, \ldots, j-1$ и α_j слабее символа β_j .

Например, среди четырех формул

$$\Delta\ a \times c\ \Delta\ a \times c + b\ \Delta\ a \times b + b\ \Delta\ a \times c + a\ \Delta \qquad (4)$$

третья слабее первой, первая слабее четвертой, четвертая слабее второй.

Пустая формула слабее любой другой формулы. Она является единственным минимальным элементом в семействе всех формул. В семействе всех непустых формул минимальных и максимальных элементов нет, поскольку для любой непустой формулы $Щ$ формула Δ ($\tilde{Щ}$) Δ слабее, а формула $\Delta\ \tilde{Щ}\ \sigma\ \Delta$, где $\sigma \in \Sigma$, сильнее исходной формулы $Щ$.

§ 3. ГЛАВНЫЕ СВЯЗИ

ОПРЕДЕЛЕНИЕ 3. Рассмотрим множество Σ ($Ш$) всех связей α_i , $1 \le i \le n$, до каждой из которых в формуле

$$Ш = \Delta\ \alpha_1 \alpha_2 \ldots \alpha_n \Delta \qquad (5)$$

раскрывающая скобка встречается столько же раз, сколько и закрывающая скобка. Если множество Σ ($Ш$) не пусто, то его минимальный элемент называется главной связью формулы $Ш$. Если Σ ($Ш$) $= \varnothing$, то для удобства дальнейшего изложения мы условно объявим главной связью формулы $Ш$ какую-нибудь букву, например символ ∞ (и будем считать его слабейшим элементом во множестве $П$).

В качестве иллюстрации рассмотрим формулу

$$\text{Ш} = \Delta\ \text{f}*(t\,\text{f}2);(t/s)\,\partial t\,\text{f}(2+\bar{1}\times t)\,\text{S}(t+ s+z)\times \text{f}*t\,;\,\text{S}\,\Delta\ . \tag{6}$$

Множество Σ (Ш) в этом случае содержит связи $*$, $;$, ∂ , $/$, $\int$, f , x . Слабейшей среди них является связь / . Стало быть, главной связью формулы (6) служит знак деления. Аналогично легко увидеть, что главные связи формул (2), (3) и (4) суть соответственно связи $=$, f , $=$, $\Rightarrow$, $+$, ∞ , ∂ , $\int$, ∞ , /, /, /, +, ∞ , x, +, +, + .

Рассмотрим произвольную непустую формулу (5) с главной связью σ и допустим сначала, что $\sigma = \infty$. Тогда символ α_1 в формуле Ш не может быть связью. В силу условия (δ) определения I символ α_1 не может быть закрывающей скобкой. Следовательно, α_1 есть либо буква, либо раскрывающая скобка.

Если $\alpha_1 \in \Pi$, то согласно определению формулы символ α_2 есть связь или Δ . Но если бы α_2 было связью, то главная связь σ формулы Ш не совпадала бы с символом ∞ . Значит, $\alpha_2 = \Delta$ и формула Ш имеет вид $\text{Ш} = \Delta\,\alpha_1\,\Delta$.

Пусть $\alpha_1 = ($. Обозначим через j наименьший из всех таких индексов i, $1 \le i \le n$, что в отрезке $\alpha_1\alpha_2\ldots\alpha_i$ раскрывающих скобок столько же, сколько и закрывающих. Индекс j очевидно существует и $1 < j \le n$. Символ α_j есть скобка, поскольку иначе в отрезке $\alpha_1\alpha_2\ldots\alpha_{j-1}$ обе скобки встречались бы одинаковое количество раз вопреки выбору индекса j . Но раскрывающей скобкой символ α_j быть не может, ибо в этом случае отрезок $\alpha_1\alpha_2\ldots\alpha_{j-1}$ содержал бы раскрывающих скобок на одну больше, чем закрывающих скобок, вопреки условию (β) определения I. Следовательно, $\alpha_j =)$. В силу того же условия (β) символ α_{j+1} отличен от закрывающей скобки. Связью он также быть не может благодаря равенству $\sigma = \infty$. Значит, согласно условию (α) определения I, $\alpha_{j+1} = \Delta$, т.е. $j = n$ и формула Ш имеет вид

$$\text{Ш} = \Delta\,(\,\alpha_2\alpha_3\ldots\alpha_{n-1}\,)\,\Delta = \Delta\,(\,\tilde{\text{Ш}}\,)\,\Delta$$

Здесь последовательность

$$\tilde{\text{Ш}} = \Delta\,\alpha_2\alpha_3\ldots\alpha_{n-1}\,\Delta$$

является формулой, поскольку для нее условие (β) определения

I следует из выбора индекса j, оказавшегося равным n, а остальные требования этого определения выполнены благодаря тому, что они выполнены для формулы Ш.

Допустим теперь, что $\sigma \leq \infty$. Согласно определению главной связи существуют такие индексы i, $1 \leq i \leq n$, что $\alpha_i = \sigma$ и в отрезке $\alpha_1 \alpha_2 \dots \alpha_{i-1}$ формулы Ш раскрывающих скобок столько же, сколько и закрывающих. Обозначим первый из таких индексов через i_1, второй – через i_2, ... , последний – через i_{s-1}. Тогда $s \geq 2$ и

$$Ш = \Delta \, \tilde{Ш}_1 \, \sigma \, \tilde{Ш}_2 \, \sigma \quad \dots \quad \sigma \, \tilde{Ш}_s \, \Delta \,, \qquad (7)$$

где через $Ш_1$, $Ш_2$, ... , $Ш_s$ обозначены соответственно последовательности

$$\Delta \, \alpha_1 \alpha_2 \dots \alpha_{i_1-1} \, \Delta \, \alpha_{i_1+1} \, \alpha_{i_1+2} \dots \alpha_{i_2-1} \, \Delta \dots$$

$$\dots, \, \Delta \, \alpha_{i_{s-1}+1} \, \alpha_{i_{s-1}+2} \dots \alpha_n \, \Delta \,.$$

До каждого отрезка $\tilde{Ш}_\kappa$, $\kappa = 1, 2, \dots s$, и после него в формуле Ш количество раскрывающих скобок равно количеству закрывающих скобок. Отсюда нетрудно заключить, что все последовательности $Ш_\kappa$ суть формулы. Главные связи этих формул отличны от главной связи σ формулы Ш согласно выбору индексов i_κ и не могут быть слабее σ в силу очевидного включения $\Sigma(Ш_\kappa) \subset \Sigma(Ш)$, $\kappa = 1, 2, \dots, s$.

Тем самым доказано

ПРЕДЛОЖЕНИЕ 1. Для любой непустой формулы Ш справедливо одно (и только одно) из следующих трех утверждений:

(a) Главная связь формулы Ш равна ∞ и существует такая буква α, что $Ш = \Delta \, \alpha \, \Delta$.

(b) Главная связь формулы Ш равна ∞ и существует такая формула $Ш$, что $Ш = \Delta \, (Ш) \, \Delta$.

(c) Главная связь σ формулы Ш отлична от ∞ и существуют формулы $Ш_1$, $Ш_2$, ... , $Ш_s$, $s \geq 2$, главные связи которых сильнее, чем связь σ, такие, что формула Ш имеет вид (7).

Отсюда индукцией по числу членов в формуле выводится

ПРЕДЛОЖЕНИЕ 2. Для любой формулы Ш найдется конечная последовательность формул $Ш_1$, $Ш_2$, ... , $Ш_m$, такая, что $Ш_m = Ш$ и для каждого $\kappa = 1, 2, \dots, m$ выполнено одно из четырех условий:

(a) Формула $\mathbb{Ш}$ пуста или имеет вид $\Delta\,\alpha\,\Delta$, где $\alpha \in \Pi$.

(b) Существует такой индекс i , $1 \leq i < K$, что $\mathbb{Ш}_K = \Delta\,(\,\widetilde{\mathbb{Ш}}_i\,)\,\Delta$.

(c) Главная связь σ формулы $\mathbb{Ш}_K$ ассоциативна, и существуют индексы $i < K$ и $j < K$, такие, что $\mathbb{Ш}_K = \Delta\,\widetilde{\mathbb{Ш}}_i\,\sigma\,\mathbb{Ш}_j\,\Delta$, причем главные связи формул $\mathbb{Ш}_i$ и $\mathbb{Ш}_j$ не слабее связи σ .

(d) Главная связь σ формулы $\mathbb{Ш}_K$ неассоциативна, отлична от ∞ и существуют такие индексы $i_1 < K$, $i_2 < K$, ... , $i_S < K$, $S \geqslant 2$, что

$$\mathbb{Ш}_K = \Delta\,\widetilde{\mathbb{Ш}}_{i_1}\,\sigma\,\widetilde{\mathbb{Ш}}_{i_2}\,\sigma\,\ldots\,\sigma\,\widetilde{\mathbb{Ш}}_{i_S}\,\Delta, \qquad (8)$$

и связь σ слабее главных связей формул

$$\mathbb{Ш}_{i_1}\ ,\ \mathbb{Ш}_{i_2}\ ,\ \ldots\ ,\ \mathbb{Ш}_{i_S}\ . \qquad (9)$$

ОПРЕДЕЛЕНИЕ 4. Мы говорим, что последовательность $\mathbb{Ш}_1$, $\mathbb{Ш}_2$, ... , $\mathbb{Ш}_m$, о которой идет речь в предложении 2, является процессом построения формулы $\mathbb{Ш}$. Мы называем этот процесс полустрогим, если для всякого индекса $i < m$ формула $\mathbb{Ш}_i$ может быть использована при конструировании некоторой формулы $\mathbb{Ш}_K$, $i < K \leq m$, по методу условий (b), (c) и (d) предложения 2. Мы именуем процесс построения формулы $\mathbb{Ш}$ строгим, он перестает быть процессом построения формулы $\mathbb{Ш}$ после удаления из него хотя бы одной формулы. Мы говорим, что строгий процесс построения формулы $\mathbb{Ш}$ является ультрастрогим, если не существует такой перестановки его членов, после которой он оказался бы нестрогим процессом построения формулы $\mathbb{Ш}$.

Предложение 2 утверждает, что всякая формула $\mathbb{Ш}$ обладает (и как легко видеть, всегда не единственным) процессом построения. Исходя из данного процесса $\mathbb{Ш}_1$, $\mathbb{Ш}_2$, ... , $\mathbb{Ш}_m$ построения формулы $\mathbb{Ш}$ нетрудно получить строгий (а значит и полустрогий) процесс ее построения. Для этого достаточно данный процесс преобразовать по следующему алгоритму:

1). Исходный процесс построения формулы $\mathbb{Ш}$ обозначим через A и параметр j положим равным m .

2). Если последовательность формул, полученная из A выбрасыванием формулы $\mathbb{Ш}_j$, является процессом построения формулы $\mathbb{Ш}$, то через A обозначаем этот новый процесс. Иначе A не меняем.

3). Если $j > 1$, то уменьшаем j на I и переходим на блок 2. Иначе конец.

Итак, всякий процесс построения формулы $\mathbb{Ш}$ содержит в себе

последовательность, являющуюся строгим процессом построения формулы Ш. Если формула Ш обладает единственным строгим процессом построения, то этот процесс – ультрастрогий. В противном случае, произведя надлежащую перестановку членов в каком-либо строгом процессе построения формулы Ш и затем применив описанный выше алгоритм, мы получим новый строгий процесс построения формулы Ш, который содержит меньше членов, чем исходный процесс. Отсюда следует, что всякая формула обладает хотя бы одним ультрастрогим процессом построения.

ПРИМЕРЫ. Для пустой формулы единственным строгим процессом ее построения служит одночленная последовательность, образованная одной этой формулой. То же самое справедливо для формулы $\Delta\alpha\Delta$, где $\alpha\in\Pi$. Если $\sigma\in\Sigma$, то единственным строгим процессом построения формулы $\Delta\sigma\Delta$ служит двухчленная последовательность формул $\Delta\Delta\sigma\Delta$. Можно привести и другие примеры формул с единственными строгими процессами построения. Таковы, например, формулы

$$\Delta a+a\Delta+\times\dagger\Delta a\dagger a\dagger a\dagger a\ \Delta((a))\Delta(a+a\times a)\dagger(a+a\times a)\Delta$$

Однако, как правило формула обладает несколькими строгими и даже ультрастрогими процессами построения. Так, для формулы

$$\Delta(a\dagger c)\dagger(a\dagger c)+b\Delta$$

можно написать 11 ультрастрогих процессов ее построения, два из которых таковы:

$$\Delta a\Delta b\Delta c\Delta a\dagger c\Delta(a\dagger c)\Delta(a\dagger c)\dagger(a\dagger c)\Delta(a\dagger c)\dagger(a\dagger c)+b\Delta$$

$$\Delta a\Delta c\Delta a\dagger c\Delta(a\dagger c)\Delta(a\dagger c)\dagger(a\dagger c)\Delta b\Delta(a\dagger c)\dagger(a\dagger c)+b\Delta.$$

Аналогично для формулы

$$\Delta\ a\times b\times c\times d+a/b/c\Delta$$

ультрастрогими процессами ее построения служат, например, последовательности

$$\Delta a\Delta b\Delta c\Delta d\Delta a/b/c\Delta a\times b\Delta a\times b\times c\Delta a\times b\times c\times d\Delta a\times b\times c\times d+a/b/c\Delta,\quad \Delta a\Delta b\Delta c\Delta d\Delta a/b/c\Delta a\times b\Delta c\times d\Delta a\times b\times c\times d\Delta a\times b\times c\times d+a/b/c\Delta.$$

В этих примерах каждый строгий процесс построения формулы оказывался ультрастрогим процессом ее построения. Существуют формулы, для которых некоторые строгие процессы их построения не являются ультрастрогими. Например, последовательность формул

$$\Delta\, a\, \Delta\, b\, \Delta\, a \times a\, \Delta\, a \times a \times b\, \Delta\, a \times b\, \Delta\, a \times a \times b + a \times b\, \Delta$$

является строгим, но не ультрастрогим процессом построения для формулы

$$\Delta\, a \times a \times b + a \times b\, \Delta$$

Всякий строгий процесс построения формулы является полустрогим. Если взять строгий процесс построения формулы $\mathbb{Ш}$ и повторить в нем какую-нибудь (не не последнюю) формулу несколько раз, то получится полустрогий процесс, но являющийся строгим. Иным примером полустрогого, но не строгого процесса построения формулы может служить последовательность

$$\Delta\, a\, \Delta\, a + a\, \Delta\, a + a + a\, \Delta\, a + a + a + a\, \Delta$$

где $\mathbb{Ш} = \Delta\, a + a + a + a\, \Delta$.

Предложение 2 и связанное с ним понятие процесса построения формулы позволяют дать рекурсивное определение формулы, эквивалентное определению I:

ОПРЕДЕЛЕНИЕ $I^{\bullet}$ (a) Последовательность $\Delta\Delta$ есть формула с главной связью ∞ .

(b) Последовательность $\Delta\, \alpha\, \Delta$, где $\alpha \in \Pi$, есть формула с главной связью ∞ .

(c) Если $\mathbb{Ш}$ есть формула, то последовательность элементарных символов $\Delta\, (\tilde{\mathbb{Ш}})\, \Delta$ есть формула с главной связью ∞ .

(d) Если σ – ассоциативная связь и $\mathbb{Ш}$, $\mathbb{Щ}$ – две формулы, главные связи которых не слабее σ , то последовательность элементарных символов $\Delta\, \tilde{\mathbb{Ш}}\, \sigma\, \tilde{\mathbb{Щ}}\, \Delta$ есть формула с главной связью σ.

(e) Если σ – неассоциативная связь и $\mathbb{Ш}_1$, $\mathbb{Ш}_2$, ... , $\mathbb{Ш}_s$, где $s \geqslant 2$, – произвольный набор формул, главные связи которых сильнее σ , то последовательность элементарных символов

$$\Delta\, \tilde{\mathbb{Ш}}_1\, \sigma\, \tilde{\mathbb{Ш}}_2\, \sigma \ \ldots\ \sigma\, \tilde{\mathbb{Ш}}_s\, \Delta$$

есть формула с главной связью σ.

Кроме того, предложение 2 (и определение $I^{\bullet}$) позволяет понять содержательную интерпретацию понятия главной связи. Если все связи, входящие в формулу $\mathbb{Ш}$, "имеющую смысл", суть знаки обычных операций арифметики, алгебры и анализа, т.е.

такие знаки, как $+, \int, \times, \partial, *, \ldots$, то главная связь формулы Ш совпадает со знаком "самой внешней операции" в формуле Ш. Аналогичное замечание справедливо и для тех формул, все связи которых суть знаки логических или теоретико-множественных операций и отношений. Поэтому главная связь часто отражается в названии формулы. Например, говорят: сумма $\Delta a/2 + a \times b \Delta$, произведение $\Delta (a \times b) \times f * t \Delta$, степень $\Delta a \uparrow (a+b) \Delta$, логарифм $\Delta a \log (a+b) \Delta$, интеграл $\Delta f * t \int t ; T \Delta$, дробь $\Delta (a+b)/(a \cdot b) \Delta$, производная $\Delta f * t \partial t \Delta$, функция $\Delta f * t \Delta$, пересечение $\Delta (a \cup b) \cap (c \setminus d) \Delta$, импликация $\Delta A \Rightarrow B \Delta$, тождество $\Delta a \Leftrightarrow b \Delta$, равенство $\Delta a = b = c \Delta$, включение $\Delta a \cap b \subset a \cup b \Delta$, отрицание $\Delta \neg a = b \Delta$, неравенства $\Delta a \neq b \Delta$, $\Delta a \leq b \Delta$ и т.д.

§ 4. ПОДФОРМУЛЫ

ОПРЕДЕЛЕНИЕ 5. Мы говорим, что формула $\Ш$ является подформулой формулы Ш или, точнее, входит в нее начиная с i-го места, если формула Ш представима в виде

$$Ш = \Delta \alpha_1 \alpha_2 \ldots \alpha_{i-1} \, \Ш \, \alpha_{j+1} \alpha_{j+2} \ldots \alpha_n \Delta , \qquad (10)$$

причем выполнены следующие условия:

(a) Символ α_{i-1} есть либо Δ, либо (, либо связь.

(b) Символ α_{j+1} есть либо Δ, либо), либо связь.

(c) Если главная связь формулы $\Ш$ ассоциативна, то она не слабее каждого из символов α_{i-1} и α_{j+1}.

(d) Если главная связь формулы $\Ш$ неассоциативна, то она сильнее каждого из символов α_{i-1} и α_{j+1}.

ПРИМЕРЫ. Подформулами формулы

$$\Delta a/a/a + b/b + a \times a \times a \Delta$$

кроме самой этой формулы являются формулы

$$\Delta a \Delta b \Delta a/a/a \Delta b/b \Delta a \times a \Delta a \times a \times a \Delta a/a/a + b/b \Delta b/b + a \cdot a \times a \Delta$$

и только они, причем формула $\Delta a \times a \Delta$ входит в данную формулу дважды (начиная с 11-го и с 13-го места), формула $\Delta b \Delta$ – также дважды (с 7-го и с 9-го места), а формула $\Delta a \Delta$ входит в рассматриваемую формулу 6 раз (при i = 1, 3, 5, 11, 13, 15). Формулы $\Delta a/a \Delta$, $\Delta a + b \Delta$, $\Delta b + a \times a \Delta$ и т.п. не

входят в данную формулу, хотя и являются ее подпоследовательностями.

В формуле $\Delta + \times ( ) + (++) \dashv \Delta$ кроме самой этой формулы имеются следующие подформулы

$\Delta\Delta + \times ( ) \Delta \times ( ) \Delta ( ) \Delta (++) \Delta (++) \dashv \Delta + \Delta ++ \Delta$,

причем подформула $\Delta + \Delta$ входит в данную формулу дважды (с 7-го и 8-го места), а пустая подформула $\Delta\Delta$ – 7 раз (при $i = 1, 2, 4, 7, 8, 9, 11$).

Вообще непустая формула Ш содержит в себе пустую подформулу $\Delta\Delta$ в том и только в том случае, когда выполнено хотя бы одно из условий: формула Ш начинается с некоторой связи; формула Ш оканчивается связью; в формуле Ш перед некоторой связью находится раскрывающая скобка; в формуле Ш после некоторой связи находится закрывающая скобка; в формуле Ш имеются две связи, расположенные рядом; формула Ш содержит подформулу $\Delta ( ) \Delta$.

Если $\alpha_i \in \Pi$, то формула $\Delta \alpha_i \Delta$ входит в формулу

$$Ш = \Delta \alpha_1 \alpha_2 \ldots \alpha_{i-1} \alpha_i \alpha_{i+1} \ldots \alpha_n \Delta$$

с i -го места. Напротив, если $\alpha_i \in \Sigma$, то формула $\Delta \alpha_i \Delta$ вообще говоря, в формулу Ш не входит; если же эта формула входит в Ш (с i -го места), то формула $\Delta\Delta$ входит в формулу Ш по меньшей мере дважды – с i -го и с $(i + 1)$-го места.

Отношение линейного порядка между формулами, введенное определением 2, индуцирует во множестве всех подформул данной формулы важное отношение линейного порядка, которое мы существенно используем в § 7 при определении операции коммутирования формулы. Для этого отношения порядка мы резервируем термины "сильнее", "слабее", "не сильнее", и "не слабее".

Кроме того, для определения операции подстановки (см. § 9) нам потребуется еще одно отношение линейного порядка между подформулами, определяемое при помощи анализа их взаимного расположения в исходной формуле. Для этого отношения порядка мы употребляем термины "предшествует", "следует", "не предшествует" и "не следует". Поскольку в данном случае речь идет о сравнении подформул по месту их записи в исходной формуле Ш, то нужно тщательно отличать друг от друга подформулы, представляющие собой одну и ту же формулу, но входящую в формулу Ш начиная с разных мест. В этом смысле формула $\Delta a + a \Delta$ совпадает с тремя различными подформулами формулы $\Delta a + a + a + a \Delta$,

входящими в нее начиная с I-го, с 3-го и с 5-го места соответственно. Точнее было бы говорить здесь о различных вхождениях или о множестве пар (i , Щ), где $i \geq 1$ и Щ – формула, входящая в формулу Ш начиная с i -го места. Однако мы предпочитаем применять в этой ситуации более простой термин "подформула", допуская тем самым известную вольность речи, и надеемся, что это не приведет к недоразумениям.

ОПРЕДЕЛЕНИЕ 6. Будем говорить, что подформула Щ, входящая в формулу Ш с i -го места, предшествует подформуле Э, входящей в формулу Ш с K -го места, или, что то же, подформула Э следует за подформулой Щ, если $i < K$, либо если $i = K$ и подформула Щ короче, т.е. содержит меньше элементарных символов, чем подформула Э.

Ясно, что это отношение порядка отлично от предыдущего. Однако, если две подформулы входят в данную формулу с одного и того же места, то для них оба отношения порядка очевидно совпадают.

ПРИМЕР. В формуле

$$\Delta\, b + a + a + a + b + a + a\, \Delta$$

подформула $\Delta a + a + a\, \Delta$ предшествует подформуле $\Delta\, a + a\, \Delta$, начинающейся с 5-го места, хотя и сильнее ее, но следует за подформулой $\Delta\, a + a\, \Delta$, начинающейся с 3-го места. Все эти подформулы следуют за подформулой $\Delta\, b + a\, \Delta$, начинающейся с I-го места, и предшествуют подформуле $\Delta\, b + a\, \Delta$, начинающейся с 9-го места.

Ниже нам потребуется еще одно понятие, связанное с подформулами.

ОПРЕДЕЛЕНИЕ 7. Мы говорим, что подформулы

$$Щ = \Delta\, \alpha_i\, \alpha_{i+1} \dots \alpha_j\, \Delta \qquad \text{и} \quad Э = \Delta\, \alpha_K\, \alpha_{K+1} \dots \alpha_m \Delta, \quad (II)$$

входящие в формулу

$$Ш = \Delta\, \alpha_1\, \alpha_2 \dots \alpha_n \Delta$$

с i -го и с K -го места соответственно, не соприкасаются, если $j + I < K$ или $m + I < i$.

ПРИМЕР. В формуле

$$\Delta\, a + b + + c + d + f + + g + h\, \Delta$$

не соприкасаются друг с другом подформулы $\Delta a + b\, \Delta$ и $\Delta c + d \Delta$, $\Delta b \Delta$ и $\Delta + c\, \Delta$, $\Delta\, a + b + \Delta$ и $\Delta\, c + d + f\, \Delta$, пустые

подформулы, начинающиеся с 5-го и с I2-го места, и т.д. С другой стороны, подформулы $\Delta\, a + b + \Delta$ и $\Delta\, b + + c\, \Delta$, $\Delta\, b + + c + d\, \Delta$ и $\Delta + c\, \Delta$, $\Delta\, b + \Delta$ и $\Delta + c\, \Delta$, $\Delta\, b + \Delta$ и пустая подформула, начинающаяся с 5-го места соприкасаются друг с другом.

Следующие свойства подформул очевидны:

Если подформулы (II) формулы Ш соприкасаются, то последовательности

$$\Delta\ \alpha_{max(i,k)}\ \alpha_{max(i,k)+1}\ \ldots\ \alpha_{min(j,m)}\ \Delta\ \alpha_{min(i,k)}$$

$$\alpha_{min(i,k)+1}\ \ldots\ \alpha_{max(j,m)}\ \Delta$$

суть подформулы формулы Ш, начинающиеся с i_1-го и i_2-го места соответственно, где $i_1 = max(i,k)$ и $i_2 = min(i,k)$ Эти подформулы естественно объявить пересечением и соответственно объединением соприкасающихся подформул (II).

Пересечение соприкасающихся подформул Щ и Э формулы Ш есть подформула каждой из формул Щ и Э. В свою очередь, формулы Щ и Э суть подформулы своего объединения.

Пересечение соприкасающихся подформул (II) формулы Ш, из которых первая предшествует второй, пусто тогда и только тогда, когда $j = k - I$. Если это условие выполнено и обе подформулы (II) не пусты, то символы α_j и $\alpha_k = \alpha_{j+1}$ совпадают и представляют собой ассоциативную связь, которая слукит главной связью обеих подформул (II).

Различные пустые подформулы формулы Ш не соприкасаются друг с другом.

Если соприкасающиеся подформулы (II) формулы Ш обе не совпадают со своим пересечением и первая из них предшествует второй, то символы α_{j+1} и α_{k-1} представляют собой одну и ту же ассоциативную связь, и эта связь является главной связью обеих подформул (II).

Если формула Щ входит в формулу Ш с i-го места, а формула Ш входит в формулу Ю с ℓ-го места, то Щ входит в формулу Ю начиная с $(i + \ell - 1)$-го места. Следовательно, бинарное отношение "Щ входит в Ш" есть отношение частичного порядка во множестве всех формул.

ПРЕДЛОЖЕНИЕ 3. Для того, чтобы формула Щ была подформулой формулы Ш, необходимо и достаточно, чтобы формула Щ участвова-

ла в некотором полустрогом процессе построения формулы Ш.

ДОКАЗАТЕЛЬСТВО. Достаточность обеспечивается транзитивностью отношения вхождения одной формулы в другую и следующими очевидными замечаниями. Формула $Ш_i$ есть подформула формулы $\Delta(\bar{Ш}_i)\Delta$. Если главные связи формул $Ш_i$ и $Ш_j$ не слабее связи $\sigma \in \Sigma_a$, то $Ш_i$ и $Ш_j$ суть подформулы формулы $\Delta \bar{Ш}_i \sigma \bar{Ш}_j \Delta$. Если $s \geqslant 2$ и главные связи формул (9) сильнее связи σ, то формулы (9) суть подформулы формулы (8).

Докажем необходимость индукцией по числу символов в формуле Ш. Для формулы $Ш = \Delta\Delta$ утверждение очевидно. Пусть оно верно для любой формулы, содержащей не более $(n-1)$-го неграничного элементарного символа.

Возьмем формулу Ш, содержащую n отличных от Δ символов, и допустим, что формула Щ входит в формулу Ш с i-го места, т.е. формула Ш имеет вид (10), причем удовлетворены условия (a), (b), (c) и (d) определения 5.

Предложение I указывает для структуры формулы Щ три возможности. В первом случае формула Ш имеет вид $Ш = \Delta \alpha \Delta$, где $\alpha \in \Pi$. В этом случае утверждение очевидно.

Во втором случае существует такая формула $Ш_1$, что $Ш = \Delta(\bar{Ш}_1)\Delta$. Если в этом случае $i = 1$ или $j = n$, то из определений I и 5 следует, что $Щ = Ш$, и утверждение очевидно. Если же $i > 1$ и $j < n$, то формула Щ входит не только в формулу Ш, но и в ее подформулу $Ш_1$. Согласно предположению индукции существует полустрогий процесс построения формулы Ш, содержащий формулу Щ. Добавляя к этому процессу в качестве последнего члена формулу Ш, мы получим искомый полустрогий процесс построения формулы Ш.

В третьем случае главная связь σ формулы Ш отлична от и формула Ш имеет вид

$$Ш = \Delta \bar{Щ}_1 \sigma \bar{Щ}_2 \sigma \ldots \sigma \bar{Щ}_s \Delta =$$

$$\Delta \alpha_{i_0+1} \alpha_{i_0+2} \ldots \alpha_{i_1-1} \sigma \alpha_{i_1+1} \alpha_{i_1+2} \ldots \alpha_{i_2-1} \sigma \ldots$$

$$\sigma \alpha_{i_{s-1}+1} \alpha_{i_{s-1}+2} \ldots \alpha_{i_s-1} \Delta,$$

где $s \geqslant 2$, $i_0 = 0$, $i_s = n+1$ и главные связи формул

$$Щ_k = \Delta \alpha_{i_{k-1}+1} \alpha_{i_{k-1}+2} \ldots \alpha_{i_k-1} \Delta, \quad k = 1, 2, \ldots, s \qquad (12)$$

сильнее связи σ. Ясно, что каждая формула $Щ_k$ содержит не бо-

лее (n – I) -го отличного от Δ символа и подформула $\mathfrak{Ш} = \Delta\,\alpha_i\,\alpha_{i+1}\,\ldots\,\alpha_j\,\Delta$ в формуле (8^0) соприкасается по крайней мере с одной подформулой вида (I2).

Допустим, что $\mathfrak{Ш}$ соприкасается лишь с одной подформулой $\mathfrak{Ш}_\ell$ вида (I2). Тогда $i_{\ell-1} + I \le i, j \le i_\ell - I$ и, следовательно, $\mathfrak{Ш}$ есть подформула формулы $\mathfrak{Ш}_\ell$. По предположению индукции существует полустрогий процесс $\mathfrak{Ц}_1, \mathfrak{Ц}_2, \ldots, \mathfrak{Ц}_{m_\ell}$ построения формулы $\mathfrak{Ш}_\ell$, в котором участвует $\mathfrak{Ш}$. Рассмотрим последовательность $\mathfrak{Ц}_1, \mathfrak{Ц}_2, \ldots, \mathfrak{Ц}_q$, содержащую

$$q = m_\ell + m_1 + m_2 + \ldots + m_s$$

формул, из которых первые m_ℓ формул образуют вышеупомянутый процесс построения формулы $\mathfrak{Ш}_\ell$, следующие m_1 формул образуют полустрогий процесс построения формулы $\mathfrak{Ш}_1$, следующие m_2 формул образуют полустрогий процесс построения формулы $\mathfrak{Ш}_2$ и т.д. Если $\sigma \notin \Sigma_a$, то в силу (8^0) последовательность $\mathfrak{Ц}_1, \mathfrak{Ц}_2, \ldots, \mathfrak{Ц}_q, \mathfrak{Ш}$ будет полустрогим процессом построения формулы $\mathfrak{Ш}$, содержащим в себе $\mathfrak{Ш}$. Если же $\sigma \in \Sigma_a$, то искомым полустрогим процессом построения формулы $\mathfrak{Ш}$ будет последовательность формул

$$\Delta\,\tilde{\mathfrak{Ц}}_1\,\Delta\,\tilde{\mathfrak{Ц}}_2\,\Delta\,\ldots\Delta\,\tilde{\mathfrak{Ц}}_q\,\Delta\,\tilde{\mathfrak{Ш}}_1\,\sigma\,\tilde{\mathfrak{Ш}}_2\,\Delta\,\ldots\,\Delta\,\tilde{\mathfrak{Ш}}_1\,\sigma$$
$$\tilde{\mathfrak{Ш}}_2\,\sigma\,\ldots\,\sigma\,\tilde{\mathfrak{Ш}}_s\,\Delta\,.$$

Допустим, что подформула $\mathfrak{Ш}$ соприкасается с несколькими подформулами вида (I2) в формуле (8^0). Обозначим через p и q номера первой и последней среди таких подформул. Тогда $p < q$ и

$$i_{p-1} + 1 \le i \le i_p \le i_{q-1} \le j_q - 1$$

Покажем, что в этом случае

$$\mathfrak{Ш} = \Delta\,\alpha_i\,\alpha_{i+1}\ldots\alpha_j\,\Delta = \Delta\,\tilde{\mathfrak{Ш}}_p\,\sigma\,\tilde{\mathfrak{Ш}}_{p+1}\,\sigma\ldots\sigma\,\tilde{\mathfrak{Ш}}_q\,\Delta\,, \qquad (I3)$$

т.е. $i = i_{p-1} + I$ и $j = i_q - I$. Предположим, что $i > i_{p-1} + 1$. Тогда отрезок

$$\alpha_i\,\alpha_{i+1}\,\ldots\,\alpha_{i_p-1}$$

(пустой в случае $i = i_p$) является началом формулы $\mathfrak{Ш}$ и концом формулы $\mathfrak{Ш}_p$ и не совпадает ни с $\mathfrak{Ш}$, ни с $\tilde{\mathfrak{Ш}}_p$, ибо не содержит членов α_{i_p} и α_{i-1}. Ясно также, что в этом отрезке раскрывающих скобок столько же, сколько и закрывающих. Поэтому

$$\alpha_{i-1} \in \Sigma(Щ_\rho)\ ,\ \sigma = \alpha_{i_\rho} \in \Sigma(Щ)$$

Но тогда для главных связей $\sigma_{Щ}$, σ , $\sigma_{Щ_\rho}$ формул Щ, Ш, $Щ_\rho$ соответственно имеем неравенства

$$\alpha_{i-1} \leq \sigma_{Щ} \leq \sigma < \sigma_{Щ_\rho} \leq \alpha_{i-1}$$

(первое верно, ибо Щ входит в Ш с i -го места, второе и четвертое – по определению главных связей, третье – благодаря выбору разложения (8^0) формулы Ш). Противоречие. Значит, $i = i_{\rho-1}+1$ Равенство $j = i_q - 1$ устанавливается аналогично.

Если $\rho = 1$ и $q = s$, то формулы (8^0) и (13) совпадают и в этом случае наше утверждение очевидно. Пусть $\rho > 1$ и $q < s$. В этом случае связь σ ассоциативна, поскольку иначе формула (13) не могла бы входить в формулу (8^0) начиная с i -го места. Рассмотрим произвольные полустрогие процессы

$$\Delta\, \tilde{Э}_1\, \Delta\, \tilde{Э}_2\, \Delta \ldots \Delta\, \tilde{Э}_r\, \Delta\ ,\ \Delta\, \tilde{Ш}_1\, \Delta\, \tilde{Ш}_2\, \Delta \ldots \Delta\, \tilde{Ш}_m\, \Delta\ ,$$

$$\Delta\, \tilde{Ю}_1\, \Delta\, \tilde{Ю}_2\, \Delta \ldots \Delta\, \tilde{Ю}_t\, \Delta$$

построения формул

$$Э = \Delta\, \tilde{Щ}_1\, \sigma \ldots \sigma\, \tilde{Щ}_{\rho-1}\, \Delta\ ,\ Щ = \Delta\, \tilde{Щ}_\rho\, \sigma \ldots \sigma\, \tilde{Щ}_q\, \Delta\ ,$$

$$Ю = \Delta\, \tilde{Щ}_{q+1}\, \sigma \ldots \sigma\, \tilde{Щ}_s\, \Delta$$

соответственно. Тогда последовательность формул

$$\Delta\, \tilde{Э}_1\, \Delta \ldots \Delta\, \tilde{Э}_r\, \Delta\, \tilde{Ш}_1\, \Delta \ldots \Delta\, \tilde{Ш}_m\, \Delta\, \tilde{Ю}_1\, \Delta \ldots \Delta\, \tilde{Ю}_t\, \Delta\, \tilde{Э}$$

$$\sigma\, \tilde{Щ}\ \Delta\ \tilde{Э}\, \sigma\, \tilde{Щ}\, \sigma\, \tilde{Ю}\, \Delta$$

будет искомым полустрогим процессом построения формулы Ш.

Случаи $\rho > 1$ и $q = s$, $\rho = 1$ и $q < s$ рассматриваются аналогично. Доказательство предложения 3 завершено.

ЗАМЕЧАНИЕ. Наперед заданный полустрогий процесс построения формулы Ш как правило не обязан содержать все ее подформулы. С другой стороны, можно привести пример формулы, некоторая подформула которой не содержится ни в одном строгом и тем более ультрастрогом процессе ее построения. Таковы, например, подфор-

мула $\Delta\, a + a + a + a + a\, \Delta$

в формуле $\Delta\, a + a + a + a + a + a\, \Delta$

или подформула $\Delta + + \Delta$ в формуле $\Delta + + + \Delta$.

§ 5. ПАРЫ СКОБОК

ОПРЕДЕЛЕНИЕ 8. Пусть в формуле

$$Ш = \Delta\, \alpha_1 \alpha_2 \dots \alpha_n\, \Delta$$

символ α_i есть раскрывающая скобка. Из определения формулы вытекает существование единственного индекса j , $i < j \leq n$, для которого α_j есть закрывающая скобка, т.е.

$$Ш = \Delta\, \alpha_1 \alpha_2 \dots \alpha_{i-1} (\alpha_{i+1} \alpha_{i+2} \dots \alpha_{j-1}) \alpha_{j+1} \alpha_{j+2} \dots \alpha_n\, \Delta , \qquad (I4)$$

и $\Delta\, \alpha_{i+1} \alpha_{i+2} \dots \alpha_{j-1}\, \Delta$ есть формула. Мы говорим, что эти скобки $\alpha_i = ($ и $\alpha_j =)$ соответствуют друг другу или, иначе, образуют в формуле Ш пару скобок $\{\alpha_i , \alpha_j\}$

ПРИМЕР. В формуле

$$\Delta\, \underset{12}{((}\, a + \underset{2}{b}\, \underset{1}{)} \times \underset{3}{c}\, \underset{3}{)} + \underset{4}{2} \times \underset{4}{(}\, a + \underset{56}{b}\, \underset{65}{)} \times (a + c) \times ((\, a\,)) \Delta$$

имеется 6 пар скобок; скобки, образующие пару скобок, помечены одним подстрочным индексом.

Следующие свойства скобок очевидны:

В любой формуле каждая скобка участвует в образовании единственной пары скобок.

Если $\{\alpha_i , \alpha_j\}$ – пара скобок в формуле (I4), то последовательность

$$Ш = \Delta\, \alpha_1 \alpha_2 \dots \alpha_{i-1} \alpha_{i+1} \alpha_{i+2} \dots \alpha_{j-1} \alpha_{j+1} \alpha_{j+2} \dots \alpha_n\, \Delta \qquad (I5)$$

является формулой.

Если $\{\alpha_i , \alpha_j\}$ и $\{\alpha_\kappa , \alpha_m\}$ – две разные пары скобок в формуле (I4), то либо $i < \kappa < m < j$, либо $i < j < \kappa < m$, либо $\kappa < m < i < j$, либо $\kappa < i < j < m$.

Если $\{\alpha_i , \alpha_j\}$ и $\{\alpha_\kappa , \alpha_m\}$ – две разные пары скобок в формуле (I4), то $\{\alpha_\kappa , \alpha_m\}$ есть пара скобок в формуле (I5).

Если $\{\alpha_i, \alpha_j\}$ – пара скобок в формуле (I4), то последовательности

$$\Delta\, \alpha_{i+1} \alpha_{i+2} \ldots \alpha_{j-1}\, \Delta \qquad \text{и} \qquad \Delta\, (\alpha_{i+1} \alpha_{i+2} \ldots \alpha_{j-1})\, \Delta$$

Суть подформулы формулы (I4), начинающиеся соответственно с $(i+1)$-го и с i-го места. Эти подформулы участвуют в любом процессе построения формулы (I4).

ОПРЕДЕЛЕНИЕ 9. Мы говорим, что пары скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_\kappa, \alpha_m\}$ в формуле (I4) эквивалентны, если формула (I5) совпадает с формулой

$$\Delta\, \alpha_1 \alpha_2 \ldots \alpha_{k-1}\, \alpha_{k+1} \ldots \alpha_{m-1}\, \alpha_{m+1}\, \alpha_{m+2} \ldots \alpha_n\, \Delta \qquad (I6)$$

Непосредственно ясно, что данное отношение во множестве всех пар скобок данной формулы рефлексивно, симметрично и транзитивно, т.е. действительно является отношением эквивалентности.

Если пары скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_\kappa, \alpha_m\}$, где $i<m$, в формуле (I4) эквивалентны, то, сравнивая формулы (I5) и (I6) почленно, замечаем, что

$$\alpha_i = \alpha_{i+1} = \alpha_{i+2} = \ldots = \alpha_\kappa \qquad \text{и} \qquad \alpha_m = \alpha_{m+1} = \alpha_{m+2} = \ldots = \alpha_j$$

Следовательно, $\kappa - i = j - m$ и формула (I4) имеет вид

$$\text{Ш} = \Delta\, \alpha_1 \alpha_2 \ldots \alpha_{i-1} \underbrace{((\ldots((}_{\kappa-i+1 \text{ скобка}} \alpha_{\kappa+1}\, \alpha_{\kappa+2} \ldots \alpha_{m-1} \underbrace{))\ldots))}_{j-m+1 \text{ скобка}} \alpha_{j+1}\, \alpha_{j+2} \ldots \alpha_n\, \Delta,$$

причем в ней

$$\{\alpha_i, \alpha_j\},\ \{\alpha_{i+1}, \alpha_{j-1}\},\ \ldots,\ \{\alpha_{\kappa-1}, \alpha_{m+1}\} \text{ и } \{\alpha_\kappa, \alpha_m\}$$

суть попарно эквивалентные пары скобок. Это семейство пар скобок будет исчерпывать некоторый класс эквивалентности во множестве пар скобок формулы Ш в том и только в том случае, когда $\{\alpha_{i-1}, \alpha_{j+1}\}$ и $\{\alpha_{\kappa+1}, \alpha_{m-1}\}$ не являются парами скобок.

ПРИМЕР. В формуле

$$\Delta((((((x)+)))\ x\ (((+)x))+((((\ ))))))\ \Delta \qquad (17)$$
$$\quad 123456\ \ 6\ \ 543\quad 789\ \ 9\ \ 87\quad 10\,11\,13\,12\,11\,10\,21$$

13 пар скобок. Они образуют 6 классов эквивалентности, соответствующих следующим множествам их индексов: $\{1,2\}$, $\{3,4,5\}$, $\{6\}$, $\{7,8\}$, $\{9\}$, $\{10,11,12,13\}$.

ОПРЕДЕЛЕНИЕ 10. Пара скобок $\{\alpha_i, \alpha_j\}$ в формуле (14) называется устранимой если формула

$$Э = \Delta\ \alpha_{i+1}\ \alpha_{i+2} \cdots \alpha_{j-1}\ \Delta$$

входит в формулу (15) начиная с i-го места, т.е. если главная связь σ формулы Э ассоциативна и не слабее символов α_{i-1} и α_{j+1}, либо неассоциативна и сильнее этих символов. Если это условие выполнено, то формула (15) называется $\{\alpha_i, \alpha_j\}$-упрощением формулы (14).

ПРИМЕРЫ. В формуле (17) устранимы все пары скобок кроме девятой. В формуле

$$\Delta((a\,t(\,t*(t;s)))\times a)+(((a)+q))\times(b+a)\,\Delta \qquad (18)$$

устранимы все пары скобок кроме той, которая окаймляет подформулу $\Delta\, b + a\, \Delta$. В формулах

$$\Delta(((\ )))\,\Delta\,a+(b+c)\,\Delta\,(a\times b)\times((b\times c)\times d)\Delta\,sin*(cos*t)\,\Delta\,(sin*cos)*t\,\Delta$$

устранимы все пары скобок. Напротив, в формулах

$$\Delta(a+b)\times c\,\Delta((a+b)\times c)+(a+b)\Delta\ (a/(b/c))/((a/b)/c)\Delta \qquad (19)$$
$$a\,t(a\,t\,a)\ \Delta$$

устранимых пар скобок нет. В формуле $\Delta((a+b))\times c\ \Delta$ обе пары скобок устранимы, но их нельзя устранить одновременно, ибо после устранения одной из них другая оказывается неустранимой.

ПРЕДЛОЖЕНИЕ 4. Если $\{\alpha_i, \alpha_j\}$ – устранимая пара скобок в формуле (14) и $i > 1$ или $j < n$, то формула (14) и ее $\{\alpha_i, \alpha_j\}$ – упрощение (15) имеют одну и ту же главную связь.

ДОКАЗАТЕЛЬСТВО. Предположим для определенности, что $i > 1$.

Если в отрезке $\alpha_1\alpha_2 \dots \alpha_{i-1}$ формулы (I4) раскрывающих скобок больше, чем закрывающих, то в формуле (I5) тем же свойством обладают отрезки

$$\alpha_1\alpha_2 \dots \alpha_{i-1}\ \alpha_{i+1}\alpha_{i+2} \dots \alpha_k\ ,\quad i < k < j$$

Поэтому в данном случае $\Sigma(Ш) = \Sigma(Щ)$. Это равенство обеспечивает совпадение главных связей формул Ш и Щ.

Допустим, что в отрезке $\alpha_1\alpha_2 \dots \alpha_{i-1}$ раскрывающих скобок столько, сколько закрывающих. Тогда символ α_{i-1} отличен от Δ и (. Значит, α_{i-1} есть связь и содержится в Σ (Ш). Кроме того ясно, что

$$\Sigma(Щ) = \Sigma(Ш) \cup \Sigma(Э)$$

где

$$Э = \Delta\ \alpha_{i+1}\alpha_{i+2} \dots \alpha_{j-1}\ \Delta$$

Всякая связь из множества Σ (Э) не слабее главной связи формулы Э. Главная связь формулы Э не слабее символа α_{i-1} . Связь $\alpha_{i-1} \in \Sigma$ (Ш) не слабее главной связи формулы Ш. Таким образом, минимальные элементы множеств Σ (Щ) и Σ (Ш) совпадают. Предложение доказано.

Если две пары скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_k, \alpha_m\}$ в формуле (I4) эквивалентны, то обе они устранимы. Примеры однако показывают, что пара скобок $\{\alpha_i, \alpha_j\}$ в $\{\alpha_k, \alpha_m\}$ – упрощении (I6) и пара скобок $\{\alpha_k, \alpha_m\}$ в $\{\alpha_i, \alpha_j\}$ – упрощении (I5) формулы (I4) могут оказаться неустранимыми. Тем не менее справедливо:

ПРЕДЛОЖЕНИЕ 5. Если устранимые пары скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_k, \alpha_m\}$ в формуле (I4) неэквивалентны, то

(α) пара $\{\alpha_k, \alpha_m\}$ устранима в $\{\alpha_i, \alpha_j\}$ – упрощении (I5) формулы (I4);

(β) пара $\{\alpha_i, \alpha_j\}$ устранима в $\{\alpha_k, \alpha_m\}$ – упрощении (I6) формулы (I4);

(γ) $\{\alpha_k, \alpha_m\}$ – упрощение $\{\alpha_i, \alpha_j\}$ – упрощения формулы (I4) совпадает с $\{\alpha_i, \alpha_j\}$ – упрощением $\{\alpha_k, \alpha_m\}$ – – упрощения формулы (I4).

ДОКАЗАТЕЛЬСТВО. Ясно, что можно ограничиться доказательством утверждения (α). Для этого рассмотрим различные варианты взаимного расположения пар скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_k, \alpha_m\}$ в формуле (I4).

СЛУЧАЙ I: $j \prec K$. В этом случае $\{\alpha_i, \alpha_j\}$ – упрощение формулы (I4) имеет вид

$$Ш = \Delta\, \alpha_1 \alpha_2 \ldots \alpha_{i-1}\, \alpha_{i+1}\, \alpha_{i+2} \ldots \alpha_{j-1}\, \alpha_{j+1}\, \alpha_{j+2} \ldots \alpha_{k-1}$$

$$(\alpha_{k+1}\, \alpha_{k+2} \ldots \alpha_{m-1})\, \alpha_{m+1}\, \alpha_{m+2} \ldots \alpha_n\, \Delta$$

Поскольку вслед за скобкой) скобка (в формуле (I4) стоять не может, то $j \prec K - I$. Устранимость пары скобок $\{\alpha_K, \alpha_m\}$ в формуле (I4) обуславливается взаимным расположением элементарных символов в отрезке

$$\alpha_{k-1}\, (\alpha_{k+1}\, \alpha_{k+2} \ldots \alpha_{m-1})\, \alpha_{m+1}$$

Но этот отрезок целиком входит и в формулу Ш. Значит, пара скобок $\{\alpha_K, \alpha_m\}$ в формуле Ш также устранима.

СЛУЧАЙ 2: $m \prec i$ – аналогичен предыдущему случаю.

СЛУЧАЙ 3: $i = K - I$ и $m \prec j - I$. В этом случае $\{\alpha_i, \alpha_j\}$ – – упрощение формулы (I4) имеет вид

$$Ш = \Delta\, \alpha_1 \alpha_2 \ldots \alpha_{i-1}\, (\alpha_{k+1}\, \alpha_{k+2} \ldots \alpha_{m-1})\, \alpha_{m+1}\, \alpha_{m+2} \ldots \alpha_{j-1}\, \alpha_{j+1}\, \alpha_{j+2} \ldots \alpha_n\, \Delta$$

Поскольку пары скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_K, \alpha_m\}$ в формуле (I4) устранимы, то последовательности

$$Э = \Delta\, \alpha_{k+1}\, \alpha_{k+2} \ldots \alpha_{m-1}\, \Delta, \quad Ю = \Delta\, \alpha_{i+1}\, \alpha_{i+2} \ldots \alpha_{j-1}\, \Delta =$$

$$= \Delta\, (\tilde{Э})\, \alpha_{m+1} \ldots \alpha_{j-1}\, \Delta$$

являются формулами и для их главных связей $\sigma_э$, $\sigma_ю$ и элементарных символов α_{i-1}, α_{m+1} справедливы соотношения

$$\alpha_{m+1} \in \Sigma(Ю), \quad) \neq \alpha_{i-1} \leq \sigma_ю \leq \alpha_{m+1} \leq \sigma_э,$$

причем $\alpha_{i-1} \neq \sigma_ю$, если $\sigma_ю \notin \Sigma_a$, и $\alpha_{m+1} \neq \sigma_э$, если $\sigma_э \notin \Sigma_a$. Следовательно,

$$) \neq \alpha_{i-1} \leq \sigma_э \quad \text{и} \quad (\neq \alpha_{m+1} \leq \sigma_э,$$

причем α_{i-1} и α_{m+1} не совпадают с $\sigma_э$, если $\sigma_э \notin \Sigma_a$. Этих соотношений достаточно для устранимости пары скобок $\{\alpha_K, \alpha_m\}$ в формуле Ш.

СЛУЧАЙ 4: $i \prec K - I$ и $m = j - I$ – аналогичен предыдущему случаю.

СЛУЧАЙ 5: $i \prec K - I$ и $m \prec j - I$ – аналогичен случаю I.

СЛУЧАЙ 6: $i = K - I$ и $m = j - I$ – невозможен по условию.

СЛУЧАЙ 7: $K \prec i$ и $j \prec m$. – I. Формула (I4) и ее

$\{\alpha_i, \alpha_j\}$ – упрощение Ш в данном случае имеют соответственно вид

$$Ш = \Delta\,\alpha_1\alpha_2\ldots\alpha_{k-1}(\alpha_{k+1}\alpha_{k+2}\ldots\alpha_{i-1}(\alpha_{i+1}\alpha_{i+2}\ldots\alpha_{j-1})\alpha_{j+1}$$

$$\alpha_{j+2}\ldots\alpha_{m-1})\alpha_{m+1}\alpha_{m+2}\ldots\alpha_n\Delta,\quad Щ = \Delta\,\alpha_1\alpha_2\ldots\alpha_{k-1}$$

$$(\alpha_{k+1}\alpha_{k+2}\ldots\alpha_{i-1}\alpha_{i+1}\alpha_{i+2}\ldots\alpha_{j-1}\alpha_{j+1}\alpha_{j+2}\ldots\alpha_{m-1})\alpha_{m+1}\alpha_{m+2}\ldots\alpha_n\Delta$$

Для доказательства утверждения (α) в данном случае достаточно показать, что главные связки формул

$$\Delta\,\alpha_{k+1}\alpha_{k+2}\ldots\alpha_{i-1}(\alpha_{i+1}\alpha_{i+2}\cdots\alpha_{j-1})\alpha_{j+1}\alpha_{j+2}\ldots\alpha_{m-1}\,\Delta\,\alpha_{k+1}$$

$$\alpha_{k+2}\ldots\alpha_{i-1}\alpha_{i+1}\alpha_{i+2}\ldots\alpha_{j-1}\alpha_{j+1}\alpha_{j+2}\cdots\alpha_{jm-1}\,\Delta$$

совпадают. Но это вытекает из предложения 4, поскольку последняя формула является $\{\alpha_i,\alpha_j\}$ – упрощением предыдущей формулы.

СЛУЧАЙ 8: $K < i - 1$ и $j < m$ – аналогичен предыдущему случаю.

СЛУЧАЙ 9: $K = i - 1$ и $j = m - 1$ – невозможен по условию.

Иных случаев взаимного расположения пар скобок $\{\alpha_i, \alpha_j\}$ и $\{\alpha_k, \alpha_m\}$ в формуле (I4) нет. Предложение 5 доказано.

ПРЕДЛОЖЕНИЕ 6. Если $\{\alpha_i, \alpha_j\}$ и $\{\alpha_k, \alpha_m\}$ – две пары скобок в формуле (I4), из которых первая устранима, а вторая – неустранима, то в $\{\alpha_i, \alpha_j\}$ – упрощении формулы (I4) пара скобок $\{\alpha_k, \alpha_m\}$ также неустранима.

Это предложение доказывается аналогично предыдущему.

§ 6. ПОЧТИ НОРМАЛЬНЫЕ ФОРМУЛЫ

ОПРЕДЕЛЕНИЕ II. Формула называется почти нормальной, если в ней нет устранимых пар скобок.

ПРИМЕРЫ. Среди конкретных формул предыдущего параграфа почти нормальны только формулы (I9). Дальнейшими примерами почти нормальных формул могут служить формулы

$$\Delta\,(a \Rightarrow b) \Rightarrow c\;\Delta\;a \Rightarrow (b \Rightarrow c)\;\Delta\;(a \Leftrightarrow c) \Rightarrow ((a \Leftrightarrow b) \Leftrightarrow$$

$$(b \Leftrightarrow c))\;\Delta\;(a \setminus b) \setminus c\;\Delta\;a \setminus (b \setminus c)\;\Delta$$

Пусть $\mathbb{Ш}_I$ – произвольная формула. Если эта формула не является почти нормальной, то удаляя из нее какую-нибудь устранимую пару скобок, мы получим формулу $\mathbb{Ш}_2$, в которой устранимых пар скобок на одну или на две меньше, чем в исходной формуле $\mathbb{Ш}_I$. Если формула $\mathbb{Ш}_2$ не является почти нормальной, то удаляя из нее какую-нибудь устранимую пару скобок, мы получим формулу $\mathbb{Ш}_3$, в которой устранимых пар скобок еще меньше. Продолжая этот процесс далее, мы на некотором шаге получим очевидно почти нормальную формулу $\mathbb{Ш}_m$.

ОПРЕДЕЛЕНИЕ I2. Описанная конечная последовательность формул $\mathbb{Ш}_I$, $\mathbb{Ш}_2$, ... , $\mathbb{Ш}_m$ называется процессом приведения формулы $\mathbb{Ш}_I$ к почти нормальному виду. Формула $\mathbb{Ш}_m$ называется почти нормальным видом формулы $\mathbb{Ш}_I$.

Для всякой формулы $\mathbb{Ш}$, имеющей по крайней мере две неэквивалентные устранимые пары скобок, можно написать несколько различных процессов приведения ее к почти нормальному виду. Однако из предложений 5 и 6 следует, что любая формула $\mathbb{Ш}$ обладает единственным почти нормальным видом.

ЗАМЕЧАНИЕ. Хотя мы и игнорируем понятие смысла формулы, однако надеемся, что почти нормальный вид формулы, имеющей смысл, также имеет смысл и притом тот же самый.

ПРИМЕРЫ. Для формулы (I8) процессом приведения ее к почти нормальному виду может служить последовательность из следующих семи формул

$$\mathbb{Ш}_I = \Delta((a+(f*(t;s)))\times a)+(((a)+g))\times(b+a)\Delta,$$

$$\mathbb{Ш}_2 = \Delta((a+(f*t;s))\times a)+(((a)+g))\times(b+a)\Delta,$$

$$\mathbb{Ш}_3 = \Delta((a+f*t;s)\times a)+(((a)+g))\times(b+a)\Delta,$$

$$\mathbb{Ш}_4 = \Delta(a+f*t;s\times a)+(((a)+g))\times(b+a)\Delta,$$

$$\mathbb{Ш}_5 = \Delta a+f*t;s\times a+(((a)+g))\times(b+a)\Delta,$$

$$\mathbb{Ш}_6 = \Delta a+f*t;s\times a+((a+g))\times(b+a)\Delta,$$

$$\mathbb{Ш}_7 = \Delta a+f*t;s\times a+(a+g)\times(b+a)\Delta$$

Формула $\mathbb{Ш}_7$ есть почти нормальный вид формулы (I8). Аналогично можно найти, что почти нормальным видом формулы (I7) является формула

$$\Delta(x+)\times(+)\times+\Delta$$

Следующие утверждения очевидны:

Почти нормальная формула совпадает со своим почти нормальным видом.

Если главная связь почти нормальной формулы Ш равна ∞, то Ш = $\Delta\,\alpha\,\Delta$, где $\alpha \in \Pi$.

Если главная связь формулы Ш отлична от ∞, то она совпадает с главной связью почти нормального вида формулы Ш.

Если формула Щ входит в почти нормальную формулу, то либо формула Щ сама почти нормальна, либо Щ = $\Delta\,(\widetilde{Э})\,\Delta$, где Э – некоторая непустая почти нормальная формула.

Если почти нормальный вид формулы Ш не пуст, то он сильнее (в смысле определения 2) формулы Ш.

Два разных процесса приведения произвольной формулы Ш к почти нормальному виду содержат в себе одинаковое число членов. Это число не больше m_1 и не меньше $m_1 - m_2$, где m_1 – количество устранимых пар скобок в формуле Ш, а m_2 – количество классов эквивалентности во множестве этих пар скобок.

§ 7. ИНВЕРСИИ

ОПРЕДЕЛЕНИЕ I3. Пусть при некотором j символ α_j в формуле

$$Ш = \Delta\,\alpha_1\,\alpha_2 \ldots \alpha_n\,\Delta \qquad (20)$$

является связью. Легко видеть, что существуют единственные индексы i и κ, $1 \leqslant i \leqslant j \leqslant \kappa \leqslant n$, такие, что

($а$) последовательности

$$Э = \Delta\,\alpha_i\,\alpha_{i+1} \ldots \alpha_{j-1}\,\Delta, \quad Ю = \Delta\,\alpha_{j+1}\,\alpha_{j+2} \ldots \alpha_\kappa\,\Delta \qquad (21)$$

суть формулы, входящие в Ш с i-го и с (j + I)-го места соответственно;

($б$) главные связи формул (2I) сильнее связи α_j;

($с$) символы α_{i-1} и $\alpha_{\kappa+1}$ в формуле Ш (ясно, что $\alpha_{i-1} \neq)$ и $\alpha_{\kappa+1} \neq ($) не сильнее связи α_j.

Формулы Э и Ю мы будем называть левым и соответственно правым аргументами связи α_j в формуле Ш.

ПРИМЕРЫ. В формуле

$$\Delta\,a + b + a \uparrow 2 \times (a + b)\,\Delta$$

левый и правый аргументы I-го и 2-го знаков +, знаков $\uparrow$, $\times$ и

третьего знака + суть соответственно формулы $\Delta a \Delta$ и $\Delta b \Delta$, $\Delta b \Delta$ и $\Delta a \uparrow 2 \times (a+b) \Delta$, $\Delta a \Delta$ и $\Delta 2 \Delta$, $\Delta a \uparrow 2 \Delta$ и $\Delta (a+b) \Delta$, $\Delta a \Delta$ и $\Delta b \Delta$. В формуле $\Delta + \times \uparrow \Delta$ левые аргументы связей +, x, $\uparrow$ пусты, а правые аргументы суть соответственно $\Delta \times \uparrow \Delta$, $\Delta \uparrow \Delta$ и $\Delta \Delta$.

Если α_j – связь в формуле (20) и формулы (2I) суть ее левый и правый аргументы, то последовательность $\Delta\ \widetilde{Э}\ \alpha_j\ \widetilde{Ю}\ \Delta$ является формулой с главной связью α_j. Эта формула входит в формулу (20) начиная с i-го места в том и только в том случае, когда связь α_j ассоциативна, либо когда эта связь неассоциативна и отлична от символов α_{i-1} и α_{i+1}. Если формула $\Delta\ \widetilde{Э}\ \alpha_j\ \widetilde{Ю}\ \Delta$ входит в формулу (20) начиная с i-го места, то $\Delta\ \widetilde{Ю}\ \alpha_j\ \widetilde{Э}\ \Delta$ также с i-го места входит в формулу

$$Ш = \Delta \alpha_1 \alpha_2 \dots \alpha_{i-1} \alpha_{j+1} \alpha_{j+2} \dots \alpha_k \alpha_j \alpha_i$$

$$\alpha_{i+1} \dots \alpha_{j-1} \alpha_{k+1} \alpha_{k+2} \dots \alpha_n \qquad (22)$$

ОПРЕДЕЛЕНИЕ I4. Пусть связь α_j в формуле (20) коммутативна и формулы (2I) суть ее аргументы. Если левый аргумент Э сильнее (в смысле определения 2) правого аргумента Ю, то мы говорим, что эти аргументы в формуле (20) образуют инверсию, и называем переход от формулы (20) к формуле (22) (в которой аргументы связи α_j инверсию уже не образуют) устранением данной инверсии или однократным коммутированием формулы (20) относительно связи α_j.

ПРИМЕРЫ. Подформулы $\Delta b \Delta$ и $\Delta a \Delta$ образуют инверсию в каждой из следующих формул

$$\Delta b \Leftrightarrow a \Delta b = a \Delta b \vee a \Delta b \wedge a \Delta b \cup a \Delta b \cap a$$

$$\Delta b + a \Delta b \times a \Delta b + a + c \Delta b \times a + c \Delta$$

Операция однократного коммутирования приводит эти формулы к виду

$$\Delta a \Leftrightarrow b \Delta a = b \Delta a \vee b \Delta a \wedge b \Delta a \cup b \Delta a \cap b \Delta$$

$$a + b \Delta a \times b \Delta a + b + c \Delta a \times b + c \Delta, \qquad (23)$$

где никаких инверсий нет. Инверсий нет также в формулах

$$\Delta b \uparrow a \Delta a \uparrow b \Delta b / a / b \Delta b / / b / b \Delta (b) \Leftrightarrow a \Delta ((a)) + ( ) + (+ a) \Delta$$

В формуле $\Delta C \times a + b \times c \Delta$ две инверсии. Одну из них образуют аргументы первого знака x, а другую – аргументы знака +. Коммутирование данной формулы относительно первого знака x сразу приводит к формуле $\Delta a \times c + b \times c \Delta$, в которой инверсий нет. Если же исходную формулу прокоммутировать относительно знака +, то в полученной формуле $\Delta b \times c + c \times a \Delta$ останется одна инверсия. Устраняя эту инверсию, мы приходим к формуле $\Delta b \times c + a \times c \Delta$, в которой появилась новая инверсия, образованная аргументами знака +. И лишь третье применение операции однократного коммутирования приводит к формуле $\Delta a \times c + b \times c \Delta$ без инверсий.

Аналогичную ситуацию мы встречаем в формуле

$$Ш = \Delta ((f + a) \times (c / d) + (c / a) / a) \times ((b / f) / c) \Delta$$

Здесь 4 коммутативных связи и аргументы каждой из них образуют инверсию. Все эти инверсии исчезают в результате коммутирования формулы Ш относительно первого знака +. Иной порядок устранения инверсий может потребовать до семи применений операции коммутирования.

Примеры несколько иного типа доставляют формулы

$$\Delta b + c + d + a \Delta b \Leftrightarrow c \Leftrightarrow d \Leftrightarrow a \Delta b \times c + b \times c \times d \times a \Delta$$

$$a \vee b \vee c \vee \Delta \; a \wedge b \wedge c \wedge (d) \Delta$$

В каждой из этих формул имеется единственная инверсия (ее образуют аргументы последней связи). Здесь также для полного освобождения формулы от инверсий требуется несколько применений операции однократного коммутирования, но в отличие от предыдущего это совершается единственным способом.

В формуле $\Delta b + d + a + c \Delta$ инверсия одна, но в результате ее устранения появляются две новые инверсии. Тем же свойством обладает формула $\Delta b + c \times a \times b \Delta$. В формуле

$$\Delta b \times c + b \times d = b \times c + b \times e \times a \times c \Delta \qquad (24)$$

также единственная инверсия, но ее устранение приводит к одновременному образованию четырех новых инверсий.

Следующие утверждения очевидны:

В формулах Ψ и $\Delta\ (\widetilde{\Psi})\ \Delta$ количество инверсий одно и то же.

Если в непустой формуле Ψ инверсий нет и ее главная связь сильнее связи $\sigma \in \Sigma_\kappa$, то в формуле $\Delta\ (\widetilde{\Psi})\ \sigma\ \widetilde{\Psi}\ \Delta$ инверсий нет, а в формуле $\Delta\ \widetilde{\Psi}\ \sigma\ (\widetilde{\Psi})\ \Delta$ одна инверсия есть. Она образована аргументами $\widetilde{\Psi}$ и $\Delta\ (\widetilde{\Psi})\ \Delta$ главной связи σ этой формулы.

Если в формулах $\Psi_1, \Psi_2, \ldots, \Psi_s$ инверсий нет, главные связи этих формул сильнее связи $\sigma \in \Sigma_\kappa$ и каждая из этих формул не слабее предыдущей, то в формуле

$$\Psi = \Delta\ \widetilde{\Psi_1}\ \sigma\ \widetilde{\Psi_2}\ \sigma\ \ldots\ \sigma\ \widetilde{\Psi_s}\ \Delta \qquad (7)$$

инверсий нет.

В последних двух утверждениях ограничения на главные связи формул Ψ и Ψ_κ ослабить нельзя. Так, формула $\Delta . + a\ \Delta$ инверсий не имеет, но в формулах

$$\Delta\ (+a) + +a\ \Delta + a + (+a)\Delta + a + +a\ \Delta$$

инверсии есть.

Если в формулах $\Psi_1, \Psi_2, \ldots, \Psi_s$ инверсий нет и их главные связи не слабее связи $\sigma \notin \Sigma_\kappa$, то в формуле (7) инверсий также нет.

Однократное коммутирование формулы не меняет ее главной связи.

Формула Ψ, полученная из формулы Ψ в результате применения операции однократного коммутирования, слабее исходной формулы Ψ.

Если Э и Ю – аргументы связи α_j в формуле Ψ и $\{\alpha_\ell, \alpha_m\}$-пара скобок в формуле Ψ, то либо $\{\alpha_\ell, \alpha_m\}$ есть пара скобок в одном из аргументов Э и Ю, либо формула Э α_j Ю является подпоследовательностью отрезка

$$\alpha_{\ell+1}\ \alpha_{\ell+2}\ \ldots\ \alpha_{m-1},$$

заключенного в скобках $\{\alpha_\ell, \alpha_m\}$.

Если $\{\alpha_\ell, \alpha_m\}$ – пара скобок в формуле Ψ и Ψ – результат коммутирования формулы Ψ, то $\{\alpha_\ell, \alpha_m\}$ есть пара скобок и в формуле Ψ. Эта пара скобок в формуле Ψ устранима тогда и только тогда, когда она устранима в исходной формуле Ψ.

Результат однократного коммутирования почти нормальной формулы является почти нормальной формулой.

Устранение устранимой пары скобок может привести к исчезновению одних и к появлению других инверсий. Примером подобной ситуации может служить формула $\Delta\ a + (b) \times a\ \Delta$

§ 8. НОРМАЛЬНЫЕ ФОРМУЛЫ

ОПРЕДЕЛЕНИЕ I5. Формула Ш называется нормальной, если она почти нормальна и не имеет инверсий.

ПРИМЕРЫ. Пустая формула нормальна. Формула $\Delta \alpha \Delta$, где α – буква или связь, нормальна. Формулы (I9) и (23) нормальны. Формулы

$$\Delta + \times (+) \times (+ \times) \Delta\ (a + b) \times (a + g) + a \times a\ 1\ f * t_{;}\ s\ \Delta$$

$$a \times b \times c \times e + b \times c = b \times c + b \times d\ \Delta \qquad (25)$$

также нормальны. Формулы (I7), (I8), (24) и большая часть других конкретных формул в последних трех параграфах не являются нормальными формулами. Формулы

$$\Delta\ b + a\ \Delta\ a + c + d\ \Delta\ \sin * a + \Delta * / + \Delta\ (a * b) + a \times b\ \Delta$$

также ненормальны.

ОПРЕДЕЛЕНИЕ I6. Процессом нормализации формулы $Ш_I$ или, иначе, процессом приведения этой формулы к нормальному виду называется последовательность формул $Ш_I$, $Ш_2$, ..., в которой каждая формула $Ш_{i+1}$ является результатом устранения из формулы $Ш_i$ какой-нибудь инверсии или устранимой пары скобок, и в которой последняя формула (если она есть) нормальна. Эта нормальная формула называется нормальным видом формулы $Ш_I$.

ПРЕДЛОЖЕНИЕ 7. Всякая формула $Ш_I$ обладает единственным нормальным видом. Этот нормальный вид формулы $Ш_I$ является последним членом любого процесса нормализации формулы $Ш_I$.

ДОКАЗАТЕЛЬСТВО данного предложения проведем индукцией по числу членов в формуле $Ш_I$. В пределах этого доказательства под упрощением формулы условимся понимать результат устранения из нее какой-нибудь инверсии или устранимой пары скобок.

Утверждение очевидно для нормальной формулы. В частности,

оно очевидно для пустой формулы и для всякой формулы вида $\Delta\alpha\Delta$, где $\alpha \in \Sigma \cup \Pi$

Зафиксируем произвольное натуральное число $n \geq 2$ и допустим, что наше утверждение верно для всякой формулы, содержащей не более $(n-1)$-го отличного от Δ элементарного символа. Возьмем произвольную формулу

$$Ш_I = \Delta\,\alpha_1\,\alpha_2 \ldots \alpha_n\,\Delta$$

Предложение I для структуры формулы $Ш_I$ предоставляет только две возможности (ибо $n \geq 2$):

СЛУЧАЙ I. Формула $Ш_I$ имеет вид $Ш_I = \Delta\,(\widetilde{Ш}_I)\,\Delta$, где

$$Ш_I = \Delta\,\alpha_2\,\alpha_3 \ldots \alpha_{n-1}\,\Delta$$

есть формула. Согласно предположению индукции все процессы нормализации формулы $Ш_I$ конечны и их последние члены совпадают с одной и той же нормальной формулой.

Пусть $Ш_I, Ш_2, \ldots$ – произвольный процесс нормализации формулы $Ш_I$. В формуле $Ш_I = \Delta\,(\widetilde{Ш}_I)\,\Delta$ пара скобок $\{\alpha_1, \alpha_n\}$ устранима. Все остальные устранимые пары скобок и все инверсии формулы $Ш_I$ суть устранимые пары скобок и инверсий формулы $Ш_I$ (и наоборот). Следовательно, либо $Ш_2 = Ш_I$, либо $Ш_2 = \Delta\,(\widetilde{Ш}_2)\,\Delta$, где $Ш_2$ есть упрощение формулы $Ш_I$. Аналогично, либо $Ш_3 = Ш_2$, либо $Ш_3 = \Delta\,(\widetilde{Ш}_3)\,\Delta$, где $Ш_3$ есть упрощение формулы $Ш_2$. Продолжая эти рассуждения, получаем следующую альтернативу: Либо существует такой индекс k, что формула $Ш_{k+1}$ есть $\{\alpha_1, \alpha_n\}$ – упрощение формулы $Ш_k$, либо такого индекса не существует. В последнем случае процесс $Ш_I, Ш_2, \ldots$ не содержит нормальных формул и следовательно бесконечен. Но тогда $Ш_i = \Delta\,(\widetilde{Ш}_i)\,\Delta$ для каждого $i = 1, 2, \ldots$, и последовательность формул $Ш_I, Ш_2, \ldots$ оказывается бесконечным процессом нормализации формулы $Ш_I$ вопреки предположению индукции. Значит, индекс k с указанным выше свойством существует всегда и наш процесс нормализации формулы $Ш_I$ имеет вид

$$Ш_1 = \Delta\,(\widetilde{Ш}_1)\,\Delta,\ Ш_2 = \Delta\,(\widetilde{Ш}_2)\,\Delta, \ldots,$$

$$Ш_k = \Delta\,(\widetilde{Ш}_k)\,\Delta, Ш_{k+1} = Ш_k,\ Ш_{k+2} = Ш_{k+1}, \ldots$$

Последовательность $Ш_I, Ш_2, \ldots$ и в этом случае является процессом нормализации формулы $Ш_I$. По предположению индукции этот про-

цесс конечен и его последний член $Ш_m$ совпадает с единственным нормальным видом формулы $Ш_I$. Ясно, что $m \geqslant к$. Но тогда формула $Ш_{m+1} = Ш_m$ будет последним членом процесса $Ш_I$, $Ш_2$, ..., однозначно определенным выбором исходной формулы $Ш_I$. В рассматриваемом случае предложение доказано.

СЛУЧАЙ 2. Главная связь σ формулы $Ш_I$ отлична от ∞ и существуют формулы $Ш_I$, $Ш_2$, ... , $Ш_s$, $s \geqslant 2$, главные связи которых сильнее σ , такие, что формула $Ш_I$ имеет вид

$$Ш_1 = \Delta \widetilde{Ш}_1 \sigma \widetilde{Ш}_2 \sigma \ldots \sigma \widetilde{Ш}_s \Delta . \qquad (26)$$

Каждая формула $Ш_j$, $j = 1,2, \ldots , s$, содержит не более $(n - 1)$ -го отличного от Δ элементарного символа. Значит, согласно предположению индукции все процессы нормализации формулы $Ш_j$ конечны и оканчиваются одной и той же нормальной формулой $Ю_j$ – единственным нормальным видом формулы $Ш_j$.

Подформулы $Ш_j$, $j = 1,2, \ldots , s$, формулы $Ш_I$ будем называть главными. Ясно, что каждая устранимая пара скобок формулы $Ш_I$ является устранимой парой скобок в некоторой главной подформуле формулы $Ш_I$, и наоборот, каждая устранимая пара скобок любой главной подформулы является таковой и в формуле $Ш_I$. Что касается инверсий, то некоторые из них могут быть образованными соседними главными подформулами (при наличии таких инверсий связь σ коммутативна). Все эти инверсии в формуле $Ш_I$ мы будем именовать главными инверсиями. Любая иная инверсия в формуле $Ш_I$ является инверсией в некоторой главной подформуле и наоборот, любая инверсия любой главной подформулы является одновременно неглавной инверсией в формуле $Ш_I$.

Далее нам потребуется лемма:

ЛЕММА. Для любой формулы Ш вида (26) конечна любая последовательность формул Ш, Ш′ , Ш″ , ... , в которой каждая следующая формула получена из предыдущей формулы путем устранения в ней некоторой главной инверсии.

Эта лемма очевидна.

Продолжим доказательство предложения 7. Пусть $Ш_I$, $Ш_2$, ... – произвольный процесс нормализации формулы $Ш_I$, имеющей по предположению вид (26). Из леммы следует существование последовательности индексов $I < к_1 < к_2 < \ldots$, такой, что при $i \notin \{1, к_1, к_2, \ldots\}$ формула $Ш_i$ есть результат устранения из формулы

$\mathbb{Ш}_{i-1}$ некоторой главной инверсии, а при $i \in \{K_1, K_2, \dots\}$ формула $\mathbb{Ш}_i$ получена путем устранения некоторой инверсии или устранимой пары скобок в некоторой главной подформуле предыдущей формулы $\mathbb{Ш}_{i-1}$. Поскольку формула (26) имеет конечное число главных подформул $\mathbb{Ш}_j$ и поскольку каждая из этих подформул обладает (согласно предположению индукции) конечным процессом нормализации, то последовательность $K_1 < K_2 < \dots$ не может быть бесконечной. Пусть K_ρ – ее последний член. (Если все главные подформулы формулы (26) нормальны, то последовательность индексов $K_1 < K_2 < \dots$ пустая; в этом случае для дальнейшего можно условно положить $K_\rho = 1$). Формула $\mathbb{Ш}_{K_\rho}$ имеет вид

$$\mathbb{Ш}_{K_\rho} = \Delta\, \widetilde{\mathbb{Ш}}'_{j_1} \sigma\, \widetilde{\mathbb{Ш}}'_{j_2} \sigma \dots \sigma\, \widetilde{\mathbb{Ш}}'_{j_s}\, \Delta,$$

где $\{j_1, j_2, \dots, j_s\}$ есть некоторая перестановка индексов $\{1, 2, \dots, n\}$ и для каждого $j = 1, 2, \dots, s$ формула $\mathbb{Ш}'_j$ получена из формулы $\mathbb{Ш}_j$ в результате последовательного устранения некоторого количества инверсий и устранимых пар скобок. При $i > K_\rho$ формула $\mathbb{Ш}_i$ согласно определению индекса K_ρ получена в результате устранения некоторой главной инверсии из формулы $\mathbb{Ш}_{i-1}$. По лемме последовательность формул $\mathbb{Ш}_{K_\rho}$, $\mathbb{Ш}_{K_\rho+1}$, $\mathbb{Ш}_{K_\rho+2}, \dots$ конечна. Значит, весь процесс $\mathbb{Ш}_1, \mathbb{Ш}_2, \dots$ конечен. Пусть $\mathbb{Ш}_m$ – его последняя формула (ясно, что $m \geq K_\rho$). Формула $\mathbb{Ш}_m$ нормальна и совпадает с формулой $\mathbb{Ш}_{K_\rho}$ (при $m = K_\rho$), либо получена из нее в результате некоторой перестановки ее главных подформул, т.е. имеет вид

$$\mathbb{Ш}_m = \Delta\, \widetilde{\mathbb{Ш}}'_{\ell_1} \sigma\, \widetilde{\mathbb{Ш}}'_{\ell_2} \sigma \dots \sigma\, \widetilde{\mathbb{Ш}}'_{\ell_s}\, \Delta,$$

где $\{\ell_1, \ell_2, \dots, \ell_s\}$ есть некоторая перестановка индексов $\{1, 2, \dots, n\}$. Но тогда формулы $\mathbb{Ш}'_1, \mathbb{Ш}'_2, \dots, \mathbb{Ш}'_s$ также нормальны и совпадают стало быть с нормальными видами $\mathbb{Ю}_1, \mathbb{Ю}_2, \dots, \mathbb{Ю}_s$ формул $\mathbb{Ш}_1, \mathbb{Ш}_2, \dots, \mathbb{Ш}_s$ соответственно. Из единственности нормальных видов $\mathbb{Ю}_j$ формул $\mathbb{Ш}_j$ и из того факта, что формулы $\mathbb{Ю}_1, \mathbb{Ю}_2, \dots, \mathbb{Ю}_s$ лишь единственным образом могут быть расположены в виде монотонно возрастающей последовательности (в смысле отношения линейного порядка, описанного в определении 2), вытекает единственность нормального вида $\mathbb{Ш}_m$ формулы $\mathbb{Ш}_1$.

Предложение 7 доказано.

ПРИМЕРЫ. Для формулы

$$\Delta\, a\times((a+(b)))\,\Delta$$

примерами процессов ее нормализации могут служить следующие две последовательности формул

$$\Delta\, a\times((a+(b)))\,\Delta\ ((a+(b)))\times a\,\Delta\ (((b)+a))\times a\,\Delta$$
$$((b+a))\times a\,\Delta\ (b+a)\times a\,\Delta\ (a+b)\times a\,\Delta,$$
$$\Delta\, a\times((a+(b)))\,\Delta\, a\times(a+(b))\,\Delta\, a\times(a+b)\,\Delta\ (a+b)\times a\,\Delta$$

В обоих случаях последняя формула $\Delta\,(a+b)\times a\,\Delta$ одна и та же. Это есть нормальный вид исходной формулы.

Подобным же образом можно найти, что три формулы (25) суть нормальные виды формул (I7), (I8) и (24) соответственно.

Определение I7. Мы называем две формулы эквивалентными, если они имеют один и тот же нормальный вид.

ПРИМЕРЫ. Формулы

$$\Delta\,((a)+(b))\times c\,\Delta\, c\times(a+((b)))\,\Delta\, c\times(a+b)\,\Delta$$

попарно эквивалентны друг другу. Напротив, формулы

$$\Delta\, c\times(a+b)\,\Delta\, c\times a+c\times b\,\Delta\, c\times a+b\,\Delta+\Delta\, c\times(a+b)\uparrow 1\,\Delta$$

попарно неэквивалентны.

Ясно, что отношение эквивалентности, заданное во множестве всех формул определением I7, рефлексивно, симметрично и транзитивно, т.е. действительно является отношением эквивалентности. Это отношение разбивает множество всех формул на попарно непересекающиеся непустые классы эквивалентности; две формулы входят в один класс тогда и только тогда, когда они эквивалентны. В каждом классе эквивалентности содержится единственная нормальная формула – она является общим нормальным видом для всех формул данного класса.

В соответствии со смысловой нагрузкой понятия коммутативной связи мы считаем, что устранение инверсии в формуле "со смыслом" не меняет смысла этой формулы. Поэтому мы можем утверждать, что все осмысленные формулы данного класса эквивалентности имеют один и тот же смысл. В частности, этот же смысл имеет

нормальная формула рассматриваемого класса. Отметим еще, что поскольку не исключены "бессмысленные" нормальные формулы, то в некоторых классах эквивалентности нет ни одной осмысленной формулы.

Аналогично предложению 7 можно доказать

ПРЕДЛОЖЕНИЕ 8. Для любой формулы $Ш_I$ последовательность формул $Ш_I$, $Ш_2$, ... , каждая из которых получена в результате применения операции однократного коммутирования к предыдущей формуле, конечна и ее последний член, не содержащий инверсий, однозначно определяется исходной формулой $Ш_I$.

ОПРЕДЕЛЕНИЕ 18. Последовательность формул $Ш_I$, $Ш_2$, ... , $Ш_m$, о которой идет речь в предложении 8, называется процессом коммутирования формулы $Ш_I$. Последняя формула $Ш_m$ процесса коммутирования формулы $Ш_I$ называется результатом коммутирования формулы $Ш_I$.

Подобно процессам приведения формулы к нормальному или к почти нормальному виду, процесс ее коммутирования определен вообще говоря неоднозначно. Однако результат коммутирования данной формулы всегда один и тот же.

Если исходная формула Ш не является почти нормальной, то результат ее коммутирования также не будет почти нормальной формулой. Если же формула Ш почти нормальна, то таков же и результат ее коммутирования. Более того этот результат будет нормальной формулой, так как инверсии он содержать не может.

Таким образом, если для произвольной формулы Ш записать процесс ее приведения к почти нормальному виду и приписать к этому процессу справа процесс коммутирования (без первого члена) ее почти нормального вида, то полученная последовательность формул будет процессом приведения формулы Ш к нормальному виду.

Примером процесса коммутирования может служить последовательность

$$\Delta в + а \times (в \times а) \Delta\ а \times (в \times а) + в \Delta\ (в \times а) \times а + в \Delta\ (а \times в) \times а + в \Delta$$

В заключение данного параграфа отметим, что несколько изменяя доказательство предложения 7, можно дополнительно показать, что для каждой формулы Ш существует натуральное число

$n = n$ (Ш) такое, что всякий процесс нормализации (и тем более, всякий процесс коммутирования) формулы Ш содержит не более, чем n членов.

§ 9. ПРОСТАЯ ПОДСТАНОВКА

В данном параграфе мы рассмотрим несколько модификаций операции простой подстановки – одной из основных операций Авто-Аналитика. Интуитивное содержание этой операции заключается в том, что в данной формуле подформулы указанного вида заменяются другими заданными подформулами. Так результатом простой подстановки формулы $\Delta\, W \Delta$ в формулу $\Delta\, a + b + c + d\, \Delta$ вместо ее подформулы $\Delta\, b + c\, \Delta$ должна быть формула $\Delta\, a + W + d\, \Delta$. Подставляя формулу $\Delta\, W \Delta$ в формулу

$$Ш = \Delta\, a + b \times c \times (a + b) + d\ \Delta$$

вместо ее подформулы $\Delta\, a + b\, \Delta$, мы ожидаем получить формулу

$$\Delta\, a + b \times c \times W + d\ \Delta .$$

Подставляя $\Delta\, a + b\, \Delta$ вместо $\Delta\, b \times c\, \Delta$ в ту же формулу Ш, мы должны получить

$$\Delta\ a + (a + b) \times (a + b) + d\ \Delta$$

Для некоторого упрощения стилистических оборотов, связанных с определением и использованием понятия простой подстановки, введем вспомогательное понятие правила:

ОПРЕДЕЛЕНИЕ I9. Правилом называется любая упорядоченная пара $\{Э, Ю\}$, состоящая из двух формул Э и Ю. Формула Э называется левой частью данного правила, а формула Ю – его правой частью.

Рассмотрим еще один предварительный пример. Пусть исходная формула Ш и правило ее преобразования $\{Э, Ю\}$ суть соответственно

$$\Delta\, a + b + a + a + a + b + a + a + a + b + a\ \Delta \qquad \text{и}$$

$$\{\Delta\ a + b + a\, \Delta, \Delta\, b + a + b\, \Delta\}$$

В этом случае в зависимости от нужд задачи математика результатом простой подстановки формулы Ю вместо Э, т.е. результатом преобразования формулы Ш по правилу $\{Э, Ю\}$, можно объявить

любую из следующих четырех формул:

$$Ш_1 = \Delta\, б + а + б + а + а + б + а + а + а + б + а\, \Delta,$$
$$Ш_2 = \Delta\, б + а + б + а + б + а + б + а + б + а + б\, \Delta,$$
$$Ш_3 = \Delta\, б + б + б + а + б + б + б + а + б + б + б\, \Delta,$$
$$Ш_4 = \Delta\, б + б + а + б + б + б + а + б + б + а + б\, \Delta.$$

Здесь формула $Ш_1$ получена путем замены формулой $\Delta\, б + а + б\, \Delta$ лишь одной (а именно той, которая начинается с I-го места) подформулы вида $\Delta\, а + б + а\, \Delta$ исходной формулы Ш. Мы будем говорить, что формула $Ш_1$ есть результат однократной простой подстановки. Формула $Ш_2$ получена путем одновременной замены формулой $\Delta\, б + а + б\, \Delta$ наибольшего количества взаимно не соприкасающихся подформул вида $\Delta\, а + б + а\, \Delta$ формулы Ш (в данном случае заменены три подформулы, начинающиеся в формуле Ш соответственно с I-го, с 9-го и с I7-го места). Мы будем говорить, что формула $Ш_2$ есть результат строчной простой подстановки. Последние две формулы $Ш_3$ и $Ш_4$ получены путем повторения однократной и соответственно строчной простых подстановок.

Перейдем к строгим определениям.

ОПРЕДЕЛЕНИЕ 20. Пусть заданы формула Ш и правило $\{Э, Ю\}$. Если левая часть Э данного правила не входит в формулу Ш, то мы говорим, что простая подстановка в формулу Ш по правилу $\{Э, Ю\}$ невозможна; результатом простой подстановки в этом случае объявляется почти нормальный вид исходной формулы Ш.

Если же Э входит в Ш, то формула

$$Ш = \Delta\, \alpha_1\, \alpha_2 \ldots \alpha_n\, \Delta$$

единственным образом представима в виде

$$Ш = \Delta\, \alpha_1 \ldots \alpha_{i_1 - 1}\, \widetilde{Э}\, \alpha_{j_1} \ldots \alpha_{i_2 - 1}\, \widetilde{Э}\, \alpha_{j_2} \ldots \ldots \ldots \alpha_{i_\ell - 1}$$
$$\widetilde{Э}\, \alpha_{j_\ell} \ldots \ldots \ldots \alpha_{i_s - 1}\, \widetilde{Э}\, \alpha_{j_s} \ldots \alpha_n\, \Delta,$$

где индексы $i_1, i_2, \ldots, i_s$ таковы, что выполнены условия:

(а) Самая первая (в смысле определения 6) подформула Э формулы Ш входит в Ш начиная с i-го места.

(в) При $2 \le \ell \le s$ в формуле Ш есть подформула Э, которая

следует за подформулой Э, начинающейся с $i_{\ell-1}$-го места, но не соприкасается с ней. Самая первая такая подформула Э начинается в формуле Ш с i_ℓ-го места.

(C) В формуле Ш нет подформулы Э, которая следовала бы за подформулой Э, начинающейся с i_1-го места, и не соприкасалась бы с ней.

В этом случае мы говорим, что простая подстановка возможна. Результатом однократной и соответственно строчной простой подстановки в формулу Ш по правилу {Э, Ю} называется почти нормальный вид формулы

$$Ш_1 = \Delta \alpha_1 \dots \alpha_{i_1-1} (\tilde{Ю}) \alpha_{j_1} \dots \alpha_{i_2-1} \tilde{Э} \alpha_{j_2} \dots \dots \alpha_{i_3-1} (\tilde{Э}) \alpha_{j_3} \dots \alpha_n \Delta$$

и соответственно почти нормальный вид формулы

$$Ш_2 = \Delta \alpha_1 \dots \alpha_{i_1-1} (\tilde{Ю}) \alpha_{j_1} \dots \alpha_{i_2-1} (\tilde{Ю}) \alpha_{j_2} \dots \dots \alpha_{i_3-1} (\tilde{Ю}) \alpha_{j_3} \dots \alpha_n \Delta.$$

ОПРЕДЕЛЕНИЕ 21. Для любой формулы Ш и любого правила {Э,Ю} однозначно определена последовательность формул Ш, Ш′, Ш″, ... , Ш$^{(m)}$, ..., каждая из которых (кроме первой) есть результат однократной (соответственно строчной) простой подстановки по правилу {Э, Ю} в предыдущую формулу. Если в этой последовательности есть формула, подстановка в которую по правилу {Э, Ю} невозможна, то почти нормальный вид первой такой формулы объявляется результатом полной (соответственно циклической) простой подстановки в формулу Ш по данному правилу. Если же такой формулы в последовательности Ш, Ш′, Ш″, ... , Ш$^{(m)}$, ... нет, то мы говорим, что полная (соответственно циклическая) простая подстановка зацикливает; ее результат в этом случае неопределен.

Ниже приведены несколько примеров простых подстановок. В этих примерах исходная формула обозначается через Ш, левая и правая части правила – через Э и Ю, результаты однократной, строчной, полной и циклической простых подстановок в формулу Ш по правилу {Э, Ю}- через $Ш_1$, $Ш_2$, $Ш_3$ и $Ш_4$ соответственно.

ПРИМЕР I. Если

$$Э = \Delta (\alpha+2)+2 \Delta \quad , \quad Ю = \Delta \alpha+2+2\times\alpha\times\beta+\beta+2 \Delta$$

$$Ш = \Delta (\alpha+\beta)+2\times(\alpha+\beta)+2+(\alpha+\beta)+2 \Delta,$$

то

$$Ш_I = \Delta\,(a\uparrow 2 + 2\times a\times b + b\uparrow 2)\times(a+b)\uparrow 2 + (a+b)\uparrow 2\,\Delta,$$

$$Ш_2 = Ш_3 = Ш_4 = \Delta\,(a\uparrow 2 + 2\times a\times b + b\uparrow 2)\times(a\uparrow 2 + 2\times a\times b + b\uparrow 2) + a\uparrow 2 + 2\times a\times b + b\uparrow 2\,\Delta$$

ПРИМЕР 2. Если $Э = \Delta\,a\times b\,\Delta$, $Ю = \Delta\,b\uparrow 2\,\Delta$ и $Ш = \Delta\,a\times b\times a\times b\times c\times(a\times b)\,\log(a\times b) + a\times b\,\Delta$,

то

$$Ш_I = \Delta\,b\uparrow 2\times a\times b\times c\times(a\times b)\,\log(a\times b) + a\times b\,\Delta;$$

$$Ш_2 = Ш_3 = Ш_4 = \Delta\,b\uparrow 2\times b\uparrow 2\times c\times b\uparrow 2\,\log b\uparrow 2 + b\uparrow 2\,\Delta.$$

ПРИМЕР 3. Если

$Э = \Delta\,a\cap(b\cup c)$, $Ю = \Delta\,a\cap b\cup a\cap c\,\Delta$

и

$Ш = \Delta\,(b\setminus a\cap(b\cup c))\cup(c\cup a\cap(b\cup c))\cup(a\cap a\cap(b\cup c))\,\Delta$,

то

$$Ш_I = \Delta\,(b\setminus a\cap b\cup a\cap c)\cup c\cup a\cap(b\cup c)\cup a\cap a\cap(b\cup c)\,\Delta;$$

$$Ш_2 = Ш_3 = Ш_4 = \Delta\,(b\setminus a\cap b\cup a\cap c)\cup c\cup a\cap b\cup a\cap(\cup a\cap(a\cap b\cup a\cap c)\,\Delta.$$

ПРИМЕР 4. Если

$Э = \Delta\,0+0\,\Delta$, $Ю = \Delta\,0\,\Delta$ и $Ш = \Delta\,0+0+0+0+0+0\,\Delta$,

то

$Ш_I = \Delta\,0+0+0+0+0\,\Delta$, $Ш_2 = \Delta\,0+0+0+0\,\Delta$, $Ш_3 = Ш_4 = \Delta\,0\Delta$.

ПРИМЕР 5. Если

$Э = \Delta\,0\times a\,\Delta$, $Ю = \Delta\,0\,\Delta$ и $Ш = \Delta\,0\times a\times a\times a\times a\times a\times a\times 0\Delta$,

то

$Ш_I = Ш_2 = \Delta\,0\times a\times a\times a\times a\times a\times 0\Delta$, $Ш_3 = Ш_4 = \Delta\,0\times 0\,\Delta$.

ПРИМЕР 6. Если

$Э = \Delta / * \times \uparrow \Delta$, $Ю = \Delta + W / \Delta$

$Ш = \Delta \vee (/ * \times \uparrow) / * \times \uparrow (+ / * \times \uparrow) \iint / * \times \uparrow \vee / * \times \uparrow \Rightarrow$

$(/ * \times \uparrow / * \times \uparrow (/ * \times \uparrow)) \Delta$,

то

$Ш_I = \Delta \vee (+W) / * \times \uparrow (+ / * \times \uparrow) \iint / * \times \uparrow \vee / * \times \uparrow \Rightarrow$

$/ * \times \uparrow / * \times \uparrow (/ * \times \uparrow) \Delta$.,

$Ш_2 = Ш_3 = Ш_4 = \Delta \vee (+W) / * \times \uparrow (+ + W) \iint (+W) \vee + W \Rightarrow$

$/ * \times \uparrow / * \times \uparrow (+W) \Delta$

Рассмотрение примеров подобного сорта помогает выявить формальный алгоритм, при помощи которого человек ищет результат простой подстановки.

ПРИМЕР 7. Если

$Э = \Delta a + b \Delta$, $Ю = \Delta W \Delta$ и

$Ш = \Delta W + (a + b) \times W + a + b \Delta$

или

$Ш = \Delta a + b + W \times W + a + b \Delta$,

то

$Ш_I = \Delta W + W \times W + a + b \Delta$, $Ш_2 = Ш_3 = Ш_4 = \Delta W + W \times W + W \Delta$

ПРИМЕР 8. Если

$Э = \Delta W / \Delta$, $Ю = \Delta a + b \Delta$ и $Ш = \Delta W + W \times W + W \Delta$,

то

$Ш_I = \Delta a + b + W \times W + W \Delta$, $Ш_2 = Ш_3 = Ш_4 = \Delta a + b + (a + b) \times (a + b) + a + b \Delta$.

Последние два примера показывают, что вообще говоря ни одна модификация простой подстановки не является обратимой; результат любой простой подстановки и правило, по которому эта подстановка совершена, не позволяют однозначно восстановить исходную формулу Ш. Однако, в некоторых частных случаях это возможно. Например,

если правая часть правила Ю есть буква, отсутствующая в формуле Ш, то, применяя к результату $Ш_i$ строчную простую подстановку по правилу $\{Ю, Э\}$, мы очевидно получим почти нормальный вид исходной формулы Ш.

ПРИМЕР 9. Если Э = Δ (a×b) Δ , Ю = Δ W Δ и

Ш = Δ (a×b)↑(a×b×c)×sin×(a×b)×a×b+a×b Δ,

то

$Ш_I$ = Δ W↑(a×b×c)×sin×(a×b)×a×b+a×b Δ,

$Ш_2 = Ш_3 = Ш_4$ = Δ W↑(a×b×c)×sin×W×a×b+a×b Δ.

ПРИМЕР 10. Если Э = Δ (a+b) Δ , Ю = Δ W Δ и

Ш = Δ (a+b)↑2÷a+b+(a+b)+(a+b)×(a+b) Δ,

то

$Ш_I$ = Δ W↑2+a+b+a+b+(a+b)×(a+b) Δ,

$Ш_2 = Ш_4$ = Δ W↑2+a+b+W+W×W Δ,

$Ш_3$ = Δ W↑2+a+b+a+b+W+W Δ.

ПРИМЕР 11. Если

Э = Δ(a+b)×c Δ , Ю = Δ W Δ и Ш = Δ((a)+b)×c Δ,

то простая подстановка невозможна. В этом случае

$Ш_I = Ш_2 = Ш_3 = Ш_4$ = Δ (a+b)×c Δ .

Данный пример показывает, что в некоторых случаях результат полной или циклической простой подстановки содержит в себе подформулу, совпадающую с левой частью правила, и следовательно, допускает дальнейшее преобразование по этому правилу. Подобная ситуация не имеет места по крайней мере в следующих двух случаях: (1) исходная формула Ш почти нормальна; (2) простая подстановка возможна.

ПРИМЕР 12. Если Э = ΔΔ , Ю = Δ W Δ и

Ш = Δ ++(+a)×/(/a)+/a+/*a×Δ ,

то

$Ш_I$ = Δ W++(+a)×/(/a)+/a+/*a×Δ,

$Ш_2 = Ш_3 = Ш_4$ = Δ W+W+(W+a)×W/(W/a)+W/a+W/W*a×WΔ.

ПРИМЕР 13. Если Ю = Δ∫Δ , а формулы Э и Ш те же, что

и в предыдущем примере, то

$$Ш_I = \Delta\int{+}{+}(+a)\times/(/a)+/a+/*a\times\Delta,$$

$$Ш_2 = \Delta\int{+}\int{+}(\int{+}a)\times(\int)/((\int)/a)+(\int)/a+(\int)/(\int)$$
$$*a\times(\int)\Delta.$$

Полная и циклическая простые подстановки в данном случае зацикливают. Формулы $Ш_3$ и $Ш_4$ неопределены.

ПРИМЕР I4. Если Э = $\Delta\,a\,\Delta$, Ю = $\Delta\,a+a\,\Delta$ и

Ш = $\Delta\,a+b+a\,\Delta$, то

$Ш_I = \Delta\,a+a+b+a\,\Delta$ и $Ш_2 = \Delta\,a+a+b+a+a\,\Delta$.

Полная и циклическая простые подстановки в данном случае также зацикливают.

Предположение, что полная и циклическая простые подстановки зацикливают тогда и только тогда, когда левая часть правила Э входит в правую часть Ю, не верно ни в одну сторону. В самом деле, если формула Э не входит в формулу Ш, то простая подстановка невозможна и зацикливания нет независимо от того, входит или не входит формула Э в формулу Ю. Далее, если формула Э не является почти нормальной и не представима в виде $\Delta\,(\widetilde{Э}_I)\,\Delta$, где $Э_I$ – почти нормальная формула, то независимо от вхождения Э в Ю формула Э не является подформулой никакой почти нормальной формулы. Поэтому в данном случае зацикливания также нет.

Если же формула Э почти нормальна и входит в обе формулы Ш и Ю, то она очевидно будет входить и в результаты $Ш_I$ и $Ш_2$ однократной и строчной простых подстановок в формулу Ш по правилу $\{Э, Ю\}$. Значит, в этом случае зацикливание неизбежно. То же самое имеет место и в том случае, когда формула Э имеет вид $\Delta(\widetilde{Э}_I)\,\Delta$, где $Э_I$ – почти нормальная формула и входит в формулу Ш и в почти нормальный вид формулы Ю. Тем самым установлено

ПРЕДЛОЖЕНИЕ 9. Пусть формула Э входит в обе формулы Ш и Ю. Для зацикливания полной и циклической простых подстановок в формулу Ш по правилу $\{Э, Ю\}$ достаточно, чтобы по крайней мере одна из формул этого правила была почти нормальна.

Приведенные ниже примеры I5–I7 показывают, что в том случае, когда формулы Э и Ю обе не являются почти нормальными (но Э = $\Delta(\widetilde{Э}_I)\,\Delta$, где формула $Э_I$ почти нормальна) и формула Э

входит в формулы Ш и Ю, зацикливание возможно, но не обязательно. Значит, указанное в предложении 9 достаточное условие зацикливания не является необходимым.

ПРИМЕР I5. Пусть: Э = Ю = $\Delta\,(\alpha+\beta)\Delta$. Если Ш = $= \Delta\,(\alpha+\beta)\times\alpha\,\Delta$, то полная и циклическая простые подстановки зацикливают. Если же Ш = $\Delta\;(\alpha+\beta)+\alpha\,\Delta$, то

$$Ш_1 = Ш_2 = Ш_3 = Ш_4 = \Delta\;\alpha+\beta+\alpha\;\Delta\,.$$

Зацикливания в этом случае очевидно нет.

ПРИМЕР I6. Пусть

Э = $\Delta\,(\alpha+\beta)\Delta$ и Ю = $\Delta\;(\alpha+\beta)\times\alpha+(\alpha+\beta)\,\Delta$.

Полная и циклическая простые подстановки по правилу {Э, Ю} в формулу Ш зацикливают всякий раз, когда простая подстановка возможна.

ПРИМЕР I7. Пусть

Э = $\Delta\,(\alpha+\beta)\,\Delta$ и Ю = $\Delta\,(\alpha+\beta)+\alpha\,\Delta$.

Полная и циклическая простые подстановки по правилу {Э, Ю} в любую формулу Ш никогда не зацикливает.

Для практики важно иметь надежные и достаточно общие критерии невозможности зацикливания полной и циклической простых подстановок. Несколько таких критериев указано в следующем очевидном предложении:

ПРЕДЛОЖЕНИЕ I0. Пусть формулы Э и Ю почти нормальны и Э не входит в Ю. Для того чтобы зацикливание полной и циклической простых подстановок по правилу {Э, Ю} было невозможным, достаточно, чтобы выполнялось по крайней мере одно из следующих условий:

(α) Для некоторой буквы или связи α число членов формулы Э, совпадающих с α , больше, чем число таких же членов формулы Ю.

(β) В формуле Э есть такая буква или связь, которой нет в формуле Ю.

(γ) Формула Э длиннее формулы Ю.

(δ) Главные связи формул Э и Ю различны.

(ε) Главная связь хотя бы одной из формул Э и Ю неассоциативна.

Следующие примеры показывают, что зацикливание возможно и в том случае, когда формула Э не входит в формулу Ю.

ПРИМЕР I8. Пусть

$$Э = \Delta\,(a+b)\,\Delta \quad , Ю = \Delta\,a+b\,\Delta \quad \text{и } Ш = \Delta\,(a+b)+d\,\Delta.$$

Однократная и строчная простые подстановки здесь возможны и не меняют формулу Ш. Следовательно, полная и циклическая простые подстановки зацикливают.

ПРИМЕР I9. Пусть

$$Э = \Delta\,a+b\,\Delta \quad , Ю = \Delta\,(a)+(b)+d\,\Delta$$

и Ш – любая формула. Если простая подстановка по правилу Э, Ю возможна, то полная и циклическая простые подстановки по этому правилу зацикливают.

ПРИМЕР 20. Пусть

$$Э = \Delta\,a+b+b+a\,\Delta \quad , Ю = \Delta\,b+a+b+a+a+b+a+b\,\Delta$$

и

$$Ш = \Delta\,a+b+a+b+b+a+b+a\,\Delta.$$

Последовательное применение однократной и соответственно строчной простой подстановки по правилу {Э, Ю} сначала к исходной формуле Ш, а затем к результату предыдущей простой подстановки дает нам последовательности следующих формул

$$Ш_I' = \Delta\,a+b+b+a+b+a+a+b+a+b+b+a\,\Delta,$$

$$Ш_I'' = \Delta\,b+a+b+a+a+b+a+b+b+a+a+b+a+b+b+a\,\Delta,$$

$$Ш_I''' = \Delta\,b+a+b+a+a+b+b+a+b+a+a+b+a+b+a+b+a+b+b+a\,\Delta,$$

$$Ш_I^{IV} = \Delta\,b+a+b+a+b+a+b+a+a+b+a+b+b+a+a+b+a+b+a+b+a+b+a+b+a+b+a\,\Delta,$$

. . . . . .

и соответственно формул

$$Ш_2' = \Delta\,a+b+b+a+b+a+a+b+a+b+b+a\,\Delta,$$

$$Ш_2'' = \Delta\,b+a+b+a+a+b+a+b+b+a+a+b+b+a+a+b+a+b+a+a+b+a+b\,\Delta,$$

$$Ш_2''' = \Delta\,b+a+b+a+a+b+b+a+b+a+a+b+a+b+b+a+b+a+a+b+a+b+a+a+b+a+b+b+a+a+b+a+b\,\Delta,$$

$$Ш_2^{IV} = \Delta\,b+a+b+a+b+a+b+a+a+b+a+b+b+a+a+b+a+b+b+a+a+b+b+a+a+b+a+b+a+b+a+b+a+a+b+a+b+a+b+a+b+a+b+a+b+a+b\,\Delta,$$

. . . . . .

Легко видеть, что формулы $Ш_I^{(m)}$ и $Ш_2^{(m)}$ при любом натуральном m допускают дальнейшее преобразование по правилу {Э, Ю}. Следо-

вательно, полная и циклическая простые подстановки в рассматриваемом случае зацикливают.

ПРИМЕР 21. Пусть

$$Э = \Delta\, a\,b\,b\,a\, \Delta \quad , \quad Ю = \Delta\, b\,a\,b\,a\,w\,a\,b\,a\,b\, \Delta$$

и

$$Ш = \Delta\, a\,b\,a\,b\,b\,a\, \Delta .$$

Последовательное применение однократной или, что в данном случае то же самое, строчной простой подстановки дает нам следующую последовательность формул

$$Ш_1' = Ш_2' = \Delta\, a\,b\,b\,a\,b\,a\,w\,a\,b\,a\,b\, \Delta ;$$

$$Ш_1'' = Ш_2'' = \Delta\, b\,a\,b\,a\,w\,a\,b\,a\,b\,b\,a\,w\,a\,b\,a\,b\, \Delta ;$$

$$Ш_1''' = Ш_2''' = \Delta\, b\,a\,b\,a\,w\,a\,b\,b\,a\,b\,a\,w\,a\,b\,a\,b\,w\,a\,b\,a\,b\, \Delta ;$$

$$Ш_1^{IV} = Ш_2^{IV} = \Delta\, b\,a\,b\,a\,w\,b\,a\,b\,a\,w\,a\,b\,a\,b\,w\,a\,b\,a\,b\,w\,a\,b\,a\,b\, \Delta .$$

. . . . . .

Ясно, что в этом случае полная и циклическая простые подстановки зацикливают.

ПРИМЕР 22. Пусть формулы Э и Ю те же самые и $Ш = \Delta\, a\,b\,b\,a\,b\,a\,b\,b\,a\, \Delta$. Последовательное применение однократной простой подстановки дает нам следующую последовательность формул

$$Ш_1' = \Delta\, b\,a\,b\,a\,w\,a\,b\,a\,b\,b\,a\,b\,b\,a\, \Delta ;$$

$$Ш_1'' = \Delta\, b\,a\,b\,a\,w\,a\,b\,b\,a\,b\,a\,w\,a\,b\,a\,b\,b\,b\,a\, \Delta ;$$

$$Ш_1''' = \Delta\, b\,a\,b\,a\,w\,b\,a\,b\,a\,w\,a\,b\,a\,b\,b\,a\,w\,a\,b\,a\,b\,b\,b\,a\, \Delta ;$$

$$Ш_1^{IV} = \Delta\, b\,a\,b\,a\,w\,b\,a\,b\,a\,w\,a\,b\,b\,a\,b\,a\,w\,a\,b\,a\,b\,w\,a\,b\,a\,b\,b\,b\,a\, \Delta .$$

. . . . . .

Ясно, что полная простая подстановка зацикливает. С другой стороны, результат строчной простой подстановки в формулу Ш по правилу {Э, Ю} имеет вид

$$Ш_2 = \Delta\, b\,a\,b\,a\,w\,a\,b\,a\,b\,b\,b\,a\,b\,a\,w\,a\,b\,a\,b\, \Delta .$$

Дальнейшее применение операции простой подстановки по правилу {Э, Ю} уже невозможно. Значит, циклическая простая подстановка в данном случае не зацикливает и ее результат совпадает с формулой $Ш_2$.

Последний пример показывает, что полная простая подстановка может зациклить даже в том случае, когда циклическая простая подстановка не циклит. Неизвестно, существуют ли такие формулы Ш, Э и Ю, что полная простая подстановка в формулу Ш по правилу {Э, Ю} не циклит, а циклическая – зацикливает.

Операция простой подстановки позволяет решать некоторые задачи аналитического характера. Примером такой задачи может служить задача вычисления производной от элементарной функции.

Блок-схема алгоритма решения этой задачи указана на

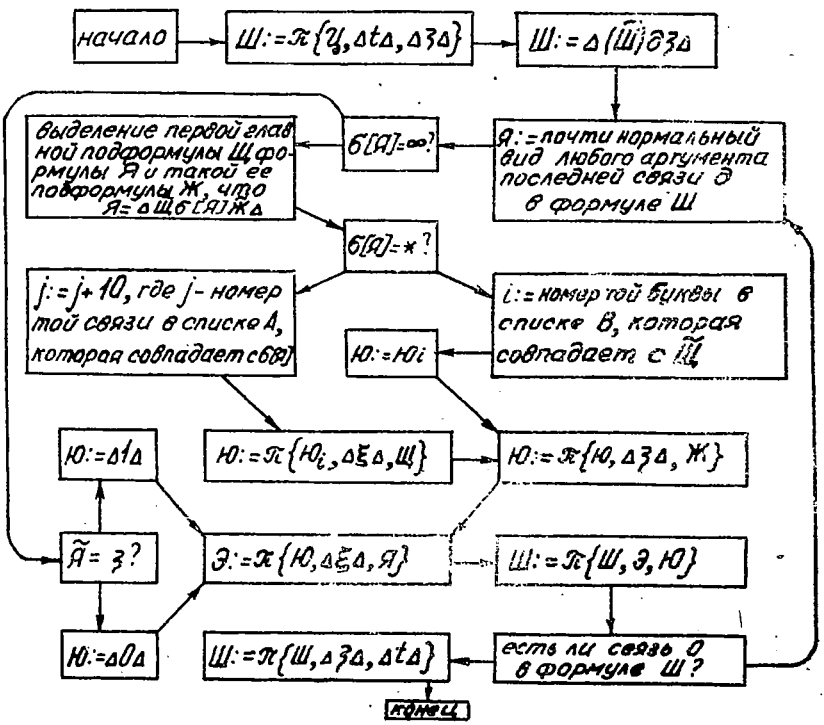


Рис. I. Алгоритм дифференцирования

рис.I. В этой блок-схеме использованы следующие обозначения:

Ц – формула, изображающая исходную функцию;

t – буква, изображающая переменную дифференцирования;

Ш – формула, изображающая результат (т.е.производную функции Ц по параметру t);

ξ , η и ζ – буквы, являющиеся внутренними (или "рабочими") параметрами алгоритма; предполагается, что эти буквы в формуле Ц не содержатся;

Ж, Щ, Э, Ю и Я – формулы, возникающие в качестве промежуточных результатов в процессе работы алгоритма ("рабочие формулы");

$\sigma[Я]$ – главная связь формулы Я;

А и В суть соответственно упорядоченные множества элементарных символов:

$$\{ + \times / \uparrow \log \} \quad , \{ \exp \ \ln \ \sin \ \cos$$

$$\operatorname{tg} \ \operatorname{ctg} \ \arcsin \ \arccos \ \operatorname{arctg} \ \operatorname{arcctg} \} ;$$

i и j – рабочие параметры алгоритма, принимающие натуральные значения I, 2, ... ;

π – операция строчной простой подстановки; результат применения этой операции к формуле Ш по правилу $\{Э, Ю\}$ обозначается через $\pi\{Ш, Э, Ю\}$;

$Ю_0$, $Ю_I$, ... , $Ю_{I5}$ суть соответственно формулы

$$\Delta\xi\partial\zeta\Delta \exp\ast\eta\times\eta\partial\zeta\Delta\eta\partial\zeta/\eta\Delta\cos\ast\eta\times\eta\partial\zeta\Delta\bar{I}\times\sin\ast$$

$$\eta\times\eta\partial\zeta\Delta\eta\partial\zeta/\cos\ast\eta\uparrow 2\Delta\bar{I}\times\eta\partial\zeta/\sin\ast\eta\uparrow 2\Delta\eta\partial\zeta/($$

$$1+\bar{I}\times\eta\uparrow 2)\uparrow\tfrac{1}{2}\Delta\bar{I}\times\eta\partial\zeta/(1+\bar{I}\times\eta\uparrow 2)\uparrow\tfrac{1}{2}\Delta\eta\partial\zeta/(1+\eta\uparrow 2$$

$$)\Delta\bar{I}\times\eta\partial\zeta/(1+\eta\uparrow 2)\Delta\xi\partial\zeta+\tfrac{1}{2}\partial\zeta\Delta\xi\partial\zeta\ast\eta+\zeta\times\eta\partial\zeta\Delta(\xi\partial\zeta\times\eta$$

$$+\bar{I}\times\xi\times\eta\partial\zeta)/\eta\uparrow 2\Delta\xi\uparrow\eta\times(\xi\partial\zeta\times\eta/\xi+\ln\ast\zeta\times\eta\partial\zeta)\Delta(\zeta\times\eta\partial\zeta$$

$$+\bar{I}\times\xi\cos\eta\times\xi\partial\zeta\times\zeta)/(\ln\ast\xi\times\zeta\times\eta)\Delta ,$$

где символом $\bar{I}$ обозначено вещественное число – I;

знаки + и –, поставленные около стрелок, заменяют слова "да" (или "условие выполнено") и соответственно "нет" (или "условие не выполнено").

§ IO. ОБОБЩЕННАЯ ПОДСТАНОВКА

Операция простой подстановки в принципе позволяет проводить любые аналитические преобразования. Однако она требует точного указания, какую подформулу преобразуемой формулы нужно заменять и какой именно. Например, если желательно преобразовать выражение

$$Ш = \Delta\,(\sin * t + \cos * t)\,\partial t + t\,\partial t\,\Delta,$$

используя известный закон дифференцирования суммы (производная суммы равна сумме производных), то мы должны указать правило

$$\{Э,\ Ю\} = \{\Delta\,(\sin * t + \cos * t)\,\partial t\,\Delta\,\sin * t\,\partial t + \cos * t\,\partial t\,\Delta\}$$

Аналогичное преобразование выражения

$$Ш = \Delta\,(t\uparrow 2 + \ln * t)\,\partial t + t\,\partial t\,\Delta$$

требует уже иного правила

$$\{Э,\ Ю\} = \{\Delta\,(t\uparrow 2 + \ln * t)\,\partial t\,\Delta\,(t\uparrow 2)\,\partial t + \ln * t\,\partial t\,\Delta\}$$

Ясно, что в алгоритме, предназначенном для дифференцирования произвольной элементарной функции и опирающемся только на операцию простой подстановки, перечислить все необходимые правила невозможно. Значит, такой алгоритм должен сам генерировать эти правила. В частности, закон дифференцирования суммы может быть реализован следующим образом (см. рис.I). В исходной формуле Ш ищется связь ∂, у которой почти нормальный вид левого аргумента Я имеет главную связь +. В формуле Я выделяются такие подформулы Щ и Ж, что $Я = \Delta\ \widetilde{Щ} + \widetilde{Ж}\ \Delta$, и образуются правила

$$\{\Delta\,\xi\,\Delta,\ Щ\},\ \{\Delta\,\eta\,\Delta,\ Ж\},\ \{\Delta\,\zeta\,\Delta\,t\,\Delta\}$$

Строчная простая подстановка по этим правилам в формулы

$$Э = \Delta\,(\xi + \eta)\,\partial\zeta\,\Delta \qquad , \quad Ю = \Delta\,\xi\,\partial\zeta + \eta\,\partial\zeta\,\Delta \qquad (27)$$

приводит к почти нормальным видам Э′ и Ю′ формул

$$\Delta\,(\widetilde{Щ} + \widetilde{Ж})\,\partial t\,\Delta\,(\widetilde{Щ})\,\partial t + (\widetilde{Ж})\,\partial t\,\Delta$$

соответственно. Из формул Э′ и Ю′ образуется искомое правило $\{Э',\ Ю'\}$, которое и применяется к исходной формуле Ш.

Реализация законов дифференцирования произведения, дроби, синуса и т.д. аналогична, только формулы (27) заменяются соответственно формулами

$$\begin{aligned} Э_1 &= \Delta(\xi * \eta)\,\partial\zeta\,\Delta \quad , & Ю_1 &= \Delta\,\xi\,\partial\zeta * \eta + \xi * \eta\,\partial\zeta\,\Delta, \\ Э_2 &= \Delta(\xi/\eta)\,\partial\zeta\,\Delta \quad , & Ю_2 &= \Delta(\xi\,\partial\zeta + \bar{1} * \xi * \eta\,\partial\zeta)/\eta\uparrow 2\,\Delta, \\ Э_3 &= \Delta\,sin * \eta\,\partial\zeta\Delta \quad , & Ю_3 &= \Delta\,cos * \eta * \eta\,\partial\zeta\,\Delta. \end{aligned} \qquad (28)$$

. . . . . .

Подобным же образом можно организовать генерацию правил, необходимых для приведения подобных членов, умножения или деления полинома на полином, разложения функции в ряд Тейлора, решения системы линейных алгебраических уравнений с буквенными коэффициентами и др.

Во всех перечисленных случаях основу схемы генерации правил простой подстановки составляют пары формул типа (27) или (28), где некоторые буквы ξ , η , ζ , ... носят так сказать "переменный" характер, т.е. могут быть заменены любыми подформулами; эти пары формул фактически выступают как записи правил в общем виде. Буквы ξ , η , ζ , ... , входящие в такие "обобщенные правила" и допускающие замену себя на любую подформулу, удобно назвать переменными буквами и условиться выбирать их из множества переменных букв Ф (см. § I). Схема генерации конкретных правил из данного обобщенного правила типа (27) или (28), иллюстрированная выше на конкретном примере, без особого труда алгоритмизируется в самом общем виде. Соединяя эту схему с алгоритмом простой подстановки, мы повидимому должны получить алгоритм подстановки по обобщенному правилу, т.е. алгоритм, осуществляющий операцию обобщенной подстановки.

Интуитивное содержание операции обобщенной подстановки проясняется следующими конкретными примерами. Если обобщенное правило совпадает с записью закона дифференцирования суммы (27), то обобщенная подстановка в произвольную формулу Ш по этому правилу должна заменить в ней каждую (или хотя бы одну) подформулу вида $Я = \Delta\,(\widetilde{Ш} + \widetilde{Ж})\,\partial t\,\Delta$ соответствующей подформулой $\Delta\,(\widetilde{Ш})\,\partial t + (\widetilde{Ж})\,\partial t\,\Delta$; в частности, результатом преобразования формулы

$$Ш = \Delta(cos * t + (s\uparrow 2 + t\_s\uparrow s)\,\partial s + (ln * (x+y) + x)\,\partial x)\,\partial t\,\Delta \qquad (29)$$

по правилу (27) может быть объявлена любая из следующих формул

Ш1 = Δ cos*t ∂t + ((s↑2 + t↑s) ∂s + (ln*(x+y)+x) ∂x) ∂t Δ,

Ш2 = Δ (cos*t + (s↑2 + t↑s) ∂s) ∂t + ((ln*(x+y)+x) ∂x) ∂t Δ,

Ш3 = Δ cos*t ∂t + ((s↑2 + t↑s) ∂s) ∂t + ((ln*(x+y)+x) ∂x) ∂t Δ,

Ш4 = Δ (cos*t + (s↑2) ∂s + (t↑s) ∂s + ln*(x+y) ∂x + x ∂x) ∂t Δ,

Ш5 = Δ cos*t ∂t + ((s↑2) ∂s) ∂t + ((t↑s) ∂s) ∂t + (ln*(x+y) $\frac{+(x\,\partial x)\partial t}{\partial x}$) ∂t Δ.

. . . . . .

Преобразование формулы

Ш = Δ(a+b↑2+cos*t+2)×(a+b+c)+(a+a)×(a+b+c) Δ

по правилу

{ Δ (ξ + η)×(τ + τ) Δ ξ×ζ + η×ζ + ξ×τ + η×τ Δ },

где ξ , η , ζ и τ – переменные буквы, должно привести ее к виду

Ш1 = Δa×a + (b↑2 + cos*t+2)×a + a×(b+c) + (b↑2+cos*t+2)×(b+c)
+ (a+a)×(a+b+c) Δ

или

Ш2 = Δ a×b + (b↑2+cos*t+2)×a + a×(b+c) + (b↑2+cos*t+2)×
(b+c) + a×a + a×a + a×(b+c) + a×(b+c) Δ,

или, в конечном счете, к виду

Ш3 = Δ a×b + (b↑2 + cos*t+2)×a + a×(b+c) + b↑2×b + (cos*t+2)
×b + b↑2×c + (cos*t+2)×c + a×a + a×a + a×(b+c) + a×(b+c) Δ.

Дополнительное поочередное применение правил

{Δ ξ×(η+ζ) Δ ξ×η + ξ×ζ Δ}; {Δ(ξ+η)×ζ Δ ξ×ζ + η×ζ Δ}

к формуле Ш3 позволяет завершить процесс раскрытия скобок в исходной формуле Ш и получить окончательный результат в виде

Ш4 = Δ a×b + b↑2×a + cos*t×a + 2×a + a×b + a×c + b↑2×b + cos*t×b +
2×b + b↑2×c + cos*t×c + 2×c + a×a + a×a + a×b + a×c + a×b + a×c Δ.

В некоторых случаях переменные буквы могут обозначать произвольные подформулы определенного типа. Например, задача приведения подобных членов в формуле

Ш = Δ a×2 + a×3 + a×6×5 + a×6×2 + a↑2×8 + a↑2×I Δ

решается применением обобщенной подстановки по правилу

{ Δ ξ×η + ξ×ζ Δ ξ×(η+ζ) Δ }

где ξ – переменная буква "общего типа", а η и ζ – переменные буквы типа "число"; результатом такой подстановки является формула

Щ = Δ a×(2+3) + a×6×(5+2) + a↑2×(8+I) Δ

Перейдем теперь к строгому описанию операции обобщенной подстановки и сопутствующих ей понятий. Как указано в § I, алфавит Авто-Аналитика Г состоит из множества специальных символов Ⓐ, множества связей Σ и множества букв П, причем последнее множество разбивается еще на подмножества постоянных букв Ω, переменных букв Φ и чисел R. В дополнение к условиям, указанным ранее, предположим, что множество Φ также разбито на несколько попарно непересекающихся подмножеств Φ_0, Φ_I, ... , Φ_ν (количество этих множеств не фиксируется и зависит от конкретной машинной реализации Авто-Аналитика; в существующей реализации Авто-Аналитика на БЭСМ-6 $\nu = 7$) и каждому множеству Φ_i, $i = 0, I, \dots, \nu$, поставлено в соответствие некоторое непустое семейство подформул Φ_i^*. В частности, семейство Φ_2^* составим из всех подформул вида Δ α Δ, где α – число. Семейство Φ_I^* составим из всех подформул вида Δ α Δ, где α – буква. Будем, далее, считать, что подформула Щ = Δ α_i α_{i+1} ... α_j Δ, входящая в формулу

Ш = Δ α_1 α_2 ... α_{i-1} α_i α_{i+1} ... α_j α_{j+1} ... α_n Δ

начиная с i-го места, принадлежит классу Φ_3^*, если главная связь в формулы Щ сильнее (или, что то же самое, отлична от) символов α_{i-1} и α_{j+1}. Подформулы класса Φ_3^* именуются в дальнейшем одночленами. В семейство Φ_0^* отнесем все подформулы вообще. Прочие семейства Φ_4^*, Φ_5^*, ... , Φ_ν^* в настоящее время не фиксированы; они зарезервированы для возможных расширений Авто-Аналитика в будущем.

Иногда нам будет удобно говорить: Переменная буква ξ име-

ет тип Φ_i . Это выражение всегда означает, что рассматриваемая переменная буква принадлежит множеству Φ_i .

ОПРЕДЕЛЕНИЕ 22. Аналогом переменной буквы $\xi \in \Phi_i$, $0 \le i \le \nu$, объявляется любая подформула класса Φ_i^* . Пусть, далее, $\xi_1, \xi_2, \ldots, \xi_m$ – все без исключения (попарно различные) переменные буквы левой части Э правила $\{Э, Ю\}$ и $Ш_1, Ш_2, \ldots, Ш_m$ – некоторые их аналоги. Формулы Э′ и Ю′ , полученные путем одновременной замены каждой переменной буквы ξ_κ , $\kappa = 1, 2, \ldots, m$, в формулах Э и Ю ее аналогом $Ш_\kappa$ (заключенным в случае надобности в скобки) называются аналогом левой части Э и соответствующим аналогом правой части Ю.

В качестве примера рассмотрим правило

$$\{Э, Ю\} = \{\Delta\xi \times \eta + \xi \times \zeta \Delta \xi \times (\eta + \zeta)\Delta\}, \qquad (30)$$

где ξ , η и ζ – переменные буквы, причем $\xi \in \Phi_0$, а η и $\zeta \in \Phi_2$, и формулу

$$Ш = \Delta(a \times 2 + a \times 3 + a \times в) \times 2 + (a \times 2 + a \times 3 + a \times в) \times 2 + a \times 2\ \Delta \qquad (31)$$

Среди подформул этой формулы аналогами левой части Э правила (30) являются подформулы

$$\Delta(a \times 2 + a \times 3 + a \times в) \times 2 + (a \times 2 + a \times 3 + a \times в) \times 2\ \Delta$$

$$a \times 2 + a \times 3\ \Delta\ a \times 2 + a \times 3\ \Delta,$$

входящие в формулу Ш начиная с 1-го, со 2-го и с 18-го мест соответственно. Соответствующие аналоги правой части Ю имеют соответственно вид

$$\Delta(a \times 2 + a \times 3 + a \times в) \times (2 + 2)\Delta\ a \times (2 + 3)\ \Delta\ a \times (2 + 3)\Delta$$

Если в рассматриваемой ситуации допустить, что переменные буквы η и ζ также имеют тип Φ_0, то кроме указанных выше аналогов в формуле Ш появятся еще два аналога, а именно, подформула $\Delta\ a \times 3 + a \times в\ \Delta$, входящая с 6-го места, и такая же подформула, начинающаяся с 22-го места. Если все переменные буквы правила (30) имеют тип Φ_1, то в формуле (31) содержится четыре аналога левой части Э, а именно таковы подформулы

$$\Delta\, a \times 2 + a \times 3 \Delta\ a \times 3 + a \times в\ \Delta\ a \times 2 + a \times 3 \Delta\ a \times 3 + a \times в\ \Delta,$$

входящие в формулу (31) начиная со 2-го, с 6-го, с 18-го и с 22-го мест соответственно.

Аналогами левой части правила

$\{Э, Ю\} = \{\Delta \xi + 0 \Delta \xi \Delta\}$

где $\xi \in \Phi_3$, в формуле

$Ш = \Delta\, 0 + a \times b + 0 + (a + 0) \times b + 0 \Delta$

служат три подформулы

$\Delta\, a \times b + 0 \Delta \quad (a + 0) \times b + 0 \Delta \quad a + 0 \Delta$

Если здесь потребовать, чтобы ξ принадлежало Φ_I, то в формуле Ш останется только один аналог $\Delta\, a + 0 \Delta$. В случае $\xi \in \Phi_0$ формула Ш содержит семь аналогов левой части данного правила:

$\Delta\, 0 + a \times b + 0 \Delta \quad 0 + a \times b + 0 + (a + 0) \times b + 0 \Delta \quad a \times b + 0 \Delta \quad a \times b + 0 +$

$(a + 0) \times b + 0 \quad \Delta\, 0 + (a + 0) \times b + 0 \Delta \quad (a + 0) \times b + 0 \Delta \quad a + 0 \Delta$

Рассмотрим еще правило

$\{Э, Ю\} = \{\Delta \xi + \xi \Delta \xi \times 2 \Delta\}$

и формулу

$\Delta\, a + a + a + b + b + b + b + a + a + b + b + b + b + a + a + b + b + b + b \Delta$

Если $\xi \in \Phi_2$, то левая часть $Э = \Delta \xi + \xi \Delta$ в формуле Ш не имеет ни одного аналога. Если $\xi \in \Phi_I$, то формула Э имеет в формуле Ш 13 аналогов. В случае $\xi \in \Phi_3$ аналоги те же самые. Наконец, в случае $\xi \in \Phi_0$ формула Ш содержит 23 аналога.

Отношения порядка между формулами и подформулами, заданные выше определениями 2 и 6, индуцируют во множестве всех аналогов левой части правила $\{Э, Ю\}$, входящих в данную формулу Ш, некоторые отношения линейного порядка (точнее квазипорядка). В определении операции обобщенной подстановки нам потребуется еще одно отношение порядка между аналогами, определяемое следующим образом.

ОПРЕДЕЛЕНИЕ 23. Левая часть Э правила $\{Э, Ю\}$ имеет вид

$Э = \Delta\, Б_0 \xi_1 Б_1 \xi_2 Б_2 \ldots \xi_{m-1} Б_{m-1} \xi_m Б_m \Delta,$

где $m \geqslant 0$, $\xi_1, \xi_2, \ldots, \xi_m$ – переменные буквы и $Б_0, Б_I, \ldots, Б_m$ – последовательности элементарных символов, не содержащие переменных букв (последовательности $Б_0$ и $Б_m$ могут быть пустыми; остальные последовательности $Б_I, Б_2, \ldots, Б_{m-1}$ не пусты). Поэтому всякий аналог Э′ формулы Э может быть запи-

сан в виде

$$Э' = \Delta\, б_0\, \widetilde{Ш}'_1\, б_1\, \widetilde{Ш}'_2\, б_2 \ldots \widetilde{Ш}'_{m-1}\, б_{m-1}\, \widetilde{Ш}'_m\, б_m\, \Delta,$$

где $Ш_i$ для всякого $i = 1,2, \ldots, m$ есть аналог переменной буквы ξ, причем $Ш_i = Ш_j$ в случае $\xi_i = \xi_j$. Пусть

$$Э = \Delta\, б_0\, \widetilde{Ш}''_1\, б_1\, \widetilde{Ш}''_2\, б_2 \ldots \widetilde{Ш}''_{m-1}\, б_{m-1}\, \widetilde{Ш}''_m\, б_m\, \Delta$$

– еще один аналог формулы Э, причем Э′ и Э″ входят в формулу Ш соответственно с κ-го и с ℓ-го мест. Будем считать, что эти аналоги совпадают в том и только в том случае, когда $\kappa = \ell$ и $Ш'_i = Ш''_i$ для всех $i = 1,2, \ldots, m$ (Например, в формуле $\Delta\, a+a+a\, \Delta$ – четыре аналога формулы $Э = \Delta\, \xi + \eta\, \Delta$, где ξ и $\eta \in \Phi_3$, в формуле $\Delta\, a+a+a+a\, \Delta$ – десять аналогов, и вообще, в формуле $\Delta\, a+a+\ldots+a\, \Delta$, состоящей из n слагаемых $\Delta\, a\, \Delta$, содержится $n(n+1)/2$ подформул, но $n(n^2-1)/6$ различных аналогов формулы Э).

Мы говорим, что аналог Э′ мажорируется аналогом Э″, если выполнено по крайней мере одно из следующих условий:

(a) Аналог Э в формуле Ш начинается раньше аналога Э″, т.е. $\kappa < \ell$.

(b) Оба аналога Э′ и Э″ начинаются в формуле Ш одновременно, т.е. $\kappa = \ell$, и существует такое i, $1 < i \le m$, что $Ш'_1 = Ш''_1$, $Ш'_2 = Ш''_2$, $\ldots$, $Ш'_{i-1} = Ш''_{i-1}$, но $Ш'_i$ слабее в смысле определения 2, чем $Ш''_i$ (или, что в данном случае одно и то же, $Ш'_i$ в формуле Ш предшествует в смысле определения 6 подформуле $Ш''_i$ или еще, $Ш'_i$ короче, чем $Ш''_i$).

(c) Аналоги Э′ и Э″ совпадают, т.е. $\kappa = \ell$ и $Ш'_i = Ш''_i$ для всех $i = 1,2, \ldots, m$.

ПРИМЕРЫ. Если перечислить все аналоги формулы

$$Э = \Delta\, \xi + \eta\, \Delta, \quad \xi \text{ и } \eta \in \Phi_0$$

входящие в формулу

$$Ш = \Delta\, a+b+c+d+e\, \Delta$$

так, чтобы каждый предыдущий аналог мажорировался каждым следующим, то получится следующий список формул

$\Delta\, a+b\, \Delta\; a+b+c\, \Delta\; a+b+c+d\, \Delta\; a+b+c+d+e\, \Delta\; a+b+c\, \Delta\; a+b+c+d\, \Delta\; a+b+c+d+e\, \Delta\; a+b+c+d+e\, \Delta\; b+c\, \Delta\; b+c+d\, \Delta\; b+c+d+e\, \Delta\; b+c+d\, \Delta\; b+c+d+e\, \Delta\; b+c+d+e\, \Delta\; c+d\, \Delta\; c+d+e\, \Delta\; c+d+e\, \Delta\; d+e\, \Delta,$

где точкой отмечены те связи, которые в каждом аналоге разделяют аналоги переменных букв ξ и η.

В случае

Ш = $\Delta\, a \times b \times c \times d + e \times f \times g \times h\, \Delta$

и

Э = $\Delta \xi \times \eta + \zeta \times \tau\, \Delta$

где ξ, η, ζ и τ Φ_0, аналогичный список имеет вид

$\Delta\, a \dot{\times} b \times c \times d \dot{+} e \dot{\times} f \times g \times h\, \Delta$ $a \dot{\times} b \times c \times d \dot{+} e \times f \dot{\times} g \times h\, \Delta$ $a \dot{\times} b \dot{\times} c \times d \dot{+} e \times f \times g \times h\, \Delta$ $a \times b \dot{\times} c \times d \dot{+} e \dot{\times} f \times g \times h\, \Delta$ $a \times b \times c \dot{\times} d \dot{+} e \times f \times g \dot{\times} h\, \Delta$ $a \times b \times c \times d \dot{+} e \dot{\times} f \times g \times h\, \Delta$ $a \times b \times c \times d \dot{+} e \times f \dot{\times} g \times h\, \Delta$ $a \times b \times c \times d \dot{+} e \times f \times g \dot{\times} h\, \Delta$

Приведенные примеры показывают, что отношения "аналог Э′ мажорируется аналогом Э″ и "подформула Э′ предшествует подформуле Э″" вообще говоря различны. Нетрудно однако показать, что эти отношения эквивалентны по крайней мере в том случае, когда все переменные буквы формулы Э содержатся в объединении $\Phi_2 \cup \Phi_1 \cup \Phi_3$.

ОПРЕДЕЛЕНИЕ 24. Пусть заданы формула Ш и правило {Э, Ю}. Если в формуле Ш нет ни одного аналога формулы Э, то мы говорим, что обобщенная подстановка в формулу Ш по правилу {Э,Ю} невозможна. Результатом обобщенной подстановки в этом случае объявляется почти нормальный вид исходной формулы Ш.

Если же формула Ш содержит хотя бы один аналог формулы Э, то она единственным образом представима в виде

Ш = $\Delta\, \alpha_1 \ldots \alpha_{i_1 - 1}\, \widehat{Э}_1\, \alpha_{j_1} \ldots \alpha_{i_2 - 1}\, \widehat{Э}_2\, \alpha_{j_2} \ldots \ldots \ldots \alpha_{i_\sigma - 1}\, \widehat{Э}_\sigma\, \alpha_{j_\sigma} \ldots$

$\alpha_n\, \Delta$,

где индексы i_1, i_2, ... , i_σ и подформулы $Э_I$, $Э_2$, ... , $Э_\sigma$ удовлетворяют следующим четырем условиям:

(α) При любом $\ell = 1,2, \ldots, \sigma$ в формуле Ш начиная с i -го места входит некоторый аналог $Э_\ell$ формулы Э.

(β) Аналог $Э_I$ мажорируется всеми другими аналогами формулы Э, входящими в формулу Ш.

(γ) При любом $\ell = 1,2,3, \ldots, \sigma$ аналог $Э_\ell$ не соприкасается с аналогом $Э_{\ell-1}$, мажорирует его и сам мажорируется всеми аналогами формулы Э, входящими в формулу Ш, мажорирующими аналог $Э_{\ell-1}$ и не соприкасающимися с ним.

($c1$) В формуле Ш нет аналога формулы Э, мажорирующего аналог $Э_j$ и не соприкасающегося с ним.

В этом случае мы говорим, что обобщенная подстановка возможна. Результатом однократной и соответственно строчной обобщенной подстановки в формулу Ш по правилу {Э, Ю} называется почти нормальный вид формулы

$$Ш_I = a\,\alpha_1 \ldots \alpha_{i_1-1}\,(\widetilde{Ю}_1)\,\alpha_{j_1} \ldots \alpha_{i_2-1}\,\widetilde{Э}_2\,\alpha_{j_2} \ldots \ldots \ldots \alpha_{i_s-1}\,\widetilde{Э}_s\,\alpha_{j_s} \ldots \alpha_n\,b$$

и соответственно почти нормальный вид формулы

$$Ш_2 = a\,\alpha_1 \ldots \alpha_{i_1-1}\,(\widetilde{Ю}_1)\,\alpha_{j_1} \ldots \alpha_{i_2-1}\,(\widetilde{Ю}_2)\,\alpha_{j_2} \ldots \ldots \ldots \alpha_{i_3-1}\,(\widetilde{Ю}_3)\,\alpha_{j_3} \ldots \alpha_n\,b,$$

где $Ю_I$, $Ю_2$, ... , $Ю_s$ – аналоги правой части Ю правила {Э,Ю}, соответствующие аналогам $Э_I$, $Э_2$, ... , $Э_s$ левой части Э.

ОПРЕДЕЛЕНИЕ 25. Для любой формулы Ш и любого правила {Э, Ю} однозначно определена последовательность формул Ш, Ш′, Ш″, ... , $Ш^{(m)}$, ... , каждая из которых (кроме первой) есть результат однократной (соответственно строчной) обобщенной подстановки по правилу {Э, Ю} в предыдущую формулу. Если в этой последовательности есть формула, в которую обобщенная подстановка по правилу {Э, Ю} невозможна, то почти нормальный вид первой такой формулы объявляется результатом полной (соответственно циклической) обобщенной подстановки в формулу Ш по данному правилу. Если же такой формулы в последовательности Ш, Ш′, Ш″, ... , $Ш^{(m)}$, ... нет, то мы говорим, что полная (соответственно циклическая) обобщенная подстановка зацикливает; ее результат в этом случае неопределен.

Если левая часть Э правила {Э, Ю} не содержит переменных букв, то любой аналог формулы Э совпадает с самой формулой Э, а соответствующий аналог формулы Ю совпадает с формулой Ю. Поэтому в данном случае операция обобщенной подстановки приведет к тому же результату, что и соответствующая операция простой подстановки. Значит, допуская вольность речи, можно считать, что простая подстановка есть частный случай обобщенной подстановки. В частности, все примеры простых подстановок, приведенных в предыдущем параграфе, могут служить примерами обобщенных

подстановок (при условии, что буквы в левых частях применяемых правил все постоянны), а выводы, сделанные на основе этих примеров, имеют отношение и для операции обобщенной подстановки.

Читателю повидимому не представит затруднений применение обобщенных подстановок в ситуациях, описанных в начале параграфа. Ниже мы приводим еще несколько примеров. Как и прежде, символы Ш, Э, Ю, $Ш_I$, $Ш_2$, $Ш_3$ и $Ш_4$ означают исходную формулу, левую и правую части правила и результаты однократной, строчной, полной и циклической обобщенных подстановок.

ПРИМЕР I. Пусть

Э = $\Delta\ \xi\ \Delta$, Ю = $\Delta\ \xi\ \dagger\ \xi\ \Delta$ и Ш = $\Delta\ (a \times a + a) \times a \times 2 + 5 \Delta$.

Если $\xi \in \Phi_2$, то

$Ш_I = \Delta\ (a \times a + a) \times a \times 2 \dagger 2 + 5 \Delta$;

$Ш_2 = \Delta\ (a \times a + a) \times a \times 2 \dagger 2 + 5 \dagger 5 \Delta$.

Если $\xi \in \Phi_I$, то

$Ш_I = \Delta\ (a \dagger a \times a + a) \times a \times 2 + 5 \Delta$;

$Ш_2 = \Delta\ (a \dagger a \times a \dagger a + a \dagger a) \times a \dagger a \times 2 \dagger 2 + 5 \dagger 5 \Delta$;

если же $\xi \in \Phi_3$ или $\xi \in \Phi_0$, то

$Ш_I = \Delta\ (a \times a + a) \dagger (a \times a + a) \times a \times 2 + 5 \Delta$;

$Ш_2 = \Delta\ (a \times a + a) \dagger (a \times a + a) \times a \dagger a \times 2 \dagger 2 + 5 \dagger 5 \Delta$.

Во всех этих случаях полная и циклическая обобщенные подстановки зацикливают.

ПРИМЕР 2. Пусть Э = $\Delta\ \xi \times 0\ \Delta$, Ю = $\Delta\ 0\ \Delta$ и

Ш = $\Delta\ (a \times b \times 0 + 1) \times a \dagger b \times 0 + 5 \times 6 \times 0 + 7 \times 0 \Delta$.

Если $\xi \in \Phi_2$, то

$Ш_I = \Delta\ (a \times b \times 0 + 1) \times a \dagger b \times 0 + 5 \times 0 + 7 \times 0 \Delta$;

$Ш_2 = \Delta\ (a \times b \times 0 + 1) \times a \dagger b \times 0 + 5 \times 0 + 0 \Delta$;

$Ш_3 = Ш_4 = \Delta\ (a \times b \times 0 + 1) \times a \dagger b \times 0 + 0 + 0 \Delta$;

если $\xi \in \Phi_I$, то

$$Ш_I = \Delta(a*0+1)*a+b*0+5*6*0+7*0\Delta,$$

$$Ш_2 = \Delta(a*0+1)*a+b*0+5*0+0\Delta;$$

$$Ш_3 = Ш_4 = \Delta(0+1)*a+b*0+0+0\Delta,$$

если $\xi \in \Phi_3$, то формула $Ш_I$ та же, что и в случае $\xi \in \Phi_I$,

$$Ш_2 = \Delta(a*0+1)*0+5*0+0\Delta \quad , \quad Ш_3 = Ш_4 = \Delta 0+0+0\Delta,$$

если, наконец, $\xi \in \Phi_0$, то

$$Ш_I = \Delta 0+5*6*0+7*0\Delta \quad , \quad Ш_2 = Ш_3 = Ш_4 = \Delta 0+0+0\Delta.$$

ПРИМЕР 3. Пусть

$$\{Э, Ю\} = \{\Delta \xi + \theta\Delta \xi * \xi + (\theta+\bar{1})\Delta, \qquad (32)$$

$$Ш = \Delta a+2*b+2+c+3+(a+b)+2*(a+b)+3*(a+c+b+6)+3\Delta.$$

Если $\xi \in \Phi_I$ и $\theta \in \Phi_2$, то

$$Ш_1 = \Delta a*a+(2+\bar{1})*b+2+c+3+(a+b)+2*(a+b)+3*(a+c+b+6)+3\Delta;$$

$$Ш_2 = \Delta a*a+(2+\bar{1})*b*b+(2+\bar{1})+c*c+(3+\bar{1})+(a+b)+2*(a*b)+3*(a+c+b*b+(6+\bar{1})+3\Delta,$$

если же $\xi \in \Phi_3 \cup \Phi_0$ и $\theta \in \Phi_2$, то формула $Ш_I$ та же самая, а $Ш_2$ имеет вид

$$Ш_2 = \Delta a*a+(2+\bar{1})*b*b+(2+\bar{1})+c*c+(3+\bar{1})+(a+b)+(2+\bar{1})*a*b*(a+b)+(3+\bar{1})*(a+c+b+6)*(a+c+b+6)+(3+\bar{1})\Delta.$$

Преобразование той же формулы Ш по несколько иному правилу

$$\{Э, Ю\} = \{\Delta(\xi)+\theta\Delta \xi*\xi+(\theta+\bar{1})\Delta\} \qquad (33)$$

где $\xi \in \Phi_3 \cup \Phi_0$ и $\theta \in \Phi_2$, приводит к формулам

$$Ш_1 = \Delta a+2*b+2+c+3+(a+b)*(a+b)+(2+\bar{1})*(a+b)+3*(a+c+b+6)+3\Delta,$$

$$Ш_2 = \Delta a+2*b+2+c+3+(a+b)*(a+b)+(2+\bar{1})*a*b*(a+b)+3*(a+c+b+c)*(a+c+b+6)+(3+\bar{1})\Delta.$$

Во всех перечисленных случаях полная и циклическая подстановки зацикливают.

Правила (32) и (33) очевидно рассчитаны на "раскручивание" степеней с натуральными показателями в виде соответствующих произведений. Однако непосредственное циклическое применение этих правил приводит к зацикливанию. Для его предупреждения после каждого применения правила (32) или (33) нужно в результате каждую подформулу вида $\Delta(\theta+\bar{1})\Delta$, где θ – конкретное число, заменить равноценной однобуквенной подформулой (так, подформула $\Delta(2+\bar{1})\Delta$ должна быть заменена подформулой $\Delta I \Delta$) и затем преобразовать его по правилу $\{\Delta\xi+1\Delta\xi\Delta\}$. Упрощения подобного типа в формулах производит специальный алгоритм, реализованный в виде отдельного стандартного оператора "Арифметик". Подробнее описание этого оператора будет дано в главе IУ. В текущей же главе мы будем лишь предполагать, что в формуле, обработанной Арифметиком, нет ни одного аналога ни одной из следующих формул

$$\Delta\xi+0\Delta\;0+\xi\Delta\;\xi*0\Delta\;0*\xi\Delta\;\xi*1\Delta\;1*\xi\Delta\;\xi+0\Delta\;0+\xi\Delta\;\xi+1\Delta$$

$$1+\xi\Delta\;,\quad\Delta(\theta)\Delta\;\theta*\tau\Delta\;\theta*\tau\Delta\;\theta+\tau\Delta,$$

где $\xi\in\Phi_3$ и $\theta,\tau\in\Phi_2$.

ПРИМЕР 4. В таблице I приведены многочисленные варианты работы правила $\{Э, Ю\}$, левая часть Э которого указана в крайнем левом столбце, а правая часть Ю совпадает с формулой $\Delta\xi*(\eta+1)\Delta$. Во всех отраженных в таблице случаях буква ζ имеет тип Φ_I. Символами + и $\oplus$ отмечена ситуация, когда обобщенная подстановка невозможна. Результаты подстановки для этого случая указаны в последней строке. Читателю рекомендуется скопировать всю таблицу кроме знаков +, $\oplus$ и – и расставить затем все эти знаки самостоятельно. Анализ возможных ошибок будет весьма полезен.

Отметим еще, что в случае $\zeta\in\Phi_3\cup\Phi_0$ таблица была бы почти той же (с тремя исключениями). Если во всех отмеченных в таблице вариантах считать ξ и $\zeta\in\Phi_I$, $\xi\in\Phi_3\cup\Phi_0$, то обобщенная подстановка всюду (за исключением двух случаев) окажется невозможной.

Левая часть Э применяемого правила	Исходная формула Ш		$\Delta a+b+(a+b)\times c\Delta$	$\Delta(a+b)+(a+b)\times c\Delta$	$\Delta a/b+(a/b)\times c\Delta$	$\Delta(a/b)+(a/b)\times c\Delta$	$\Delta a\times b+a\times b\times c\Delta$	$\Delta a\times b+(a\times b)\times c\Delta$	$\Delta(a\times b)+a\times b\times c\Delta$	$\Delta(a\times b)+(a\times b)\times c\Delta$	$\Delta a\uparrow b+a\uparrow b\times c\Delta$	$\Delta a\uparrow b+(a\uparrow b)\times c\Delta$	$\Delta(a\uparrow b)+a\uparrow b\times c\Delta$	$\Delta(a\uparrow b)+(a\uparrow b)\times c\Delta$
$\Delta\xi+\xi\times\Gamma\Delta$	$\xi\in\Phi_3$		−	+	+	+	−	−	−	+	+	−	−	+
	$\xi\in\Phi_0$		+	+	+	+	+	−	−	+	+	−	−	+
$\Delta(\xi)+\xi\times\Gamma\Delta$	$\xi\in\Phi_3$		−	+	−	+	−	−	−	−	−	−	+	−
	$\xi\in\Phi_0$		−	+	−	+	−	−	+	−	−	−	+	−
$\Delta\xi+(\xi)\times\Gamma\Delta$	$\xi\in\Phi_3$		−	−	+	−	−	+	−	−	−	+	−	−
	$\xi\in\Phi_0$		+	−	+	−	−	+	−	−	−	+	−	−
$\Delta(\xi)+(\xi)\times\Gamma\Delta$	$\xi\in\Phi_3\cup\Phi_0$		−	+	−	+	−	−	−	+	−	−	−	+
$\Delta\xi+\zeta\times\Gamma\Delta$	$\xi,\zeta\in\Phi_3$	⊕	+	+	+	+	−	+	−	+	+	+	+	+
	$\xi,\zeta\in\Phi_0$		+	+	+	+	+	+	+	+	+	+	+	+
$\Delta(\xi)+\zeta\times\Gamma\Delta$	$\xi,\zeta\in\Phi_3$		−	+	−	+	−	−	−	+	−	−	+	+
	$\xi,\zeta\in\Phi_0$		−	+	−	+	−	−	+	+	−	−	+	+
$\Delta\xi+(\zeta)\times\Gamma\Delta$	$\xi,\zeta\in\Phi_3$	⊕	+	+	+	+	−	+	−	+	−	+	−	+
	$\xi,\zeta\in\Phi_0$		+	+	+	+	−	+	−	+	−	+	−	+
$\Delta(\xi)+(\zeta)\times\Gamma\Delta$	$\xi,\in\Phi_3\cup\Phi_0$		−	+	−	+	−	−	−	+	−	−	−	+
Результаты: $Ш_1=Ш_2=Ш_3=Ш_4$ в случае +, когда обобщенная подстановка возможна			$\Delta(a+b)\times(c+1)\Delta$		$\Delta(a/b)\times(c+1)\Delta$		$\Delta a\times b\times(c+1)\Delta$				$\Delta a\uparrow b\times(c+1)\Delta$			
Результаты $Ш_1=Ш_2=Ш_3=Ш_4$ в случае −, когда обобщенная подстановка невозможна			$\Delta a+b+(a+b)\times c\Delta$		$\Delta a/b+(a/b)\times c\Delta$		$\Delta a\times b+a\times b\times c\Delta$				$\Delta a\uparrow b+a\uparrow b\times c\Delta$			

Таблица I

ПРИМЕР 5. Правило

$$\{\Delta(\xi\times\eta)\uparrow 5\,\Delta\,\xi\uparrow\eta\times\eta\uparrow 5\,\Delta\},\quad \xi,\eta\ \text{и}\ 5\in\Phi_0,$$

соответствует обычному равенству

$$(\xi\eta)^5=\xi^5\eta^5$$

и позволяет раскрывать степени произведений. Например, применяя это правило к формуле

$$\text{Ш}=\Delta(a\times b\times c)\uparrow 2\times((a\times b)\uparrow c\times a)\uparrow((a\times b)\uparrow a\,\Delta,$$

мы получим

$$\text{Ш}_1=\Delta\,a\uparrow 2\times(b\times c)\uparrow 2\times((a\times b)\uparrow c\times a)\uparrow((a\times b)\uparrow a)\Delta,$$

$$\text{Ш}_2=\Delta\,a\uparrow 2\times(b\times c)\uparrow 2\times((a\times b)\uparrow c)\uparrow((a\times b)\uparrow a)\times a\uparrow((a\times b)\uparrow a)\Delta,$$

$$\text{Ш}_3=\text{Ш}_4=\Delta\,a\uparrow 2\times b\uparrow 2\times c\uparrow 2\times(a\uparrow c)\uparrow(a\uparrow a\times b\uparrow a)\times(b\uparrow c)\uparrow$$
$$(a\uparrow a\times b\uparrow a)\times a\uparrow(a\uparrow a\times b\uparrow a)\Delta.$$

ПРИМЕР 6. Правило

$$\{\text{Э},\text{Ю}\}=\{\Delta\,\xi\,\log_\eta\zeta\,\Delta\,\ln\times\zeta\ /\ \ln\times\xi\,\Delta\},\quad \xi\ \text{и}\ \eta\in\Phi_3\cup\Phi_0,$$

соответствует обычному равенству

$$\log_\xi\zeta=\ln\zeta/\ln\xi$$

и позволяет заменять все логарифмы одним натуральным. Преобразование по этому правилу формулы

$$\text{Ш}=\Delta(b\log_a a)\log_{(a\log_a b)\log_a c}\times 2\log_{(a\times\ln\times b)}\Delta,$$

соответствующей выражению

$$\log_{\log_b a}\left(\log_{\log_a b} c\right)\times\log_2(a\ln b),$$

приводит к следующим результатам:

$$\text{Ш}_1=\Delta\ln\times((a\log_a b)\log_a c)/\ln\times(b\log_a a)\times 2\log_{(a\times\ln\times b)}\Delta,$$

$$\text{Ш}_2=\Delta\ln\times((a\log_a b)\log_a c)/\ln\times(b\log_a a)\times\ln\times(a\times\ln\times b)/\ln\times 2\Delta,$$

$$\text{Ш}_3=\text{Ш}_4=\Delta\ln\times(\ln\times c/\ln\times(\ln\times b/\ln\times a))/\ln\times(\ln\times a/$$
$$\ln\times b)\times\ln\times(a\times\ln\times b)/\ln\times 2\Delta:$$

ПРИМЕР 7. Поочередное применение правил

$$\{\Delta\, ln * (\xi \times \eta)\, \Delta\, ln * \xi + ln * \eta\, \Delta\},$$
$$\{\Delta\, ln * (\xi / \eta)\, \Delta\, ln * \xi + ln * \eta \times \bar{1}\, \Delta\},$$
$$\{\Delta\, ln * (\xi \uparrow \eta)\, \Delta\, ln * \xi \times \eta\}, \qquad (34)$$

где ξ и $\eta \in \Phi_0$, позволяет расписывать логарифмы произведений, дробей и степеней через логарифмы "элементарных сомножителей". Например, осуществляя циклическую (или полную) обобщенную подстановку в формулу

$$Ш = \Delta\, ln * (a/(a+b) \uparrow 2 \times b/(c+d) \uparrow 3 \times c \uparrow 2 \times ln * (b \uparrow (a \times 2)) \uparrow 3)\, \Delta,$$

сначала по первому, затем по второму, затем по третьему и, наконец, снова по первому из правил (34), мы последовательно получим формулы

$$Ш' = \Delta\, ln * (a/(a+b) \uparrow 2) + ln * (b/(c+d) \uparrow 3) + ln * (c \uparrow 2) + ln * (ln * (b \uparrow (a \times 2)) \uparrow 3)\, \Delta,$$

$$Ш'' = \Delta\, ln * a + ln * ((a+b) \uparrow 2) \times \bar{1} + ln * b + ln * ((c+d) \uparrow 3) \times \bar{1} + ln * (c \uparrow 2) + ln * (ln * (b \uparrow (a \times 2)) \uparrow 3)\, \Delta,$$

$$Ш''' = \Delta\, ln * a + ln * (a+b) \times 2 \times \bar{1} + ln * b + ln * (c+d) \times 3 \times \bar{1} + ln * c \times 2 + ln * (ln * b \times a \times 2) \times 3\, \Delta,$$

$$Ш^{IV} = \Delta\, ln * a + ln * (a+b) \times 2 \times \bar{1} + ln * b + ln * (c+d) \times 3 \times \bar{1} + ln * c \times 2 + (ln * ln * b + ln * a + ln * 2) \times 3\, \Delta$$

ПРИМЕР 8. Правила

$$\{\Delta\, cos * \xi\, \Delta(e \uparrow (\xi \times i) + e \uparrow (\xi \times i \times \bar{1})) \times \tfrac{1}{2}\, \Delta\},$$
$$\{\Delta\, sin * \xi\, \Delta(e \uparrow (\xi \times i) + e \uparrow (\xi \times i \times \bar{1}) \times \bar{1}) \times \tfrac{1}{2} \times i\, \Delta\},$$
$$\{\Delta\, ch * \xi\, \Delta(e \uparrow \xi + e \uparrow (\xi \times \bar{1})) \times \tfrac{1}{2}\, \Delta\},$$
$$\{\Delta\, sh * \xi\, \Delta(e \uparrow \xi + e \uparrow (\xi \times \bar{1}) \times \bar{1}) \times \tfrac{1}{2}\, \Delta\},$$

где $\xi \in \Phi_0$, соответствуют обычным формулам Эйлера. Их применение позволяет в необходимых случаях заменять тригонометрические и гиперболические функции соответствующими комбинация-

ми экспоненты. Например, окончательный результат последовательной циклической (или полной) обобщенной подстановки в формулу

$$Ш = \Delta t \int sh*t \times cos*t \times 2 = ch*t \times cos*t + sh*t \times sin*t \Delta$$

по данным правилам имеет вид

$$Ш = \Delta t \int (e*t + e*(t*\bar{I}) \times \bar{I}) \times \tfrac{1}{2} \times (e*(t \times i) + e*(t \times i \times \bar{I})) \times$$
$$\tfrac{1}{2} \times 2 = (e*t + e*(t \times \bar{I})) \times \tfrac{1}{2} \times (e*(t \times i) + e*(t \times i \times \bar{I})) \times \tfrac{1}{2} +$$
$$(e*t + e*(t \times \bar{I}) \times \bar{I}) \times \tfrac{1}{2} \times (e*(t \times i) + e*(t \times i \times \bar{I}) \times \bar{I}) \times$$
$$\tfrac{\bar{I}}{2} \times i \Delta.$$

Многочисленные дальнейшие примеры правил и обобщенных подстановок по ним имеются в следующем параграфе

§ II. ПРИМЕНЕНИЯ ОБОБЩЕННОЙ ПОДСТАНОВКИ

В данном параграфе приводятся алгоритмы для решения некоторых задач, использующие обобщенную подстановку. Почти все эти алгоритмы можно переработать так, что они смогут решать тот же или даже более широкий класс задач с меньшими затратами времени (в первом приближении можно считать, что время решения задачи пропорционально количеству обращений к оператору обобщенной подстановки). Однако основной целью мы здесь считаем иллюстрацию возможностей операции обобщенной подстановки. Поэтому вопросы эффективности предлагаемых алгоритмов и пути их усовершенствования здесь не рассматриваются.

ПРИМЕР I. ПРИВЕДЕНИЕ ПОДОБНЫХ ЧЛЕНОВ. Задача приведения подобных членов известна каждому и тем не менее нуждается в уточнении, поскольку разные люди толкуют ее по разному. Одни понимают под приведением подобных членов только группировку слагаемых, отличающихся лишь числовым сомножителем. В этом смысле формула

$$\Delta \; a*2 + a \times a \; \Delta$$

подобных членов не имеет. Другие, наоборот, отождествляют эту операцию с полным раскрытием всех скобок по закону дистрибутивности умножения относительно сложения и последующей группи-

ровкой подобных слагаемых, причем требуют чтобы такие слагаемые, как

$$\Delta\, a\, \Delta \quad \text{и} \quad \Delta\, a \times b / b\, \Delta,$$

$$\Delta\, a \uparrow 2\, \Delta \quad \text{и} \quad \Delta\, a \times a\, \Delta,$$

$$\Delta\, \cos * (a \times b)\, \Delta \quad \text{и} \quad \Delta\, \cos * (b \times a)\, \Delta,$$

$$\Delta\, a \uparrow 2 / b \uparrow 2\, \Delta \quad \text{и} \quad \Delta\, (a/b) \uparrow 2\, \Delta,$$

$$\Delta\, a \uparrow 2 / b\, \Delta \quad \text{и} \quad \Delta\, a \times b \uparrow \bar{1} / (a \sqrt{2}) \uparrow \tfrac{1}{2}\, \Delta$$

и т.д. считались подобными. Повидимому вопрос о точной постановке задачи приведения подобных членов не тривиален и заслуживает специального исследования с учетом нужд практики и возможности создания эффективного алгоритма ее решения.

Нас эта задача в данный момент интересует лишь постольку, поскольку она дает богатые возможности иллюстрации обобщенной подстановки. Поэтому мы можем ограничиться следующим: Объявим два слагаемых в сумме подобными, если они после конечного числа преобразований по обычным "школьным" правилам действия с дробями и степенями дробей и произведений (т.е. по правилам типа

$$a \cdot a = a^2, \quad a \cdot a^n = a^{n+1}, \quad (ab)^n = a^n b^n, \quad a/b = a b^{-1}, \quad (a^{-1})^m = a^{-m}$$

и т.п.) совпадут с точностью до числового сомножителя. Задача приведения подобных заключается в группировке подобных слагаемых.

Для решения поставленной задачи мы можем предложить следующий алгоритм:

АЛГОРИТМ А. I) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\, \xi / \eta\, \Delta\, \xi \times \eta \uparrow \bar{1}\, \Delta\}, \quad \xi \text{ и } \eta \in \Phi_3.$$

2) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\, (\xi \times \varphi) \uparrow \eta\, \Delta\, \xi \uparrow \eta \times \varphi \uparrow \eta\, \Delta\}, \quad \xi, \eta \in \Phi_3, \quad \varphi \in \Phi_0.$$

3) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\, (\xi \uparrow \eta) \uparrow \zeta\, \Delta\, \xi \uparrow (\eta \times \zeta)\, \Delta\}, \quad \xi, \eta, \zeta \in \Phi_3.$$

4) Коммутирование формулы Ш (в смысле определения I8).

5) Преобразование формулы Ш оператором Арифметик. Если в результате этого формула Ш изменилась, то переход на блок № 4.

6) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \xi\uparrow\eta\times\xi\uparrow\zeta\ \Delta\ \xi\uparrow(\eta+\zeta)\ \Delta\},\quad \xi,\eta,\zeta\in\Phi_3. \qquad (35)$$

7) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \xi\times\xi\ \Delta\ \xi\uparrow 2\ \Delta\},\quad \xi\in\Phi_3$$

Если в результате этого формула Ш не изменилась, то переход на блок № I0.

8) Коммутирование формулы Ш.

9) Строчная обобщенная подстановка в формулу Ш по правилу (35).

I0) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \xi\times\varphi\times\xi\uparrow\eta\ \Delta\ \varphi\times\xi\uparrow(\eta+1)\ \Delta\},\quad \xi,\eta\in\Phi_3,\ \varphi\in\Phi_0.$$

II) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \xi\times\xi\uparrow\eta\ \Delta\ \xi\uparrow(\eta+1)\ \Delta\},\quad \xi,\eta\in\Phi_3.$$

I2) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ (\xi)\uparrow\eta\times\varphi\times\xi\ \Delta\ \xi\uparrow(\eta+1)\times\varphi\ \Delta\},\quad \xi,\eta\in\Phi_3,\ \varphi\in\Phi_0.$$

I3) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ (\xi)\uparrow\eta\times\xi\ \Delta\ \xi\uparrow(\eta+1)\ \Delta\},\quad \xi,\eta\in\Phi_3.$$

I4) Преобразование формулы Ш оператором Арифметик.

I5) Коммутирование формулы Ш.

I6) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \varphi\times\theta+\varphi\times\tau\ \Delta\ \varphi\times(\theta+\tau)\ \Delta\},\quad \theta,\tau\in\Phi_2,\ \varphi\in\Phi_0.$$

Если в результате этого формула Ш не изменилась, то переход на блок № I8.

I7) Преобразование формулы Ш оператором Арифметик и безус-

ловный переход на блок № I6.

I8) Коммутирование формулы Ш.

I9) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \xi + \xi\ \Delta\ \xi \times 2\ \Delta\},\ \xi \in \Phi_3.$$

Если в результате этого формула Ш изменилась, то переход на блок № I6.

20) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \varphi + \psi + \varphi \times \theta\ \Delta\ \psi + \varphi \times (\theta + 1)\ \Delta,\ \theta \in \Phi_2,\ \varphi, \psi \in \Phi_0. \quad (36)$$

2I) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ \varphi + \varphi \times \theta\ \Delta\ \varphi \times (\theta + 1)\ \Delta\},\ \theta \in \Phi_2,\ \varphi \in \Phi_0.$$

22) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ (\xi) \times \theta + \varphi + \xi\ \Delta\ \xi \times (\theta + 1) + \varphi\ \Delta\},\ \theta \in \Phi_2,\ \xi \in \Phi_3,\ \varphi \in \Phi_0.$$

23) Строчная обобщенная подстановка в формулу Ш по правилу

$$\{\Delta\ (\xi) \times \theta + \xi\ \Delta\ \xi \times (\theta + 1)\ \Delta\},\ \theta \in \Phi_2,\ \xi \in \Phi_3.$$

24) Преобразование формулы Ш оператором Арифметик.

25) Коммутирование формулы Ш. Конец.

В этом алгоритме исходная формула обозначена символом Ш. Тем же символом обозначается результат работы каждого блока, преобразующего формулу Ш, и,в частности, окончательный результат работы алгоритма. Первые пять блоков освобождают исходную формулу от дробей и от степеней, основания которых суть произведения или степени. Результат приводится к нормальному виду и обрабатывается оператором Арифметик.

Например, формулы

$$\Delta\ (a \times 2) \uparrow 2 \times a + a \uparrow 3 \times \overline{4}\ \Delta\ t \uparrow \overline{I} \times \overline{3} + tg \times t + 1/t + 2/t\ \Delta,$$

$$\Delta\ (a \times c) \uparrow 2 \times a \times b \times c + (a \times d)/c \times a \times d + a \uparrow 3 \times b \times c \uparrow 3\ \Delta,$$

$$\Delta\ (a/a \uparrow b) \uparrow 2 / (a \uparrow (b + 1) \uparrow 2 \times ((a \uparrow 2) \uparrow b) \uparrow 2\ \Delta,$$

$$\Delta\ (((a/b)/(c/d)) \uparrow a) \uparrow 2 / ((a/c) \uparrow 2 / (b \uparrow 3/d)) \uparrow a\ \Delta,$$

$$\Delta\ ((t+t\uparrow 2+t\uparrow 2\times\bar{1})\times t\int t)\times 2+a\times a/b\times a\uparrow 2+(t\uparrow 2+t\times t+t\times t\times\bar{1})\int t\,\Delta,$$

$$\Delta\ (a\times((a/(a\times b))\uparrow a\times b/a)/(((a\times c)/(a\times b))\uparrow b\times 1/c\uparrow(2\times b))\times a\uparrow a/c\uparrow a)/(a\uparrow a/((c\uparrow(a/c))/a)\uparrow a\times b/(c\times a\uparrow a/(c\uparrow b)\uparrow 2)\times(b\uparrow c)\uparrow b/(c\times a\times((a\times b)/c)\uparrow a))\,\Delta,$$

соответствующие выражениям

$$(2a)^2a-4a^3,-3t^{-1}+tgt+\frac{1}{t}+\frac{2}{t},(ac)^2abc+\frac{ad}{c}ad+a^3bc^3,$$

$$\left\{\left(\frac{a}{a^b}\right)^2:(a^{b+1})^2\right\}\left[(a^2)^b\right]^2,\left[\left(\frac{a}{b}:\frac{c}{d}\right)^a\right]^2:\left[\left(\frac{a}{c}\right)^2:\frac{b^3}{d}\right]^a,$$

$$2\int(t+t^2-t^2)\,t\,dt+a\frac{a}{b}a^2+\int(t^2+tt-tt)\,dt,$$

$$\left\{a\frac{\left(\frac{a}{ab}\right)^a\cdot\frac{b}{a}}{\left(\frac{ac}{ab}\right)^b\cdot\frac{1}{c^{2b}}}\cdot\frac{a^a}{c^a}\right\}:\left\{\frac{a^2}{\left(\frac{c:\frac{a}{c}}{a}\right)^a}\cdot\frac{b}{c\frac{a^a}{(c^b)^2}}\cdot\frac{\left(\frac{b}{c}\right)^b}{ca\left(\frac{ab}{c}\right)^a}\right\},$$

перерабатываются блоками I ÷ 5 соответственно в формулы

$$\Delta\ a\times a\uparrow 2\times 4+a\uparrow 3\times\bar{4}\ \Delta\ t\uparrow\bar{1}+t\uparrow\bar{1}\times\bar{3}+t\uparrow\bar{1}\times 2+tg\times t\,\Delta,$$

$$\Delta\ a\times a\times c\uparrow\bar{1}=d\times d+a\times a\uparrow 2\times b\times c\times c\uparrow 2+a\uparrow 3\times b\times c\uparrow 3\,\Delta,$$

$$\Delta\ a\uparrow((b+1)\times\bar{2})\times a\uparrow(b\times\bar{2})\times a\uparrow(b\times 4)\times a\uparrow 2\,\Delta,$$

$$\Delta\ a\uparrow(a\times\bar{2})\times a\uparrow(a\times 2)\times b\uparrow(a\times\bar{2})\times b\uparrow(a\times 3)\times c\uparrow(a\times\bar{2})\times c\uparrow(a\times 2)\times d\uparrow(a\times\bar{1})\times d\uparrow(a\times 2)\,\Delta, \qquad (37)$$

$$\Delta((t+t\uparrow 2+t\uparrow 2\times\bar{1})\times t\int t)\times 2+(t\uparrow 2+t\times t+t\times t\times\bar{1})\int t+a\times a\times a\uparrow 2\times b\uparrow\bar{1}\,\Delta,$$

$$\Delta\ a\times a\times a\uparrow(a\times\bar{1})\times a\uparrow(a\times\bar{1})\times a\uparrow(a\times\bar{1})\times a\uparrow(a\times\bar{1})\times a\uparrow(b\times\bar{1})\times a\uparrow a\times a\uparrow a\times a\uparrow a\times a\uparrow a\times a\uparrow b\times a\uparrow\bar{1}\times b\times b\uparrow(a\times\bar{1})\times b\uparrow(b\times\bar{1})\times b\uparrow a\times b$$

$$\uparrow b \times b \uparrow \bar{1} \times c \times c \times c \uparrow (a \times \bar{1}) \times c \uparrow (a \times \bar{1}) \times c \uparrow (b \times \bar{2}) \times c \uparrow (b \times \bar{1}) \times c \uparrow$$
$$(b \times 2) \times c \uparrow a \times c \uparrow a \times c \uparrow b \Delta.$$

Следующие блоки (начиная с 6-го и кончая I5-тым) алгоритма группируют сомножители в подформулах формулы Ш, имеющих главную связь $\times$, по правилам умножения степеней с одинаковыми основаниями (т.е. по правилам $a^b a^c = a^{b+c}$ и $aa^b = a^{b+1}$). Учитываются все возможные взаимные положения таких сомножителей в нормальной формуле. Результат группировки обрабатывается операторами Коммутирование и Арифметик.

Например, формула (37) и формулы

$$\Delta a \times a \times a \times a \uparrow 2 \times b \times b + a \times b + a \times b \times 2 + a \uparrow 2 \times a \uparrow 3 \times b \uparrow 2 \Delta,$$
$$\Delta ((\int\int t) \log 2) \uparrow 2 \times (\int\int t) \times (\int\int t) \times (\int\int t) \times (\int\int t) \log 2 \times (\int\int t) \uparrow \bar{2} \Delta$$

блоками 6 ÷ I5 преобразуются соответственно в формулы

$$\Delta a \uparrow 3 \times \bar{4} + a \uparrow 3 \times 4 \Delta t \uparrow \bar{1} + t \uparrow \bar{1} \times \bar{3} + t \uparrow \bar{1} \times 2 + t g \times t \Delta,$$
$$\Delta a \uparrow 2 \times c \uparrow \bar{1} \times d \uparrow 2 + a \uparrow 3 \times b \times c \uparrow 3 + a \uparrow 3 \times b \times c \uparrow 3 \Delta,$$
$$\Delta a \uparrow ((b+1) \times \bar{2} + b \times \bar{2} + b \times 4 + 2) \Delta,$$
$$\Delta a \uparrow (a \times \bar{2} + a \times 2) \times b \uparrow (a \times \bar{2} + a \times 3) \times c \uparrow (a \times \bar{2} + a \times 2) \times d \uparrow (a \times \bar{1} + a \times 2) \Delta, \quad (38)$$
$$\Delta ((t + t \uparrow 2 + t \uparrow 2 \times \bar{1}) \times t \int t) \times 2 + (t \uparrow 2 + t \uparrow 2 + t \uparrow 2 \times \bar{1}) \int t + a \uparrow 4 \times b \uparrow \bar{1} \Delta,$$
$$\Delta a \uparrow (a + a + a + a + a \times \bar{1} + a \times \bar{1} + a \times \bar{1} + a \times \bar{1} + b + b \times \bar{1} + 1) \times b \uparrow (a +$$
$$a \times \bar{1} + b + b \times \bar{1}) \times c \uparrow (a + a + a \times \bar{1} + a \times \bar{1} + b + b \times \bar{2} + b \times \bar{1} + b \times 2 + 2 \Delta,$$
$$\Delta a \times b + a \times b \times 2 + a \uparrow 5 \times b \uparrow 2 + a \uparrow 5 \times b \uparrow 2 \Delta ((\int\int t$$
$$) \log 2) \uparrow 3 \times (\int\int t) \Delta.$$

Последние блоки алгоритма (начиная с I6-го) группируют подобные слагаемые в подформулах формулы Ш, имеющих главную связь +. Подобными слагаемыми в этот момент считаются лишь те,

которые отличны друг от друга разве лишь на числовой сомножитель. Учитываются все возможные взаимные положения таких слагаемых. Окончательный результат работы алгоритма очищается от подформул, допускающих чисто арифметическое упрощение, и приводится к нормальному виду.

В частности, формулы (38) заключительными блоками перерабатываются соответственно в формулы

$$\Delta 0 \Delta t_9 \times t \Delta a_{12} \times c_{\dagger} \bar{1} \times d_{12} + a_{13} \times b \times c_{13} \times 2 \Delta a_{\dagger} ((b+1) \times 2 +$$

$$b \times 2 + 2) \Delta b_{\dagger} a \times d_{\dagger} a \Delta (t \times t \int t) \times 2 + a_{14} \times b_{\dagger} \bar{1} + t_{\dagger} 2 \int t \Delta,$$

$$\Delta a_{\dagger} c_{\dagger} 2 \Delta a \times b_{13} + a_{15} \times b_{12} \times 2 \Delta ((t \int f) \log 2) 13 \times (t \int f) \Delta.$$

В этих окончательных результатах лишь две формулы

$$\Delta a_{\dagger} ((b+1) \times 2 + b \times 2 + 2) \Delta (t \int t \times t) \times 2 + a_{14} \times b_{\dagger} \bar{1} + t \int t_{\dagger} 2 \Delta$$

допускают дальнейшие упрощения. При этом одна из них не имеет подобных членов и требует применения закона дистрибутивности - операции, запрещенной в нашей версии приведения подобных. (см. по этому поводу следующий пример). Во второй формуле подобные члены приведены не полностью. Для завершения процесса их приведения достаточно еще раз пропустить эту формулу через весь алгоритм 3.

Итак, для некоторых формул наш алгоритм решает поставленную задачу при первом его применении. Легко видеть, что это справедливо, в частности, для всякой формулы вида

$$\mathbb{U} = \Delta \tilde{u}_{11} \times \tilde{u}_{12} \times \ldots \times \tilde{u}_{1m_1} + \tilde{u}_{21} \times \tilde{u}_{22} \times \ldots \times \tilde{u}_{2m_2} + \ldots$$

$$\ldots + \tilde{u}_{s1} \times \tilde{u}_{s2} \times \ldots \times \tilde{u}_{sm_s} \Delta,$$

где $s \geq 2$, $m_j \geq 1$ для любого $j = 1,2, \ldots, s$ и каждая подформула u_{jk}, $1 \leq j \leq s$, $1 \leq k \leq m_j$, такова, что

$$\mathbb{U}_{jk} = \Delta ((\ldots ((\tilde{u}_{jk} \dagger \rho_\ell) \dagger \rho_2) \dagger \ldots) \dagger \rho_\ell \Delta,$$

где

$$\ell = \ell(j,k) \geq 0$$

(при $\ell = 0$ формулы $\mathbb{U}_{jk}$ и $\mathbb{U}'_{jk}$ совпадают), $\rho_\nu = \rho_\nu(j,k)$ при любом $\nu = 1,2, \ldots, \ell$ есть вещественное число (точнее, элементарный символ, принадлежащий множеству R) и

подформула $Ш'_{jk}$ свободна от подобных членов и не является ни произведением, ни степенью, заключенными в скобки.

В общем случае для полного приведения всех подобных членов может потребоваться сколь угодно большое, но всегда конечное количество прокруток алгоритма A. Например, формула

$$Ш = \Delta((((((b+b)\times b+b\uparrow 2)\uparrow 2+b\uparrow 4\times\bar{5})\times b+b\uparrow 5)\times b+$$
$$b\uparrow b\times 2)\uparrow\tfrac{1}{2}+b\uparrow 3)\times 2+b\uparrow 3\times\bar{5})\Delta$$

только после семи прогонок через алгоритм A сворачивается до формулы $\Delta\, b\uparrow 3\,\Delta$.

Существенным недостатком описанного алгоритма приведения подобных членов является применение обобщенной подстановки по правилу (36). Оператор обобщенной подстановки (описание его см. ниже в главе IУ) по правилам такого типа организует весьма неприятный перебор обычно многочисленных аналогов формулы $\Delta\varphi+\psi\,\Delta$, φ и $\psi\in\Phi_0$, и затрачивает на это много машинного времени. В то же время приведение подобных в формулах типа

$$\Delta\, a+a\times b+a\times 2\,\Delta\; a\times b+a\times b\int c+a\times b\int d+a\times b\times 2\,\Delta\,.$$

может быть осуществлено обобщенной подстановкой только по правилу (36).

ПРИМЕР 2. РАСКРЫТИЕ СКОБОК И ПРИВЕДЕНИЕ ПОДОБНЫХ. Задача полного приведения подобных членов в формуле Ш с учетом закона дистрибутивности умножения относительно сложения решается при помощи следующего алгоритма:

АЛГОРИТМ Б. I) Копия формулы Ш обозначается через $\underline{Ш}$.

2) Формула Ш обрабатывается алгоритмом А. Результат снова обозначается через Ш.

3) В формуле Ш каждая подформула вида $\Delta(\tilde{Я})\uparrow n\,\Delta$, где n – натуральное число и подформула Я имеет главную связь +, заменяется подформулой

$$\Delta((\tilde{Я})\times\ldots\times(\tilde{Я}))\Delta,$$

в которой сомножитель (Я) повторяется n раз. Результат приводится к нормальному виду и обозначается через Ш.

4) Циклическая обобщенная подстановка в формулу Ш по правилу

$\{\Delta\varphi\times(\xi+\psi)\Delta\ \varphi\times\xi+\varphi\times\psi\Delta\}$, $\xi\in\Phi_3$, $\varphi,\psi\in\Phi_0$.

Результат обозначается через Ш.

5) Циклическая обобщенная подстановка в формулу Ш по правилу

$$\{\Delta(\xi+\varphi)\times\zeta\ \Delta\ \xi\times\zeta+\varphi\times\zeta\ \Delta\},\quad \xi,\zeta\in\Phi_3,\ \varphi\in\Phi_0.$$

Результат обозначается через Ш.

6) Если Ш $\neq$ Щ, то переход на блок № I. Иначе конец.

Например, формулу

$$\Delta\ ((a+b)\times(a+b\times I))\uparrow 3\ \Delta$$

алгоритм Б преобразует в формулу

$$\Delta\ a\uparrow 2\times b\uparrow 4\times\bar{3}+a\uparrow 4\times b\uparrow 2\times\bar{3}+a\uparrow 6+b\uparrow 6\times I\ \Delta$$

Читатель несомненно сумеет рассмотреть ряд других примеров преобразования формул по данному алгоритму.

Алгоритм Б дает как правило более существенное упрощение формулы по сравнению с предыдущим алгоритмом. Однако в некоторых важных для приложений случаях и этот алгоритм оказывается беспомощным. В частности, он не удовлетворяет естественному требованию: Алгоритм решения задачи "упростить выражение" должен выдавать результат $\Delta\ 0\ \Delta$ всякий раз, когда исходная формула чисто алгебраическим путем может быть преобразована до чистого нуля. Например, формулы

$$\Delta\ (a+b)\uparrow\bar{1}\times a+(a+b)\uparrow\bar{1}+b\ \Delta$$

$$\Delta\ (a+b\times 2+a\uparrow 2+b\uparrow 2)\uparrow\tfrac{1}{2}+a\times\bar{1}+b\times\bar{1}\ \Delta$$

алгебраически эквивалентны нулю, но наш алгоритм их не изменяет.

ПРИМЕР 3. ВЫЧИСЛЕНИЕ НЕОПРЕДЕЛЕННОГО ИНТЕГРАЛА ОТ ТРИГОНОМЕТРИЧЕСКОГО МНОГОЧЛЕНА.

Пусть задана функция $f(t)$ одного переменного t, представимая в виде

$$f(t)=P(z_1(t),z_2(t),\ldots,z_m(t)),\qquad (39)$$

где P – целый многочлен от нескольких переменных с произвольны-

ми не зависящими от t коэффициентами (или конечная суперпозиция таких многочленов) и каждое $z_i(t)$, $i = 1,2, \ldots, m$, есть либо t, либо $e^{\alpha t+\beta}$, либо $\sin(\alpha t+\beta)$, либо $\cos(\alpha t+\beta)$, причем коэффициенты α и β суть любые аналитические выражения, не зависящие от t и свои для каждого i. Известно, что первообразная функции вида (39) есть элементарная функция и может быть найдена при помощи стандартных, но довольно громоздких вычислений. Один из известных способов интегрирования функций вида (39) оформлен ниже в виде алгоритма В, блок-схема которого указана на рис.2. В этом алгоритме используются простые и обобщенные подстановки по следующим правилам (фигурные скобки $\{\cdot,\cdot\}$ опущены):

№ I. $\Delta\ \cos*(\xi+\varphi)\ \Delta\ \cos*\xi\times\cos*\varphi+\sin*\xi\times\sin*\varphi\times\bar{1}\ \Delta.$

№ 2. $\Delta\ \sin*(\xi+\varphi)\ \Delta\ \cos*\xi\times\sin*\varphi+\cos*\varphi\times\sin*\xi\ \Delta.$

№ 3. $\Delta\ e\uparrow(\xi+\varphi)\ \Delta\ e\uparrow\xi\times e\uparrow\varphi\ \Delta.$

№ 4. $\Delta\ (\xi+\varphi)\int t\ \Delta\ \xi\int t+\varphi\int t\ \Delta.$

№ 5. $\Delta\ \sin*\xi\uparrow\theta\ \Delta\ (\cos*(\xi\times 2)\times\frac{\bar{1}}{2}+\frac{1}{2})\times\sin*\xi\uparrow(\theta+\bar{2})\ \Delta.$

№ 6. $\Delta\ \cos*\xi\uparrow\theta\ \Delta\ (\cos*(\xi\times 2)\times\frac{1}{2}+\frac{1}{2})\times\cos*\xi\uparrow(\theta+\bar{2})\ \Delta.$

№ 7. $\Delta\ \cos*\xi\times\cos*\eta\ \Delta\ \cos*(\xi+\eta)\times\frac{1}{2}+\cos*(\xi+\eta\times\bar{1})\times\frac{1}{2}\ \Delta.$

№ 8. $\Delta\ \cos*\xi\times\sin*\eta\ \Delta\ \sin*(\xi+\eta)\times\frac{1}{2}+\sin*(\xi\times\bar{1}+\eta)\times\frac{1}{2}\ \Delta.$

№ 9. $\Delta\ \cos*\xi\times e\uparrow\eta\times\sin*\xi\ \Delta\ e\uparrow\eta\times\sin*(\xi+\xi)\times\frac{1}{2}+$
$e\uparrow\eta\times\sin*(\xi\times\bar{1}+\xi)\times\frac{1}{2}\ \Delta.$

№ II. $\Delta\ \varphi\times t\ \Delta\ t\times\varphi\ \Delta.$

№ IO. $\Delta\ \sin*\xi\times\sin*\eta\ \Delta\ \cos*(\xi+\eta)\times\frac{\bar{1}}{2}+\cos*(\xi+\eta\times\bar{1})\times 1/2\ \Delta.$

№ I2. $\Delta\ t+t\times\varphi\ \Delta\ t\times(\varphi+1)\ \Delta.$

№ I3. $\Delta\ t\times\varphi+t\ \Delta\ t\times(\varphi+1)\ \Delta.$

№ I4. $\Delta\ t\times\varphi+t\times\psi\ \Delta\ t\times(\varphi+\psi)\ \Delta.$

№ I5. $\Delta\ \alpha*t\ \Delta\ \alpha*(t\times 1)\ \Delta.$

№ I6. $\Delta\ e\uparrow t\ \Delta\ e\uparrow(t\times 1)\ \Delta.$

№ I7. $\Delta\ \cos*(t\times\psi)\times e\uparrow(t\times\varphi)\times t\uparrow\theta\int t\ \Delta\ ((\cos*(t\times\psi)\times e\uparrow(t\times\varphi)\times t\uparrow(\theta+\bar{1})\int t)\times\varphi\times\theta\times\bar{1}+(\cos*(t\times\psi)\times\varphi+\sin*(t\times\psi)$

$\times\psi)\times e\uparrow(t\times\varphi)\times t\uparrow\theta+(e\uparrow(t\times\varphi)\times\sin\times(t\times\psi)\times t\uparrow(\theta\times\bar{I})\int t$
$)\times\psi\times\theta\times\bar{I})\times(\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\ \Delta.$

№ I8. $\Delta e\uparrow(t\times\varphi)\times\sin\times(t\times\psi)\times t\uparrow\theta\int t\ \Delta((\cos\times(t\times\psi)\times e\uparrow(t\times\varphi)$
$\times t\uparrow(\theta\times\bar{I})\int t)\times\psi\times\theta+(\cos\times(t\times\psi)\times\psi\times\bar{I}+\sin\times(t\times\psi)\times\varphi)\times$
$e\uparrow(t\times\varphi)\times t\uparrow\theta+(e\uparrow(t\times\varphi)\times\sin\times(t\times\psi)\times t\uparrow(\theta\times\bar{I})\int t\times\varphi\times\theta\times$
$\bar{I})\times(\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\ \Delta.$

№ I9. $\Delta\cos\times(t\times\psi)\times e\uparrow(t\times\varphi)\times t\int t\ \Delta(((\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times(\varphi\uparrow 2\times$
$\bar{I}+\psi\uparrow 2)+t\times\varphi)\times\cos\times(t\times\psi)+((\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times\varphi\times\psi\times\bar{2}+$
$t\times\psi)\times\sin\times(t\times\psi))\times(\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times e\uparrow(t\times\varphi)\ \Delta.$

№ 20. $\Delta e\uparrow(t\times\varphi)\times\sin\times(t\times\varphi)\times t\int t\ \Delta(((\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times(\varphi\uparrow 2\times$
$\bar{I}+\psi\uparrow 2)+t\times\varphi)\times\sin\times(t\times\psi)+((\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times\varphi\times\psi\times 2+t\times$
$\psi\times\bar{I})\times\cos\times(t\times\psi))\times(\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times e\uparrow(t\times\varphi)\ \Delta.$

№ 2I. $\Delta\cos\times(t\times\psi)\times e\uparrow(t\times\varphi)\int t\ \Delta(\cos\times(t\times\psi)\times\varphi+\sin\times(t\times\psi)\times\psi)$
$\times(\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times e\uparrow(t\times\varphi)\ \Delta.$

№ 22. $\Delta e\uparrow(t\times\varphi)\times\sin\times(t\times\psi)\int t\ \Delta(\cos\times(t\times\psi)\times\Psi\times\bar{1}+\sin\times(t\times$
$\psi)\times\varphi)\times(\varphi\uparrow 2+\psi\uparrow 2)\uparrow\bar{I}\times e\uparrow(t\times\varphi)\ \Delta.$

№ 23. $\Delta\cos\times(t\times\varphi)\times t\uparrow\theta\int t\ \Delta(\sin\times(t\times\varphi)\times t\uparrow(\theta\times\bar{I})\int t)\times\varphi\uparrow$
$\bar{I}\times\theta\times\bar{I}+\sin\times(t\times\varphi)\times t\uparrow\theta\times\varphi\uparrow\bar{I}\ \Delta.$

№ 24. $\Delta\sin\times(t\times\varphi)\times t\uparrow\theta\int t\ \Delta(\cos\times(t\times\varphi)\times t\uparrow(\theta\times\bar{I})\int t)\times\varphi\uparrow\bar{I}$
$\times\theta+\cos\times(t\times\varphi)\times t\uparrow\theta\times\varphi\uparrow\bar{I}\times\bar{I}\ \Delta.$

№ 25. $\Delta\ cos*(t*\varphi)*t\int t\ \Delta\ cos*(t*\varphi)*\varphi\uparrow\bar{2}+sin*(t*\varphi)*t*\varphi\uparrow\bar{1}\ \Delta.$

№ 26. $\Delta\ sin*(t*\varphi)*t\int t\ \Delta\ cos*(t*\varphi)*t*\varphi\uparrow\bar{1}*\bar{1}+sin*(t*\varphi)*\varphi\uparrow\bar{2}\ \Delta.$

№ 27. $\Delta\ cos*(t*\varphi)\int t\ \Delta\ sin*(t*\varphi)*\varphi\uparrow\bar{1}\ \Delta.$

№ 28. $\Delta\ sin*(t*\varphi)\int t\ \Delta\ cos*(t*\varphi)*\varphi\uparrow\bar{1}*\bar{1}\ \Delta.$

№ 29. $\Delta\ e\uparrow(t*\varphi)*t\uparrow\theta\int t\ \Delta\ (e\uparrow(t*\varphi)*t\uparrow(\theta*\bar{1})\int t)*\varphi\uparrow\bar{1}*\theta*\bar{1}+e\uparrow(t*\varphi)*t\uparrow\theta*\varphi\uparrow\bar{1}\ \Delta$

№ 30. $\Delta\ e\uparrow(t*\varphi)*t\int t\ \Delta\ (t*\varphi\uparrow\bar{1}+\varphi\uparrow\bar{2}*\bar{1})*e\uparrow(t*\varphi)\ \Delta.$

№ 31. $\Delta\ e\uparrow(t*\varphi)\int t\ \Delta\ e\uparrow(t*\varphi)*\varphi\uparrow\bar{1}\ \Delta.$

№ 32. $\Delta\ t\uparrow\theta\int t\ \Delta\ t\uparrow(\theta*1)*(\theta*1)\uparrow\bar{1}\ \Delta.$

№ 33. $\Delta\ t\int t\ \Delta\ t\uparrow 2*1/2\ \Delta.$

№ 34. $\Delta\ 1\int t\ \Delta\ t\ \Delta$

Здесь переменное t есть постоянная буква алфавита Авто-Аналитика, θ, α, ξ, η, ζ, φ и ψ суть переменные буквы, причем $\theta\in\Phi_2$, $\alpha\in\Phi_1$, ξ, η и $\zeta\in\Phi_3$, φ и $\psi\in\Phi_0$.

Алгоритм В предполагает, что исходная формула Ш имеет вид $\alpha(\mathcal{F})\int t\ \Delta$ (или $\Delta\ \widetilde{\mathcal{F}}\int t\ \Delta$, если главная связь формулы $\mathcal{F}$ сильнее знака интеграла), где $\mathcal{F}$ – формула Авто-Аналитика, соответствующая обычной аналитической записи функции (39). На блок-схеме алгоритма приняты следующие сокращения:

"Арифметик" – означает: "Формула Ш обрабатывается оператором Арифметик; результат обозначается через Ш".

"Арифметик $*$" – означает: "В формуле Ш оператором Арифметик обрабатываются только подформулы вида $\Delta\ t\uparrow(\theta*\bar{1})\ \Delta$; результат обозначается через Ш".

"Алгоритм Б" – означает: "Формула Ш обрабатывается алгоритмом Б; результат обозначается через Ш".

"Циклич. n", где $n = 1,2,\ldots,34$, – означает: "В формулу Ш совершается циклическая обобщенная подстановка по правилу № n; результат обозначается через Ш".

"Строчная n" – означает: "В формулу Ш совершается строчная обобщенная подстановка по правилу № n; результат обозна-

чается через Ш".

"Строчная $*\,n$ " – означает: "Правый аргумент каждой связи $*$ в формуле Ш заменяется результатом строчной обобщенной подстановки в этом аргумент по правилу № n ; результат обозначается через Ш".

"Простая n " – означает: "В формулу Ш производится строчная простая подстановка по правилу № n ; результат обозначается через Ш".

"Множители" – означает: "Если в формуле Ш имеются подынтегральные выражения с главной связью $\times$, то все сомножители (= главные подформулы) таких выражений, не содержащие буквы t , выносятся за знак интеграла; результат обозначается через Ш".

Буквой h обозначен рабочий параметр алгоритма, принимающий значения 0 и I. Знаком + отмечены переходы, соответствующие случаям: "Условие выполнено" или "Последняя подстановка была возможной". Знаком – отмечены переходы в противоположных случаях.

Алгоритм В работает следующим образом. Сначала при помощи правил №№ I–2 исходная формула Ш освобождается от косинусов и синусов сумм. Затем в формуле Ш раскрываются скобки, приводятся подобные члены, интегралы сумм заменяются суммами интегралов от слагаемых и выносятся из-под знака интеграла постоянные сомножители. Применение правил №№ 5–I0, линейности интеграла и правил №№ II–I6 приводит исходную формулу Ш к виду

$$Ш = \Delta\, \widetilde{Ш}_1 + \widetilde{Ш}_2 + \cdots + \widetilde{Ш}_s\, \Delta \quad ,$$

где $s \geqslant 1$ и каждое слагаемое $\widetilde{Ш}_i$, $i = 1,2, \ldots, s$, с точностью до постоянного сомножителя есть интеграл от одной из функций

$$t^n e^{at}\cos bt,\ t^n e^{at}\sin bt,\ t e^{at}\cos bt,\ t e^{at}\sin bt,$$
$$e^{at}\cos bt,\ e^{at}\sin bt,\ t^n\cos at,\ t^n\sin at,\ t\cos at,$$
$$t\sin at,\ \cos at,\ \sin at,\ t^n e^{at},\ t e^{at},\ e^{at},\ t^n, t, 1$$

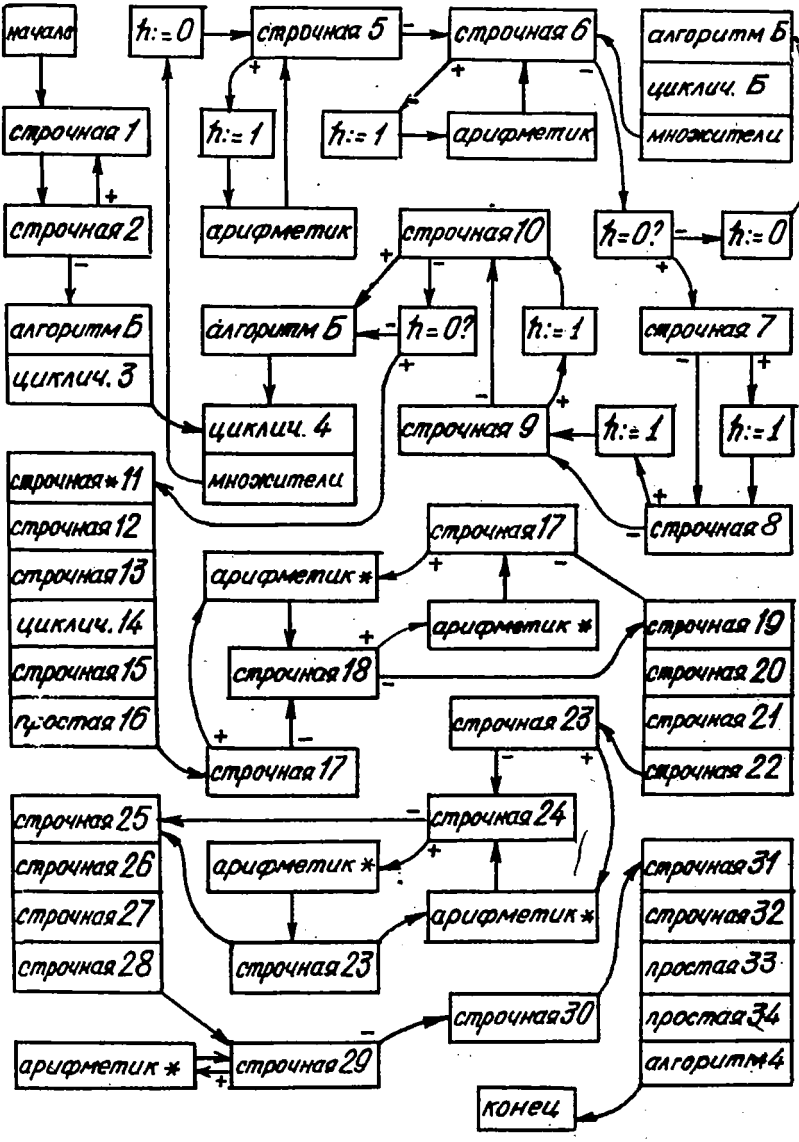


Рис.2. Блок-схема алгоритма В

с произвольным натуральным n и произвольными постоянными коэффициентами a и b. Эти интегралы заменяются соответствующими элементарными функциями с помощью правил №№ I7-34. В окончательном результате раскрываются скобки и приводятся подобные члены.

ПРИМЕР 4. ПРИВЕДЕНИЕ ФОРМУЛЫ ЛОГИКИ К ДИЗЪЮНКТИВНОЙ И КОНЪЮНКТИВНОЙ НОРМАЛЬНЫМ ФОРМАМ. При рассмотрении формул логики в рамках Авто-Аналитика естественно принять следующие соглашения. Логические связи суть знаки $\Leftrightarrow$, $\Rightarrow$, $\vee$, $\wedge$ и $\neg$ (см. также § I). Логические переменные суть постоянные буквы алфавита Авто-Аналитика (и быть может, произвольные нормальные формулы Авто-Аналитика, не содержащие логических связей). Логические значения "ложь" и "истина" обозначаются цифрами 0 и I соответственно. Логические формулы – это почти нормальные формулы Авто-Аналитика, выделяемые следующим рекурсивным способом: формула $\Delta\,\mathcal{A}\,\Delta$, где $\mathcal{A}$ есть логическое переменное или логическое значение, является логической формулой; если $\mathbb{Ш}$ и $\mathbb{Щ}$ – две логические формулы, то почти нормальные виды формул

$$\Delta\,(\widehat{\mathbb{Ш}})\Leftrightarrow(\widehat{\mathbb{Щ}})\,\Delta\,(\widehat{\mathbb{Ш}})\Rightarrow(\widehat{\mathbb{Щ}})\,\vee\,(\widehat{\mathbb{Щ}})\,\Delta\,(\widehat{\mathbb{Ш}})\,\wedge\,(\widehat{\mathbb{Щ}})\,\Delta\,\neg\,(\widehat{\mathbb{Ш}})\,\Delta$$

суть логические формулы.

В этой интерпретации формул логики элементарной конъюнкцией следует назвать логическую формулу с главной связью $\wedge$, все главные подформулы которой суть либо логические переменные и логические значения, либо их отрицания. Логические формулы, не содержащие логических связей вообще либо содержащие только одну логическую связь и эта связь есть знак отрицания $\neg$, также удобно объявить элементарной конъюнкцией. Дизъюнктивной нормальной формой данной логической формулы $\mathbb{Ш}$ называется логическая формула $\mathbb{Щ}$, которая логически равносильна формуле $\mathbb{Ш}$ и представляет собой либо одну элементарную конъюнкцию, либо дизъюнкцию конечного числа элементарных конъюнкций.

Заменяя здесь слово "конъюнкция" словом "дизъюнкция" и наоборот, мы получим определения элементарной дизъюнкции и конъюнктивной нормальной формы данной логической формулы $\mathbb{Ш}$.

В курсе математической логики доказывается, что всякая логическая формула имеет хотя бы одну дизъюнктивную нормальную форму и хотя бы одну конъюнктивную нормальную форму.

Приведение логической формулы Ш к дизъюнктивной или конъюнктивной нормальной форме осуществляется алгоритмом Г, блок-схема которого указана на рис.3. Кроме формулы Ш в качестве

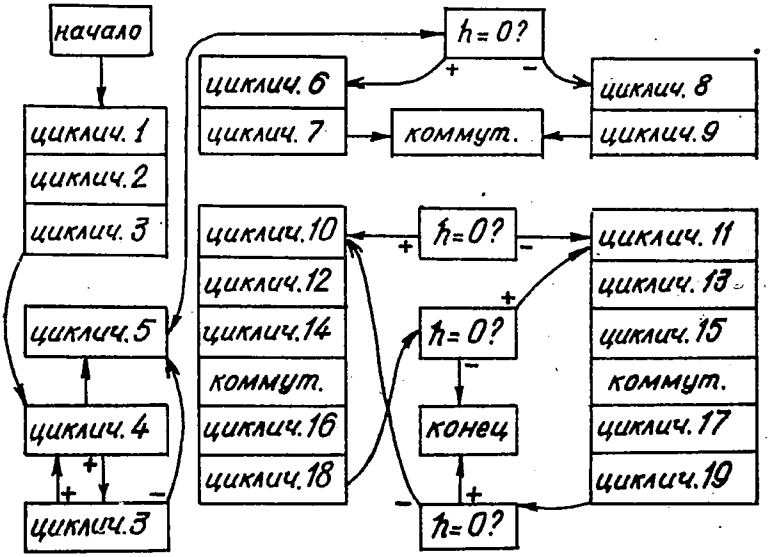


Рис. 3. Блок-схема алгоритма Г

исходной информации этот алгоритм использует параметр h, значение которого равно нулю, если нужно найти дизъюнктивную нормальную форму, и отлично от нуля, если речь идет о конъюнктивной нормальной форме. Блок "Коммут." означает полное коммутирование формулы Ш. Прочие обозначения здесь те же, что и на блок-схеме алгоритма В. Номерами I + I9 в алгоритме 6 обозначены правила

№ I. $\Delta\ \xi \Rightarrow \eta\ \Delta\ \neg\xi \vee \eta\ \Delta$,

№ 2. $\Delta\ \xi \Leftrightarrow \eta\ \Delta\ \xi \wedge \eta \vee \neg\xi \wedge \neg\eta\ \Delta$,

№ 3. $\Delta\ \neg(\xi \vee \varphi)\ \Delta\ \neg\xi \wedge \neg\varphi\ \Delta$,

№ 4. $\Delta\ \neg(\xi \wedge \varphi)\ \Delta\ \neg\xi \vee \neg\varphi\ \Delta$,

№ 5. $\Delta\ \neg(\neg\xi)\ \Delta\ \xi\ \Delta$,

№ 6. $\Delta\ (\xi \vee \varphi) \wedge \eta\ \Delta\ \xi \wedge \eta \vee \varphi \wedge \eta\ \Delta$,

№ 7. $\Delta\ \xi \wedge (\eta \vee \varphi)\ \Delta\ \xi \wedge \eta \vee \xi \wedge \varphi\ \Delta$,

№ 8. $\Delta\ \xi \wedge \varphi \vee \eta\ \Delta\ (\xi \vee \eta) \wedge (\varphi \vee \eta)\ \Delta$,

№ 9. $\Delta\ \xi \vee \eta \wedge \varphi\ \Delta\ (\xi \vee \eta) \wedge (\xi \vee \varphi)\ \Delta$,

№ I0. $\Delta\ \xi \wedge \xi\ \Delta\ \xi\ \Delta$,

№ II. $\Delta\ \xi \vee \xi\ \Delta\ \xi\ \Delta$,

№ I2. $\Delta\ \neg\xi \wedge \varphi \wedge \xi\ \Delta\ 0\ \Delta$,

№ I3. $\Delta\ \neg\xi \vee \varphi \vee \xi\ \Delta\ 1\ \Delta$,

№ I4. $\Delta\ \neg\xi \wedge \xi\ \Delta\ 0\ \Delta$,

№ I5. $\Delta\ \neg\xi \vee \xi\ \Delta\ 1\ \Delta$,

№ I6. $\Delta\ \xi \wedge 0\ \Delta\ 0\ \Delta$,

№ I7. $\Delta\ \xi \vee 0\ \Delta\ \xi\ \Delta$,

№ I8. $\Delta\ \xi \wedge 1\ \Delta\ \xi\ \Delta$,

№ I9. $\Delta\ \xi \vee 1\ \Delta\ 1\ \Delta$,

где ξ и η – переменные буквы типа Φ_3, φ – переменная буква типа Φ_0.

В заключение данного параграфа приведем пример алгоритма, который непосредственно операцию обобщенной подстановки не использует, но опирается на алгоритм Б приведения подобных и на алгоритм В поиска первообразной и, стало быть, основан в конечном счете на обобщенной подстановке.

ПРИМЕР 5. ОРТОГОНАЛИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ПОЛИНОМОВ. Говорят, что полиномы $P(t)$ и $Q(t)$ одного вещественного переменного t ортогональны на отрезке $[a, b]$, если их скалярное произведение

$$(P, Q) = \int_a^b P(t)\, Q(t)\, dt$$

равно нулю. Задача ортогонализации данной последовательности линейно независимых полиномов $P_0(t), P_1(t), \ldots, P_n(t), \ldots$ на $[a, b]$ заключается в построении такой последовательности попарно ортогональных полиномов $Q_0(t), Q_1(t), \ldots, Q_n(t), \ldots$, что при любом $n = 0, 1, 2, \ldots$ норма

$$\|Q_n\| = \left(\int_a^b |Q_n(t)|^2\, dt\right)^{\frac{1}{2}}$$

полинома Q_n равна I и (n + I) -мерные векторные функциональные пространства, порождённые двумя системами полиномов

$$\{P_0(t), P_1(t), \ldots, P_n(t)\}, \{Q_0(t), Q_1(t), \ldots, Q_n(t)\}$$

совпадают. Эта задача обычно решается следующим методом (метод Шмидта). Полином $Q_0(t)$ полагается равным полиному $P_0(t)$, поделенному на его норму. Пусть полиномы $Q_k(t)$, k = = 0,I, ... , n - I, уже построены. Известно, что разность

$$Z_n(t) = P_n(t) - R_n(t),$$

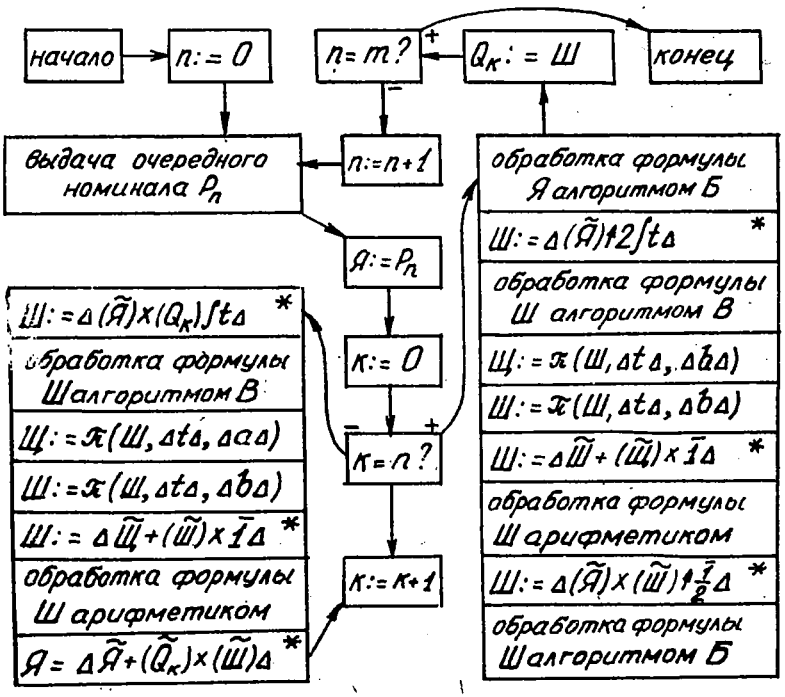


Рис. 4. Блок-схема алгоритма Д

где $R_n(t)$ есть сумма ряда Фурье полинома $P_n(t)$ по ортонормированной системе

$$\{Q_0(t), Q_1(t), \ldots, Q_{n-1}(t)\}, \qquad (40)$$

т.е.

$$R_n(t) = \sum_{k=0}^{n-1} (P_n, Q_k) Q_k(t),$$

ортогонально пространству, порожденному системой (40). Полином $Q_n(t)$ полагается разности $Z_n(t)$, поделенной на ее норму.

Блок-схема алгоритма Д, реализующего метод Шмидта, приведена на рис.4. В качестве исходной информации этот алгоритм требует формулы $\Delta\,\alpha\,\Delta$, $\Delta\,\beta\,\Delta$, натуральное число m и исходные полиномы $P_0(t), P_1(t), \ldots, P_{m-1}(t)$. Эти полиномы можно заготовить заблаговременно, но можно также указать алгоритм для их вычисления. Формулы, получающиеся в качестве промежуточных результатов, обозначаются буквами Ш, Щ и Я. Символ π (Ш, Э, Ю) означает результат строчной простой подстановки в формулу Ш по правилу {Э, Ю}. Звездочкой * в правом верхнем углу отмечены те блоки, к описанию которых нужно добавить фразу: "Полученная формула приводится к почти нормальному виду". В качестве результата выдаются полиномы $Q_0(t)$, $Q_1(t)$, ..., $Q_m(t)$.

Метод Шмидта годен для ортогонализации любой последовательности функций. Возможности соответствующего этому методу алгоритма Д ограничены лишь возможностями алгоритма В. Алгоритм Д в состоянии ортогонализировать любую (конечную) последовательность алгебраических, показательных и тригонометрических полиномов. В иных ситуациях алгоритм Д можно будет применять лишь после того, как будет соответствующим образом усовершенствован алгоритм В.

§ 12. КОММУТАТИВНАЯ ПОДСТАНОВКА

Примеры, рассмотренные в предыдущих параграфах, дают основания полагать, что простая и обобщенная подстановки являются

довольно универсальным и незаменимым орудием при проведении разнообразных аналитических преобразований. Однако они игнорируют коммутативность коммутативных связей. Поэтому иногда возникают ситуации, в которых эти операции не дают необходимого эффекта. Например, простая подстановка в формулу

$$\Delta\, a + \cos * (a + d) + d\, \Delta$$

по правилу $\{\Delta\, a + d\, \Delta\ W\, \Delta$ приводит к формуле $\Delta\, a + \cos * W + d\, \Delta$, хотя здесь повидимому более полезен был бы результат $\Delta\, \cos * W + W\, \Delta$. Обобщенная подстановка в формулу

$$Ш = \Delta\, a + b + c + \cos * (a + c) + \cos * (b + d) + d\, \Delta \qquad (41)$$

по правилу

$$\{\Delta\, \cos * \varphi + \varphi\, \Delta\ f * \varphi\, \Delta\}, \quad \varphi \in \Phi_0 \qquad (42)$$

вообще невозможна, в то время как эквивалентная почти нормальная, но не нормальная формула

$$Ш' = \Delta\, \cos * (a + c) + a + c + \cos * (b + d) + b + d\, \Delta \qquad (43)$$

преобразуется в данном случае к виду

$$\Delta\, f * (a + c) + f * (b + d\, \Delta \qquad (44)$$

Естественно и в случае формулы Ш получить такой же результат. Обобщенная подстановка по правилу

$$\{\Delta\, \varphi * \psi * 2 + \varphi \uparrow 2 + \psi \uparrow 2\ \Delta\ (\varphi + \psi) \uparrow 2\, \Delta\}, \quad \varphi, \psi \in \Phi_0 \qquad (45)$$

верно преобразует формулы

$$\Delta\, a * b * 2 + a \uparrow 2 + b \uparrow 2\ \Delta\ \cos * a * \sin * b * 2 + \cos * a \uparrow 2 + \sin * b \uparrow 2\ \Delta, \qquad (46)$$

однако бесполезна в случае формул

$$\Delta\, (b/c) \uparrow 2 + a * (b/c) * 2 + a \uparrow 2\ \Delta\ (a * c) \uparrow 2 + a * b * c * 2 + b \uparrow 2\ \Delta. \qquad (47)$$

Приведение подобных слагаемых в формуле

$$Ш = \Delta\, ((f * t + g * t) \int t) * 2 + (f * t + g * t) \int t + (f * t \int t) * 2 +$$

$$a + a * b + a * b * 2 + a * 2 + b + b * 2 + f * t \int t\, \Delta \qquad (48)$$

требует применения четырех различных правил

$$\{\Delta\varphi+\psi+\varphi\times\theta\,\Delta\,\psi+\varphi\times(\theta+1)\,\Delta\},\ \theta\in\Phi_2,\ \varphi,\psi\in\Phi_0, \tag{49}$$

$$\{\Delta\varphi+\varphi\times\theta\,\Delta\,\varphi\times(\theta+1)\,\Delta\},\ \theta\in\Phi_2,\ \varphi\in\Phi_0, \tag{50}$$

$$\{\Delta(\xi)\times\theta+\varphi+\xi\,\Delta\,\xi\times(\theta+1)+\varphi\,\Delta\},\ \theta\in\Phi_2,\ \xi\in\Phi_3,\ \varphi\in\Phi_0,$$

$$\{\Delta(\xi)\times\theta+\xi\,\Delta\,\xi\times(\theta+1)\,\Delta\},\ \theta\in\Phi_2,\ \xi\in\Phi_3,$$

а формула

$$\text{Ш}'=\Delta((f*t+g*t)\int t+((f*t+g*t)\int t)\times 2+f*t\int t+(f*t\int t)\times 2+a\times b+a\times b\times 2+a+a\times 2+b+b\times 2\,\Delta, \tag{51}$$

эквивалентная формуле Ш, приводится к тому же результату

$$\Delta((f*t+g*t)\int t)\times(2+1)+(f*t\int t)\times(2+1)+a\times b\times(2+1)+a\times(2+1)+b\times(2+1)\,\Delta \tag{52}$$

при помощи одного правила (50).

Ознакомившись с подобными примерами, читатель несомненно найдет вполне естественными следующие обобщения простой и обобщенной подстановок.

ОПРЕДЕЛЕНИЕ 26. Пусть заданы почти нормальная формула Ш и правило $\{Э, Ю\}$. По определению простая (обобщенная) коммутативная подстановка в формулу Ш по правилу $\{Э, Ю\}$ возможна тогда и только тогда, когда существует почти нормальная формула Ш', эквивалентная формуле Ш и представимая в виде

$$\text{Ш}'=\Delta\,\alpha_1\alpha_2\ldots\alpha_{i-1}\,\widetilde{Э}_I\,\alpha_j\alpha_{j+1}\ldots\alpha_n\,\Delta, \tag{53}$$

где подформула

$$Э_I=\Delta\,\alpha_i\alpha_{i+1}\ldots\alpha_{j-1}\,\Delta$$

совпадает с левой частью (является аналогом левой части) правила $\{Э, Ю\}$. Результатом простой (обобщенной) коммутативной под-

становки в этом случае объявляется нормальный вид формулы

$$\Delta\, a_1 a_2 \ldots a_{i-1}\ (\tilde{\imath} O_1)\ a_j a_{j+1} \ldots a_n\, \Delta,$$

где подформула $Ю_I$ совпадает с правой частью правила {Э, Ю} (является аналогом правой части правила {Э, Ю}, соответствующим аналогу $Э_I$ его левой части). Если простая (обобщенная) коммутативная подстановка в формулу Ш по правилу {Э, Ю} невозможна, то ее результатом объявляется нормальная формула, эквивалентная формуле Ш.

Для данной почти нормальной формулы Ш может существовать несколько различных почти нормальных формул, эквивалентных формуле Ш и имеющих вид (53). Поэтому следует ожидать (и примеры это подтверждают), что простая и обобщенная коммутативные подстановки вообще говоря определены неоднозначно. Многозначность коммутативных подстановок в принципе можно устранить, если дополнительно потребовать, чтобы среди всех почти нормальных формул вида (53), эквивалентных формуле Ш, всегда выбиралась та, которая например слабее (в смысле определения 2) любой другой такой же формулы. Однако подобное требование приводит к чрезмерному усложнению и без того далеко не тривиальных алгоритмов коммутативных подстановок. С другой стороны, в приложениях конкретный выбор результата коммутативной подстановки чаще всего не играет никакой роли (и обычно удовлетворяет тот довольно случайный вариант результата, который выдается оператором коммутативной подстановки, описанным в главе IУ).

Коммутативная подстановка в две эквивалентные почти нормальные формулы по одному и тому же правилу очевидно приводит к одним и тем же результатам. Легко также видеть, что замена правила {Э, Ю} правилом $\{Э_0, Ю_0\}$, где $Э_0$ и $Ю_0$ суть результаты преобразования формул Э и Ю оператором Коммутирование, не меняет множества результатов коммутативной подстановки в фиксированную формулу Ш по данному правилу. Поэтому мы будем говорить, что два правила {Э, Ю} и {Э′, Ю′} равносильны, если оператор Коммутирование приводит их к одному и тому же правилу $\{Э_0, Ю_0\}$. (Во избежание недоразумений подчеркнем, что оператор коммутативной подстановки, описанный в главе IУ, выдает лишь один результат из многих возможных и этот результат может измениться при переходе от формулы Ш и правила {Э, Ю} к эквива-

лентной формуле Ш и равносильному правилу $\{Э', Ю'\}$).

Легко видеть, что нормальный вид результата однократной простой (обобщенной) подстановки в формулу, эквивалентную формуле Ш, по правилу, равносильному правилу $\{Э, Ю\}$, принадлежит множеству результатов простой (обобщенной) коммутативной подстановки в формулу Ш по правилу $\{Э, Ю\}$. Обратное вообще говоря неверно. Например, формула $\Delta\ (a+b)\ \dagger\ W\ \Delta$ содержится в семействе результатов простой коммутативной подстановки в формулу

$$Ш = \Delta\ (a+b)\ \dagger\ (a+b)\ \Delta$$

по правилу $\{\Delta a+b\Delta\ W\ \Delta\}$, но не совпадает с нормальным видом любого результата однократной простой подстановки в формулу, эквивалентную формуле Ш, по правилу, равносильному данному правилу.

Приведем еще несколько примеров. Простая коммутативная подстановка в формулу $\Delta\ a+b+c\ \Delta$ по правилу $\{\Delta\ a+c\ \Delta W\Delta\}$ приводит к формуле $\Delta\ b+W\ \Delta$. Простая коммутативная подстановка в формулу

$$\Delta\ a*b*c*d*f+\cos*(a*b*f*c)+\sin*(c*f*a)\Delta \qquad (54)$$

по правилу

$$\{\Delta\ a*c*f\ \Delta\ W\Delta\}$$

в качестве результата дает одну из следующих трех формул

$$\Delta\ b*d*W+\cos*(a*b*c*f)+\sin*(a*c*f)\ \Delta,$$
$$\Delta\ a*b*c*d*f+\cos*(b*W)+\sin*(a*c*f)\ \Delta,$$
$$\Delta\ a*b*c*d*f+\cos*(a*b*c*f)+\sin*W\ \Delta.$$

Обобщенная коммутативная подстановка в формулу (41) или (43) по правилу (42) или по равносильному правилу

$$\{\Delta\ y+\cos*y\ \Delta\ f*y\ \Delta\},\quad y\in\Phi_0$$

приводит к одной из формул

$$\Delta\ b+\cos*(b+d)+d+f*(a+c)\ \Delta,$$

$$\Delta\ a+c+\cos*(a+c)+f*(b+d)\Delta.$$

Результаты обобщенной коммутативной подстановки в формулы (46) и (47) по правилу (45) или по любому из правил

$$\{\Delta\ \varphi\times 2\times\psi+\varphi\uparrow 2+\psi\uparrow 2\ \Delta\ (\varphi+\psi)\uparrow 2\ \Delta\},\quad \varphi,\psi\in\Phi_0,$$

$$\{\Delta\ 2\times\psi\times\varphi+\varphi\uparrow 2+\psi\uparrow 2\ \Delta\ (\varphi+\psi)\uparrow 2\ \Delta\},\quad \varphi,\psi\in\Phi_0,$$

$$\{\Delta\ \varphi\uparrow 2+\varphi\cdot\psi\times 2+\psi\uparrow 2\ \Delta\ (\psi+\varphi)\uparrow 2\ \Delta\},\quad \varphi,\psi\in\Phi_0,$$

$$\{\Delta\ \varphi\uparrow 2+\psi\uparrow 2+\varphi\times\psi\times 2\ \Delta\ (\varphi+\psi)\uparrow 2\ \Delta\},\quad \varphi,\psi\in\Phi_0,$$

.....

равносильных правилу (45), определены однозначно и суть соответственно формулы

$$\Delta\ (a+b)\uparrow 2\ \Delta\ (\cos*a+\sin*b)\uparrow 2\ \Delta\ (a+b/c)\uparrow 2\ \Delta\ (a\times c+b)\uparrow 2\ \Delta. \quad (55)$$

Результатом обобщенной коммутативной подстановки по правилу (49) или (50) (или по любому другому правилу, равносильному правилу (49) или (50)) в формулы (48) или (51) может служить любая из следующих пяти формул

$$\Delta\ ((f*t+g*t)\int t)\times(1+2)+f*t\int t)\times 2+a+a\times b+a\times b\times 2+$$
$$a\times 2+b+b\times 2+f*t\int t\ \Delta,$$

$$\Delta\ ((f*t+g*t)\int t)\times 2+(f*t+g*t)\int t+(f*t\int t)\times(1+2)+$$
$$a+a\times b+a\times b\times 2+a\times 2+b+b\times 2\ \Delta,$$

$$\Delta\ ((f*t+g*t)\int t)\times 2+(f*t+g*t)\int t+(f*t\int t)\times 2+$$
$$a\times(1+2)+a\times b+a\times b\times 2+b+b\times 2+f*t\int t\ \Delta,$$

$$\Delta\,((f*t+g*t)\int t)\times 2+(f*t+g*t)\int t+(f*t\int t)\times 2+$$
$$a+a*b*(1+2)+a*2+b+b*2+f*t\int t\,\Delta,$$
$$\Delta\,((f*t+g*t)\int t)\times 2+(f*t+g*t)\int t+(f*t\int t)\times 2+$$
$$a+a*b+a*b\times 2+a\times 2+b\times(1+2)+f*t\int t\,\Delta.$$

В качестве результата обобщенной коммутативной подстановки в формулу

$$\Delta\, a*b+\varphi+d+\psi\,\Delta$$

по правилу $\{\Delta\varphi+\psi\,\Delta\ W\Delta\}$, где переменные буквы φ и ψ имеют тип Φ_0, можно выбрать любую из следующих 26-ти формул

$$\Delta\,a+b+d+w\,\Delta\quad a+b+w+\varphi\,\Delta\quad a+b+w+\psi\,\Delta\quad a+d+w+\varphi\,\Delta\quad a+d+w+\psi\,\Delta$$
$$a+w+\varphi+\psi\,\Delta\quad b+d+w+\varphi\,\Delta\quad b+d+w+\psi\,\Delta\quad b+w+\varphi+\psi\,\Delta$$
$$d+w+\varphi+\psi\,\Delta\quad a+b+w\,\Delta\quad a+d+w\,\Delta\quad a+w+\varphi\,\Delta\quad a+w+\psi\,\Delta$$
$$b+d+w\,\Delta\quad b+w+\varphi\,\Delta\quad b+w+\psi\,\Delta\quad d+w+\varphi\,\Delta\quad d+w+\psi\,\Delta\quad w+\varphi+\psi\,\Delta$$
$$a+w\,\Delta\quad b+w\,\Delta\quad d+w\,\Delta\quad w+\varphi\,\Delta\quad w+\psi\,\Delta\,W\Delta$$

ОПРЕДЕЛЕНИЕ 27. Для любой почти нормальной формулы Ш и любого правила {Э, Ю} существует и притом вообще говоря не одна последовательность формул

$$Ш_1,\ Ш_2,\ \ldots,\ Ш_m,\ \ldots, \qquad (56)$$

в которой $Ш_1 = Ш$ и для каждого $m = 2, 3, \ldots$ формула $Ш_m$ есть один из результатов простой (соответственно обобщенной) коммутативной подстановки в формулу $Ш_{m-1}$ по правилу {Э, Ю}. Всякую такую последовательность (56) мы будем именовать процессом вычисления полной простой (обобщенной) коммутативной подстановки в формулу Ш по правилу {Э, Ю}. Если для данного процесса (56) существует такой индекс $m \geq 1$, что в формулу $Ш_m$ простая (обобщенная) коммутативная подстановка по правилу {Э, Ю} невозможна, то нормальный вид формулы $Ш_m$, совпадающий с самой формулой $Ш_m$ при $m \geq 2$ и с формулой $Ш_2$ при $m = 1$, естественно объявить результатом полной простой (обобщенной) коммутативной подстановки в формулу Ш по правилу {Э, Ю}. В противном случае мы будем говорить, что данный процесс (56) зацикливает.

ПРИМЕРЫ. Полная простая коммутативная подстановка в формулу

$$\Delta a + cos * (a + d) + d \Delta$$

по правилу $\{ \Delta a + d \Delta W \Delta \}$ приводит к единственному результату $\Delta cos * W + W \Delta$, но имеются два процесса его вычисления, а именно

$$\Delta a + cos * (a + d) + d \Delta \; a + cos * W + d \Delta \; cos * W + W \Delta \; cos * W + W \Delta \ldots,$$

и

$$\Delta a + cos * (a + d) + d \Delta cos * (a + d) + W \Delta \; cos * W + W \Delta$$

$$cos * W + W \Delta \ldots,$$

Полная простая коммутативная подстановка в формулу (54) по правилу

$$\{ \Delta a * c * f \Delta W \Delta \}$$

также дает единственный результат

$$\Delta \beta * d * W + cos * (\beta * W) + sin * W \Delta .$$

Этот результат может быть получен шестью различными способами.

Полная обобщенная коммутативная подстановка в формулу (41) по правилу (42) двумя путями приводит к одному и тому же результату (44). Формулы (55) являются также результатами полной обобщенной коммутативной подстановки в формулы (46) и (47) по правилу (45). Полная обобщенная коммутативная подстановка в формулу (48) по правилу (49) или (50) имеет единственный результат (52).

В рассмотренных примерах результат полной простой или обобщенной коммутативной подстановки каждый раз был определен однозначно. Повидимому вообще полная коммутативная подстановка чаще приводит к однозначному результату, нежели неполная. Однако не следует ожидать, что этот результат будет однозначным всегда. Рассмотрим например формулу

$$\Psi = \Delta a + a + a + a + a + a + a \Delta . \qquad (57)$$

и правило $\{ \Delta \xi + \xi \Delta \; \xi * 2 \Delta \}$, где $\xi \in \Phi_0$. В данном случае авторы нашли 25 различных процессов вычисления результата полной обобщенной коммутативной подстановки. Из них 20 процессов и в том числе процессы

$\Delta\, a+a+a+a+a+a+a+a\, \Delta\, (a+a+a+a)\times 2\, \Delta\, (a+a)\times 2\times 2\, \Delta\, a\times 2\times 2\times 2\Delta\ldots,$

$\Delta\, a+a+a+a+a+a+a+a\, \Delta\, a+a+a+a+a+a\times 2\, \Delta\, a+a+a$

$+a+a\times 2+a\times 2\, \Delta\, a+a+a\times 2+a\times 2+a\times 2\, \Delta\, a\times 2+a\times 2+a\times 2$

$+a\times 2\, \Delta\, a\times 2+a\times 2+a\times 2\times 2\, \Delta\, a\times 2\times 2+a\times 2\times 2\, \Delta\, a\times 2\times 2\times 2\Delta\ldots,$

$\Delta\, a+a+a+a+a+a+a+a\, \Delta\, a+a+a+a+a+a+a\times 2\, \Delta$

$a+a+a+a+a\times 2+a\times 2\, \Delta\, (a+a+a\times 2)\times 2\, \Delta$

$(a\times 2+a\times 2)\times 2\, \Delta\, a\times 2\times 2\times 2\, \Delta\ldots,$

приводят к результату $a\times 2\times 2\times 2$, а остальные 5 процессов и в том числе процесс

$\Delta\, a+a+a+a+a+a+a+a\, \Delta\, (a+a+a)\times 2+a+a\, \Delta\, (a+a\times 2)\times 2+a$

$+a\, \Delta\, (a+a\times 2)\times 2+a\times 2\, \Delta\ldots,$

$\Delta\, a+a+a+a+a+a+a+a\, \Delta\, a+a+a+a+a+a\times 2\, \Delta\, a+a+$

$a+a+a\times 2+a\times 2\, \Delta (a+a\times 2)\times 2+a+a\, \Delta\, (a+a\times 2)\times 2+a\times 2\, \Delta\ldots$

приводят к результату

$\Delta\, (a+a\times 2)\times 2+a\times 2\, \Delta$

Если в правиле $\{\Delta\xi+\xi\, \Delta\, \xi\times 2\, \Delta\}$ считать, что $\xi\in\Phi_I$ или

$\xi \in \Phi_3$, то результат полной обобщенной коммутативной подстановки в формулу (57) по этому правилу снова будет однозначно определен и равен формуле

$$\Delta\ a*2+a*2+a*2+a*2\ \Delta .$$

в случае $\xi \in \Phi_1$ и формуле $\Delta\ a*2*2*2\ \Delta$ в случае $\xi \in \Phi_3$.

Нетрудно привести пример, когда любой процесс вычисления результата полной коммутативной подстановки зацикливает. Для этого достаточно взять любую почти нормальную формулу Ш, не содержащую переменных букв типа Φ_2, и положить Э = Ю = Ш.

Возможны также ситуации, когда один процесс вычисления результата полной простой (обобщенной) коммутативной подстановки в формулу Ш по правилу $\{Э, Ю\}$ зацикливает, а другой – при тех же самых Ш и $\{Э, Ю\}$ – не зацикливает. Рассмотрим например формулу.

$$Ш = \Delta\ A*B*B*A*B*A*B*B*A\ \Delta \qquad (58)$$

и правило

$$\{Э, Ю\} = \{\Delta\ A*B*B*A\ \Delta, \Delta\ B*A*B*A*C*A*B*A*B\ \Delta\} \qquad (59)$$

(напомним, что знак функциональной зависимости * в этой главе мы считаем ассоциативной, но не коммутативной связью). Последовательность формул $Ш_1$, $Ш_2$, ... , где $Ш_1 = Ш$,

$$Ш_2 = \Delta\ A*B*B*A*B*B*A*B*A*C*A*B*A*B\ \Delta$$

и для $m > 3$

$$Ш_m = \Delta\ B*A*B*A*C*A*B*A*B*B*B*A*B*A*C*\underset{A*B*A*B}{} \Delta \qquad (60)$$

есть процесс вычисления результата полной простой коммутативной подстановки в формулу (58) по правилу (59), приводящий к результату (60). С другой стороны, последовательность формул

$$Ш_1 = Ш = \Delta\ A*B*B*A*B*A*B*B*A\ \Delta ,$$

$$Ш_2 = \Delta\ B*A*B*A*C*A*B*A*B*B*A*B*B*A\ \Delta ,$$

$$Ш_3 = \Delta\ B*A*B*A*C*A*B*B*A*B*A*C*A*B*A*B*B*B*A\ \Delta ,$$

$$Ш_4 = \Delta\ B*A*B*A*C*B*A*B*A*C*A*B*A*B*B*A*C*A*B*A*B*B*B*A\ \Delta ,$$

. . . . . .

является зацикливающим процессом при тех же самых Ш и {Э, Ю} .

Коммутативная подстановка очевидно универсальнее некоммутативной подстановки. Применение коммутативной подстановки как правило значительно сокращает и упрощает алгоритмы решения конкретных задач. Тем не менее частое использование коммутативной постановки нежелательно ввиду того, что вычисление ее результата занимает слишком много времени. Алгоритм любой обобщенной подстановки вообще, а коммутативной подстановки в особенности, подавляющую часть времени своей работы затрачивает на выявление аналогов левой части правила Э в преобразуемой формуле Ш. Поиск аналогов в формуле Ш грубо говоря заключается в переборе всех "подозрительных" подформул этой формулы (а в случае коммутативной подстановки – еще и подформул тех формул, которые эквивалентны формуле Ш). Подформулы, подозрительные в том смысле, что они могут оказаться искомыми аналогами всей левой части Э или некоторой ее подформулы и, в частности, аналогами переменных букв из Э, выявляются схемами упорядочения перебора. Эти схемы прежде всего бракуют подформулы, не соответствующие типам переменных букв. Поэтому при прочих равных условиях во всех правилах тип Φ_2 (число) предпочтительнее типа Φ_1 (буква), тип Φ_1 предпочтительнее типа Φ_3 (одночлен), а тип Φ_3 предпочтительнее типа Φ_0 (произвольная подформула). Далее, схемы упорядочения перебора игнорируют те подформулы, в которых отсутствует хотя бы один из элементарных символов, содержащихся в формуле Э и не являющихся переменными буквами. Поэтому правило с "более детальной" левой частью предпочтительнее правила с "менее детальной" левой частью (например, правила с левыми частями

$$\Delta(\xi)\times\zeta\,\Delta,\ \Delta\,\xi\times 2+\zeta\,\Delta,\ \Delta\,\cos*(\xi+\zeta)\,\Delta,\ \Delta\,\sin*\xi\,\Delta,\ldots$$

где ξ и ζ – переменные буквы, предпочтительнее правил с левыми частями

$$\Delta\,\xi*\zeta\,\Delta,\ \Delta\,\xi+\zeta\,\Delta,\ \Delta\,\cos*\xi\,\Delta,\ \Delta\,\zeta*\xi\,\Delta,\ldots$$

соответственно). Наконец, в некоммутативной подстановке анализируются подформулы лишь самой формулы Ш и не включаются в перебор подформулы формул, эквивалентных Ш. Следовательно, применение некоммутативной обобщенной подстановки значительно вы-

годнее (по времени), чем применение обобщенной коммутативной подстановки.

Вообще оценка времени работы оператора подстановки – задача нелегкая, так как оно существенно зависит от многих параметров – длины формул Ш и Э, длины и количества тех подформул формулы Ш, главные связи которых совпадают с главной связью формулы Э, начиная в формуле Э переменных букв и их распределения по типам и т.д. Можно лишь утверждать (и то весьма неуверенно), что время работы некоммутативной обобщенной подстановки по правилу $\{Э, Ю\}$, в котором все переменные буквы имеют тип Φ_3, время работы некоммутативной обобщенной подстановки по правилу $\{Э, Ю\}$, полученному из правила $\{Э, Ю\}$ заменой переменных букв типа Φ_3 переменными буквами типа Φ_0, и время обобщенной коммутативной подстановки по правилу $\{Э, Ю\}$ для всякой формулы Ш относятся друг к другу как $t : t^2 : 2^t$.

Итак, использование различных модификаций простых и обобщенных подстановок должно производиться с соблюдением по крайней мере следующих трех принципов:

1) Коммутативную подстановку следует применять только тогда, когда без нее обойтись невозможно.

2) Переменные буквы типа Φ_0 следует вводить также лишь в исключительных случаях. Вообще с точки зрения времени всегда полезно менее предпочтительный тип переменной буквы заменять типом более предпочтительным.

3) Если условия задачи позволяют заменить "слишком общее" правило более детальным правилом, то обычно такая замена влечет определенную экономию времени.

ГЛАВА II

РЕАЛИЗАЦИЯ АВТО-АНАЛИТИКА НА БЭСМ-6

§ I. ОПИСАНИЕ ЭЦВМ БЭСМ-6

Для математика ЭЦВМ БЭСМ-6 представляет собой неделимое единство ее аппаратного и математического обеспечения. Аппаратное обеспечение представлено системой доступных программисту команд и регистров машины. Математическое обеспечение диктует пользователю возможности и способы использования внешнего оборудования, сервисных устройств и т.п., и управляет одновременным решением нескольких независимых задач. Серийные модели ЭЦВМ БЭСМ-6 идентичны по аппаратному обеспечению, но фактически с точки зрения программиста существует столько различных машин БЭСМ-6, сколько функционирует несовместимых систем математического обеспечения. Мы будем ориентироваться на математическое обеспечение машины БЭСМ-6, установленной в ВЦ СО АН СССР (г. Новосибирск).

(а). ПАМЯТЬ. Память БЭСМ-6 включает в себя оперативное запоминающее устройство (ОЗУ), внешние запоминающие устройства (ВЗУ) и доступные программисту регистры центрального процессора.

ОЗУ машины БЭСМ-6 состоит из 100000_8 48- разрядных ячеек с пятизначными восьмеричными адресами от 00000 до 77777. Разряды в ячейке нумеруются справа налево в десятичной системе и имеют номера от I-го до 48-го. Каждая ячейка имеет еще два контрольных разряда, в которых записывается признак команды или операнда (числа). Выборка в качестве команды кода с признаком операнда вызывает в центральном процессоре сигнал ошибки в программе и прерывает счет по программе математика. С другой стороны, содержание любой ячейки независимо от значений контрольных разрядов может быть использовано в качестве операнда. Контрольные разряды недоступны программе математика кроме как через посредство системы математического обеспечения.

ОЗУ делится на 40_8 листов по 2000_8 ячеек в каждом. Лист в свою очередь разбивается на четыре абзаца по 400_8 ячеек в каждом абзаце. Каждый лист (соответственно абзац) начинается ячейкой с адресом, кратным 2000_8 (соответственно 400_8).

ВЗУ машины БЭСМ-6 состоят из магнитных барабанов (МБ) и магнитных лент (МЛ). Всего может быть подключено до 16_{10} барабанов и до 32_{10} магнитофонов. Все барабаны и магнитофоны делятся на шесть направлений, в том числе два барабанных направления с номерами 1 и 2 и четыре направления для магнитофонов с номерами 3,4,5 и 6. К каждому направлению подключается до восьми МБ или МЛ. Обмен информацией между ВЗУ и ОЗУ может происходить с одновременным участием до шести МБ и МЛ, расположенных на разных направлениях.

На каждом МБ имеется 100000_8 48-разрядных (плюс два контрольных разряда) ячеек, т.е. столько же, сколько во всем ОЗУ. Ячейки МБ группируются в 40_8 трактов по 2000_8 ячеек в каждом. Тракт разбивается на четыре сектора по 400_8 ячеек в каждом секторе. Нам будет удобно считать, что память на всех МБ представляет собой единый массив ячеек с 20-разрядными адресами от 1000000 до 2777777, в которых два старших разряда указывают направление, следующие три - номер МБ, следующие пять - номер тракта, следующие два - номер сектора и последние восемь разрядов - номер ячейки в секторе.

Каждая МЛ содержит 1000_8 зон. В каждой зоне записывается 2000_8 48-разрядных (не считая контрольных разрядов) кодов. Всего на одну МЛ помещается 2000000_8 таких кодов.

Аппаратное обеспечение БЭСМ-6 допускает только групповой обмен между ОЗУ и ВЗУ и только следующих трех видов: обмен между листом ОЗУ и зоной МЛ, обмен между листом ОЗУ и трактом МБ, обмен между абзацем ОЗУ и сектором МБ.

Из регистров центрального процессора интерес для программиста представляет 48-разрядный сумматор, 48-разрядный регистр младших разрядов (РМР), 6-разрядный регистр режимов работы арифметического устройства (РАУ) и 15_{10}-разрядные индексные регистры (ИР) с восьмеричными номерами 00, 01,..., 17. Роль каждого из этих регистров будет описана ниже.

Программист всегда должен помнить, что в ИР с номером 00 и в ячейке ОЗУ с адресом 00000 всегда записаны чисто нулевые коды.

(*в*) ПРЕДСТАВЛЕНИЕ ЧИСЕЛ. 48-разрядный машинный код вещественного числа x получается следующим образом. Если $x = 0$, то оно изображается кодом, все разряды которого равны нулю. Число $x \neq 0$ сначала представляется в двоично-нормализованной форме

$$x = sign\, x \cdot 2^{p} \cdot q ,$$

где $sign$ – знак, p– целый двоичный порядок, q – мантисса числа x, причем $\frac{1}{2} \le q < 1$ для $x > 0$ и $\frac{1}{2} < q \le 1$ для $x < 0$. В случае $p < -100_8$ число x отождествляется с нулем. В случае $p > 77_8$ число x не помещается в разрядную сетку ячейки БЭСМ–6 и не может быть введено в машину. Если подобное число возникает при арифметической операции, то выполнение программы математика прекращается. В случае $-100_8 \le p \le 77_8$ число x представимо в виде машинного кода БЭСМ–6. Знак такого числа заносится в 47–й разряд (0 означает +, 1 означает –). Мантисса q занимает 40 младших разрядов и пишется в прямом коде, если $x > 0$, и в дополнительном коде, если $x < 0$ (т.е. $1 - q$ в прямом коде). 7 старших разрядов изображают в прямом коде машинный порядок ($p_{маш} = p + 100$) числа x. Можно также считать, что 48–й разряд изображает знак порядка p (1 в случае $p \ge 0$, 0 и в случае $p < 0$), а модуль $|p|$ истинного порядка числа записывается в разрядах 47 ÷ 42 в прямом коде, если $p \ge 0$, и в дополнительном коде, если $p < 0$.

Например, машинные коды чисел 0.75, 5.125, 0.0625, – 0.75, – 5.125, – 0.0625, – 4 соответствуют следующим 16–ти значным восьмеричным константам:

0.75	– 4014	0000	0000	0000
5.125	– 4152	0000	0000	0000
0.0625	– 3650	0000	0000	0000
– 0.75	–4024	0000	0000	0000
– 5.125	– 4165	6000	0000	0000
– 0.0625	– 3620	0000	0000	0000
– 4	– 4120	0000	0000	0000

(c) СТРУКТУРА КОМАНД БЭСМ–6. Каждая команда машинного языка БЭСМ–6 является 24–разрядным кодом. В одной ячейке ОЗУ размещается две команды, из них левая занимает разряды 48 ÷ 25, а правая – остальные. Порядок выполнения команд естественный: после выполнения команд ячейки x обычно выполняются команды ячейки $x + 1$ и т.д., причем из двух команд любой ячейки первой выполняется всегда левая команда. Передача управления всегда осуществляется на левую команду ячейки. В частности, если левая команда некоторой ячейки есть команда безусловной передачи управления, то правая команда этой ячейки работать никогда не будет.

Во всякой команде старшие 4 разряда (24 + 2I) указывают номер i, $00 \leq i \leq I7_8$, индексного регистра, участвующего в формировании исполнительного адреса команды. На бланке эти разряды изображаются двумя восьмеричными цифрами, из которых старшая равна 0 или I. Следующий 20-й разряд является признаком типа команды. Если он равен 0, то команда относится к первому типу. Иначе мы имеем команды второго типа. В командах первого типа разряды 20 + I3 образуют восьмиразрядный код операции (КОП). На бланке КОП в командах первого типа изображается трехзначным восьмеричным числом, старшая цифра которого равна 0 или I. Собственно операцию, задаваемую командой первого типа, определяют разряды I8+I3. Разряд I9 в указании операции не участвует. Его роль будет описана ниже. Младшие разряды I2+I команды первого типа образуют ее адресную часть А. На бланке она изображается четырьмя восьмеричными цифрами. Итак, каждая команда первого типа имеет вид i КОП А, где $00 \leq i \leq I7$, $000 \leq$ КОП $\leq I77$, $0000 \leq A \leq 7777$.

Команда второго типа также имеет вид i КОП А, однако адресная часть А в этом случае занимает I5 младших разрядов и на бланке изображается пятью восьмеричными цифрами, а КОП – только пять разрядов (20+I6) и на бланке записывается двузначным восьмеричным числом, старшая цифра которого равна 2 или 3. Таким образом, в командах второго типа $00 \leq i \leq I7$, $20 \leq$ КОП ≤ 37, $00000 \leq A \leq 77777$.

Отметим еще, что ради удобства команды из одной ячейки обычно пишутся в две строки. В первой строке располагается левая команда, а во второй – правая. Примером программы для БЭСМ-6 может служить следующая последовательность кодов:

```
00032  К  00   040   0014
          00   013   0126
00033  К  15    26   00000
          00   010   0264
00034  К  14   036   0060
          00   011   0141
00035  К  15    30   00000
          00   000   0000
```

Здесь К означает, что в контрольные разряды нужно занести признак команды. Крайний левый столбец образуют адреса ячеек, содержащих командные коды, которые указаны в правом столбце. Среди этих команд две имеют второй тип, а остальные шесть – первый тип.

(d) ИСПОЛНИТЕЛЬНЫЙ АДРЕС КОМАНДЫ. Операнд, над которым должна произвести действие команда, как правило определяется динамически. В связи с этим мы будем различать действительный, модифицированный и исполнительный адреса команды. Действительный адрес $A_д$ команды i КОП А второго типа совпадает с ее адресной частью А. Действительный адрес $A_д$ команды i КОП А первого типа также является пятизначным восьмеричным числом. Его младшие четыре цифры совпадают с цифрами адресной части А, а старшая цифра равна 0, если I9-й разряд x команды нулевой, и 7, если x = I. Таким образом, в случае команды первого типа $A_д = 70000 \cdot x + A$.

Если предыдущая команда (по времени работы) имела КОП, отличный от 22 и от 23, то модифицированный адрес $A_{мод}$ команды i КОП А (любого типа) совпадает с ее действительным адресом. В противном случае модифицированный адрес команды i КОП А определяется равенством

$$A_{мод} = \{A_д + B\} \bmod 2^{I5},$$

где В совпадает с исполнительным адресом предыдущей команды, если она имела КОП 22, и с кодом, хранящимся в I5-ти младших разрядах ячейки, адрес которой является исполнительным адресом предыдущей команды, если она имела КОП 23.

Исполнительный адрес $A_{исп}$ команды i КОП А вычисляется по правилу

$$A_{исп} = (A_{мод} + И_i) \bmod 2^{I5},$$

где $И_i$ есть содержимое ИР с номером i (напомним, что $И_i$ = 00000, если i = 0).

Необходимо отметить, что в БЭСМ-6 широко используются команды, допускающие работу в так называемом "режиме магазина". При этом помимо специальных команд магазинного типа в указанном режиме может оказаться любая команда, использующая операнд из оперативной памяти и I7-й ИР для индексирования своей адресной части. Магазинный принцип работы мы обсудим позднее, а пока будем предполагать, что либо $i \neq$ I7, либо $A_{мод} \neq$ 00000

(*е*) ОСНОВНЫЕ СВЕДЕНИЯ О РАБОТЕ АРИФМЕТИЧЕСКОГО УСТРОЙСТВА. Обычно предполагается, что арифметические операции производятся над нормализованными числами, однако многие из них дают верный результат и при ненормализованных операндах. Мантисса результата арифметической операции обычно занимает 80 разрядов, в том числе 40 младших разрядов сумматора и 40 младших разрядов РМР. Если машинный порядок результата оказываается отрицательным (т.е. истинный порядок меньше, чем -64_{10}), то результат отождествляется с нулем и соответственно этому обнуляются все разряды сумматора и РМР. Если после нормализации результата оказывается, что его машинный порядок больше, чем 177_8 (истинный порядок больше, чем 63_{10}), то система математического обеспечения прерывает счет по программе математика. Этот вид прерываний может быть заблокирован. Для этого нужно записать I в шестой разряд регистра РАУ. В этом случае прерывания не происходит, а единица переноса из старшего разряда порядка игнорируется. Порядок результата арифметической операции заносится в старшие разряды сумматора.

Результат арифметической операции на внутренних регистрах машины может оказаться ненормализованным. В связи с этим предусмотрена возможность нормализации результата, которая проводится в два этапа: сначала результат нормализуется влево и затем округляется. Процесс нормализации результата влево может быть заблокирован математиком. Для этого нужно записать I в первый разряд регистра РАУ. Процесс округления осуществляется приформированием I к младшему разряду сумматора в следующих двух случаях: а) нормализация влево заблокирована и в 40 младших разрядах 80-разрядной мантиссы результата есть хотя бы одна единица, б) после нормализации влево, если до нее в 40 младших разрядах 80-разрядной мантиссы результата была хотя бы одна единица и в процессе нормализации влево ни одна единица не перешла в старшие 40 разрядов 80-разрядной мантиссы результата. В операции деления округление никогда не производится. Процесс округления может быть заблокирован. Для этого нужно записать I во второй разряд регистра РАУ.

Для реализации команд условного перехода все арифметические

и многие иные операции управляют выдачей сигнала ω . Способ вычисления значения сигнала ω (равного 0 или I) в любой момент времени определяется содержимым третьего, четвертого и пятого разрядов регистра РАУ. Единица в третьем разряде определяет логическое ω . В этом случае равенство ω = 0 равносильно равенству нулю содержимого сумматора. Единица в четвертом разряде определяет ω типа умножение. В этом случае ω = 0 тогда и только тогда, когда 48-й разряд сумматора содержит I (если на сумматоре записано нормализованное число x , то в этом случае $x < -\frac{1}{2}$ или $x \geqslant \frac{1}{2}$). Единица в пятом разряде РАУ определяет ω типа сложения. В этом случае ω = 0 тогда и только тогда, когда число, записанное в сумматоре, неотрицательно (точнее говоря, когда 4I - й разряд сумматора содержит 0). Если в разрядах 5 + 3 регистра РАУ имеются две или три единицы, то алгоритм выдачи сигнала ω определяется старшей единицей. Если же во всех этих разрядах содержатся нули, то вырабатывается ω = I.

В системе команд машины БЭСМ-6 две команды специально предназначены для присвоения любых удобных программисту значений разрядам регистра РАУ. Все остальные команды либо оставляют регистр РАУ без изменения, либо заносят в один из его разрядов 5,4 и 3 единицу и обнуляют другие два.

(*f*) АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОПЕРАЦИИ. Перечень всех арифметических и основных логических операций машины БЭСМ-6 приведен в таблице 2. При этом использованы следующие обозначения: C , τ, a - коды (числа), хранящиеся в сумматоре, РМР и ячейке $A_{исп}$ ОЗУ соответственно в момент начала операции, если эти буквы не расположены левее знака присвоения :=, и после операции в противном случае; p_c и p_a суть машинные порядки чисел C и a; $A^7_{исп}$ - код, образованный семью младшими разрядами исполнительного адреса $A_{исп}$; $\wedge$, $\vee$, $\sim$, $\oplus$ суть знаки операции поразрядного логического умножения, поразрядного логического сложения, поразрядного сравнения (= поразрядного сложения по модулю 2) и циклического сложения 48-разрядных кодов. Графы "Сумматор", "РМР [40 : I] ", РМР [48 + 4I] " содержат указания о том, что заносится в результате выполнения операции в сумматор, в младшие разряды РМР и в старшие разряды РМР. Знак + в графе "Норм" указывает, что результат операции нормализуется и округляется, если только нет соот-

КОП	Название Сумматор		РМР 40 + I	РМР 48 + 4I	Норм.	Тип ω
004	Сложение	$C + a$		нуль	+	слож.
005	Вычитание	$C - a$		нуль	+	слож.
006	Обратное вычитание	$a - C$		нуль	+	слож.
007	Вычитание модулей	$\|C\| - \|a\|$		нуль	+	слож.
017	Умножение	$C \times a$		нуль	+	умн.
016	Деление	Если a нормализовано, то C/a. Иначе прерывание	неопределено		+	умн.
014	Изменение знака	Если в коде a 4I-й разряд равен I, то $-C$, иначе $+C$	*нуль*	нуль	+	слож.
024	Сложение порядка с порядком	$p_c := p_c + p_a - 100_8$; знак и мантисса C сохраняются.	*нуль*	нуль	+	умн.
034	Сложение порядка с адресом	$p_c := p_c \pm A^7_{исп} - 100_8$; знак и мантисса C сохраняются	*нуль*	нуль	±	умн.
025	Вычитание порядка из порядка	$p_c := p_c - p_a + 100_8$; знак и мантисса C сохраняются	*нуль*	нуль	+	умн.
035	Вычитание адреса из порядка	$p_c := p_c - A^7_{исп} + 100_8$; знак и мантисса C сохраняются	*нуль*	нуль	+	умн.
011	Логическое умножение	$C \wedge a$	*нуль*	нуль	–	лог.
015	Логическое сложение	$C \vee a$	*нуль*	нуль	–	лог.
012	Поразрядное сравнение	$C \sim a$	C	C	–	лог.
013	Циклическое сложение	$C \oplus a$	*нуль*	нуль	–	умн.

ветствующих блокировок в регистре РАУ. В графе "Тип ω" отмечено, по какому способу вычисляется значение сигнала ω непосредственно после выполнения операции. Все арифметические и логические операции допускают работу в режиме магазина.

(9) СПЕЦИАЛЬНЫЕ ОПЕРАЦИИ НАД КОДАМИ. К этой группе операций машины БЭСМ-6 мы отнесем команды сдвига (026, 036) и операций над шкалами: (020, 021, 022, 023).

Операции сдвига сначала обнуляют РМР. Затем производится сдвиг кода на сумматоре. Величина и направление сдвига определяются семиразрядной константой сдвига, имеющей вид $100_8 + K$, где $-77_8 \leq K \leq 77_8$. В случае команды 026 константа сдвига совпадает с машинным порядком числа a из ячейки $A_{исп}$. В случае команды 036 константой сдвига служит код $A^7_{исп}$.

При $K \geq 0$ (соответственно при $K \leq 0$) содержимое сумматора сдвигается на $|K|$ разрядов вправо (влево). Код, выдвигаемый из младших (старших) разрядов сумматора, вдвигается в РМР со стороны его старших (младших) разрядов. Код, выдвигаемый из РМР со стороны его младших (старших) разрядов, исчезает. Освобождаемые слева (справа) разряды сумматора обнуляются.

Команда 020 осуществляет сборку содержимого сумматора С по маске. Маской служит код a из ячейки $A_{исп}$. По данной команде из сумматора выбираются и последовательно упаковываются разряды, которым соответствуют разряды маски, равные 1. Полученный K-разрядный код, где K – количество единиц в маске a, размещается в старших разрядах сумматора. Остальные разряды сумматора и все разряды РМР обнуляются.

Например, в случае

a = 0077 7700 7007 0556
C = 6646 3276 5315 2253

результатом сборки является код

4632 5515 0000 0000

Команда 021 осуществляет разборку содержимого сумматора С по маске a из ячейки $A_{исп}$. Результатом этой операции является 48-разрядный код, в котором последовательность K старших разрядов кода С, где K – количество единиц в коде a, размещена в тех разрядах, которым соответствуют разряды маски a, равные 1. Остальные разряды результата полагаются равными нулю. Результат записывается в сумматор, РМР обнуляется.

например, в случае

a = 0007 0077 0777 0257

C = I234 5676 5432 I000

результатом разборки будет код

0001 0023 0456 0256

Команда 022 реализует подсчет количества равных единице разрядов содержимого сумматора С. Сначала формируется промежуточный 48-разрядный код

0000 0000 0000 00K, (I)

в младших разрядах которого в виде целого двоичного числа указывается количество K единиц в коде С. Затем этот промежуточный код циклически складывается с 48-разрядным кодом a из ячейки $A_{исп}$. Результат заносится в сумматор. РМР обнуляется.

Команда 023 позволяет определить место старшего единичного разряда в коде С. Команда выполняется в два этапа. Сначала формируется промежуточный 48-разрядный код (I), где целое двоичное число K получается добавлением единицы к количеству нулевых разрядов, расположенных в коде С перед его старшей единицей. В частности K =I, если 48-й разряд кода С равен I; K =2, если старшая единица кода С занимает 47-й разряд и т.д. Если один младший разряд кода С равен I, а все остальные разряды - нулю, то $K = 60_8$. То же самое значение K формируется и в том случае, когда в сумматоре стоит чисто нулевой код. На втором этапе промежуточный код (I) циклически складывается с 48-разрядным кодом a из ячейки $A_{исп}$. Результат заносится в сумматор. В РМР поступает результат сдвига исходного кода С на K разрядов влево. Младшие K разрядов РМР обнуляются.

Все шесть команд 026, 036, 020, 021, 022, 023 заносят единицу в 3-й разряд РАУ и обнуляют 4-й и 5-й разряды, т.е. вырабатывают сигнал ω по тиду логических команд. За исключением команды 036 все остальные команды этой группы допускают работу в режиме магазина.

(h) КОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ. В системе команд машины БЭСМ-6 имеются семь команд перехода, из них - две команды безусловного перехода и пять команд передачи управления в зависимости от значений ИР с номером i (записанным в четырех старших разрядах команды перехода i КОП А) или сигнала ω. Все команды перехода являются командами второго типа. Информация о их работе

содержится в таблице 3. (Обозначения: $И_i$ – содержимое ИР с номером i; x – адрес ячейки ОЗУ, из которой в центральный процессор поступила рассматриваемая команда перехода; τ и С имеют тот же смысл, что и в таблице 2). Никаких иных действий кроме тех, что указаны в этой таблице, команды перехода не совершают. В частности, все команды перехода сохраняют содержимое сумматора и регистра РАУ (а следовательно, и значение сигнала ω).

код	Условие перехода	Куда передается управление	Дополнительные действия
26	$\omega = 0$	$А_{исп}$	$\tau := С$
27	$\omega = 1$	$А_{исп}$	$\tau := С$
30	всегда	$А_{исп}$	–
31	всегда	$А_{мод}$	$И_i := x + 1$
34	$И_i = 0$	$А_{мод}$	–
35	$И_i \neq 0$	$А_{мод}$	–
37	$И_i \neq 0$	$А_{мод}$	если $И_i \neq 0$, то $И_i := И_i + 1$

Таблица 3

Две команды перехода 31 и 37 особенно полезны. Команда 31 называется командой перехода с возвратом. Она существенно облегчает организацию обращения к подпрограммам. Команда 37 позволяет оформлять циклы в программе, повторяющиеся известное количество раз. Ее называют часто командой конца цикла.

При использовании команд перехода необходимо иметь в виду следующее. Эти команды передают управление всегда на левую команду в ячейке $А_{исп}$ или $А_{мод}$. На правую команду какой-либо ячейки, минуя ее левую команду, перейти невозможно. Если в ячейке x левая команда есть команда безусловного перехода (30 или 31), то правая команда этой ячейки никогда работать не будет. Если же левая команда была командой условного перехода (26, 27, 34, 35 или 37), то в случае невыполнения условия перехода будет работать правая команда ячейки x.

Работа команд перехода в режиме магазина невозможна.

(i) ОПЕРАЦИИ С РЕГИСТРАМИ. Ряд команд в машине БЭСМ-6 специально предназначен для работы с основными регистрами центрального процессора. Информация о работе этих команд содержится в таблице 4.

код	Название изменяемого регистра	Код, поступающий в этот регистр	Тип ω
000	Ячейка $A_{исп}$ в ОЗУ	С с признаком операнда	сохр.
010	Сумматор	Код a из ячейки ОЗУ с адресом $A_{исп}$	лог.
027	РАУ	Разряды 47÷42 кода a из ячейки $A_{исп}$	?
037	РАУ	$A^6_{исп}$	?
030	Сумматор	В разрядах 47÷ 42 результат поразрядного логического умножения содержимого РАУ с кодом $A^6_{исп}$. Остальные разряды обнуляются	сохр.
031	Сумматор	Если 3-й разряд РАУ содержит I, то τ. Иначе $p_c := p_c + A^1_{исп} - 100_8$, 41-й разряд :=0, $q_c := q_c \tau$ (в этом случае результат нормализуется)	сохр.
040	ИР с номером $A^4_{исп}$	I5 младших разрядов кода С	сохр.
042	Сумматор	Содержимое ИР с номером $A^4_{исп}$, дополненное слева нулями.	лог.
044	ИР с номером $A^4_{мод}$	Содержимое ИР с номером i	сохр.
045	ИР с номером $A^4_{мод}$	Сумма кодов из ИР с номерами i и $A^4_{исп}$ (перенос из старшего разряда игнорируется)	сохр.
24	ИР с номером i	Модифицированный адрес $A_{мод}$ команды	сохр.
25	ИР с номером i	Исполнительный адрес $A_{исп}$ команды	сохр.
22	В команде 22 адрес $A_{исп}$, а в команде 23 – код из I5-ти младших разрядов ячейки $A_{исп}$ определяют модифицированный адрес следующей команды.		сохр.
23			сохр.

Таблица 4

(Обозначения: $A^4_{исп}$ и $A^4_{мод}$ суть коды, образованные четырьмя младшими разрядами адресов $A_{исп}$ и $A_{мод}$; $A^6_{исп}$ – код, образованный шестью младшими разрядами адреса $A_{исп}$; q_c и q_τ – мантиссы 48-разрядных кодов С и τ; остальные обозначения здесь те же, что и в таблицах 2 и 3) Среди этих команды две команды 000 и 010 осуществляют обмен кодов между ячейками ОЗУ и сумматором; команды 030, 031 и 042 выдают в сумматор содержимое регистров РАУ, РМР и любого ИР; команды 027 и 037

присваивают разрядам регистра РАУ угодные программисту значения; команды 040, 044, 045, 24 и 25 обеспечивают широкие возможности изменения значений ИР с номерами 01÷17. Команды 22 и 23, условно включенные в данную группу команд, фактически уже описаны в пункте (d). Никаких иных действий, кроме указанных в таблице, операции рассматриваемой группы не производят. В частности все они сохраняют РМР. Сумматор может измениться лишь под действием команд 010, 030, 031 и 042. данной группы. Значение сигнала w может измениться лишь под действием команд 010, 027, 030, 031, 037, 042. Режим магазина допустим для команд 000, 010, 23, 027.

(j) О МАГАЗИННОЙ ОРГАНИЗАЦИИ ПАМЯТИ. Под магазином в машине БЭСМ-6 понимается произвольная последовательность подряд расположенных ячеек ОЗУ, содержащая ячейку с адресом $И_{17}$ (= содержимому 17-го ИР). Предполагается, что ячейки магазина с адресами $A < И_{17}$ заняты, а с адресами $A \geq И_{17}$ свободны. Таким образом, содержимое 17-го ИР, именуемого также счетчиком магазина, всегда указывает на первую свободную ячейку магазина. При записи в магазин производится заполнение ячейки с адресом $И_{17}$, после чего содержимое счетчика магазина увеличивается на единицу. При чтении кода из магазина в сумматор поступает содержимое последней занятой ячейки, т.е. ячейки с адресом $И_{17} - 1$; после этого данная ячейка считается свободной ячейкой магазина, т.е. значение $И_{17}$ счетчика магазина уменьшается на единицу.

Признаком того, что данная команда i КОП A, допускающая работу в режиме магазина, действительно работает в этом режиме, является одновременное выполнение двух равенств $i = 17$ и $A_{мод} = 00000$. (Если хотя бы одно из равенств не выполнено, то команда работает в обычном режиме; ее операндом в этом случае служит код a из ячейки $A_{исп}$). Работа команды 000 в режиме магазина заключается (согласно описанному выше принципу магазинной организации памяти) в том, что код C из сумматора переписывается в ячейку ОЗУ с адресом $И_{17}$ и значение $И_{17}$ автоматически увеличивается на единицу. Все остальные команды в режиме магазина считывают код из магазина (автоматически уменьшая значение $И_{17}$) и используют его в качестве одного из операндов (того, который мы ранее обозначали через a). Например, команда 17 015 0000 (перед которой нет команды с КОПом 22 или 23) сначала уменьшит значение $И_{17}$ на единицу и затем произведет логическое сложение кода C с содержимым ячейки, адрес которой равняется новому значению счетчика магазина.

(к) СПЕЦИАЛЬНЫЕ МАГАЗИННЫЕ КОМАНДЫ. В дополнение к тому, что многие команды позволяют использовать операнды в соответствии с магазинным принципом организации памяти, в машине БЭСМ-6 предусмотрены четыре команды (001, 003, 041, 043) всегда работающие в магазинном режиме.

Команда 001 сначала записывает (с признаком операнда) содержимое сумматора в ячейку $А_{исп}$ (исполнительный адрес определяется по общему правилу, даже если команда использует I7-й ИР). Затем $И_{I7}$ уменьшается на единицу и код из ячейки с адресом $И_{I7}$ поступает в сумматор. В частности, команда I7 101 7777 не изменяет содержимого сумматора.

Команда 003 записывает код С из сумматора в ячейку с адресом $И_{I7}$ (с признаком операнда), затем увеличивает $И_{I7}$ на единицу и вызывает в сумматор код из ячейки $А_{исп}$ (если i =I7, то исполнительный адрес формируется по общему правилу, но с использованием нового значения $И_{I7}$). В частности, команда I7 103 7777 не изменяет содержимого сумматора.

Команда 041 переписывает I5 младших разрядов сумматора в ИР с номером $А^{4}_{исп}$, затем уменьшает $И_{I7}$ на единицу и вызывает в сумматор код из ячейки $И_{I7}$. В частности, если в младших разрядах сумматора имеется код 00001, то команда 00 041 0017 обнулит сумматор и ИР с номером I7.

Команда 043 переписывает содержимое сумматора в ячейку $И_{I7}$ (с признаком операнда), увеличивает $И_{I7}$ на единицу и затем содержимое ИР с номером $А^{4}_{исп}$ (если i =I7, то $А_{исп}$ формируется с использованием нового значения $И_{I7}$) вызывает в I5 младших разрядов сумматора. Старшие разряды сумматора обнуляются.

Все четыре магазинные команды сохраняют РМР и определяют сигнал ω по типу логических команд.

(л) МАКРОКОМАНДЫ. Комплекс математического обеспечения машины БЭСМ-6 представляет собой естественное продолжение ее аппаратных возможностей. Он представляет в распоряжение программиста средства связи с внешним оборудованием ЭЦВМ (в частности, определяет способ обмена между ОЗУ и памятью на магнитных барабанах и лентах) и включает в себя трансляторы с алгоритмических языков, библиотеки стандартных и пакеты проблемно-ориентированных программ, сервисные услуги и т.д. Для использования программ комплекса математического

обеспечения в системе команд машины предусмотрена серия команд первого типа. (с КОПами 050 ÷ 077), специальным образом предназначенных для прерывания работы основной программы и перехода на организующую программу системы математического обеспечения с передачей ей параметров обращения. Действия, совершаемые командами 050÷077 (эти команды именуются макрокомандами), осуществляются не электронными схемами, а специальными подпрограммами. Различными комплексами математического обеспечения макрокоманды могут трактоваться по-разному. Этот факт делает любую программу, написанную в машинных командах, привязанной к определенному классу взаимосовместимых систем математического обеспечения.

Программист должен учитывать, что независимо от системы математического обеспечения, при выполнении любой макрокоманды значение ИР с номером I6 теряется безвозвратно. Далее, всякая макрокоманда есть фактически команда безусловного перехода на некоторую подпрограмму системы математического обеспечения. Значит, если она занимает левую половину ячейки, то правая команда этой ячейки пропускается.

Авто-Аналитик реализован на машине БЭСМ-6, установленной в ВЦ СО АН СССР, и, следовательно, привязан к системе математического обеспечения, имеющейся в этой машине. Из числа соответствующих макрокоманд мы здесь отметим лишь одну, имеющую КОП 075. Макрокоманда 075 переписывает содержимое сумматора в ячейку $A_{исп}$ оперативной памяти и снабжает его признаком команды. Без этой макрокоманды было бы невозможно использование формируемых команд.

Отметим, наконец, что некоторые команды (а именно, команды 002, 032, 033, 046, 047, 20, 2I, 32 и 33) относятся к числу "запрещенных" команд. Их применение в обычной программе математика вызывает прерывание счета по его программе.

Почти весь материал данного параграфа заимствован из работы [20]. Более подробное описание БЭСМ-6 имеется в книге [21].

§ 2. КОДИРОВАНИЕ ЭЛЕМЕНТОВ АЛФАВИТА АВТО-АНАЛИТИКА.

Для представления в памяти машины БЭСМ-6 произвольной формулы Авто-Аналитика каждый элементарный символ из его алфавита Г кодируется при помощи одного или нескольких I2-разрядных двоичных кодов или, что эквивалентно, при помощи одной или нескольких восьмеричных тетрад (=четырехзначных восьмеричных кодов). Если символы α_i, образующие формулу $\Delta\,\alpha_1\alpha_2\ldots\alpha_n\,\Delta$, имеют коды

$$Q_{i1}\,Q_{i2}\ldots Q_{ik_i},\quad i=1,2,\ldots,n,$$

состоящие из тетрад Q_{ij}, то кодом самой формулы объявляется последовательность тетрад

$$Q_0\,Q_{11}\,Q_{12}\ldots Q_{1k_1}\,Q_{21}\,Q_{22}\ldots Q_{2k_2}\ldots\ldots\ldots Q_{n1}\,Q_{n2}\ldots Q_{nk_n}\,Q_0 \qquad (2)$$

где тетрада Q_0 есть код граничного символа Δ.

В 48-разрядной ячейке памяти машины БЭСМ-6 можно разместить четыре восьмеричных тетрады. Первая из них всегда будет занимать старшие разряды 48÷37, вторая – разряды 36÷25, третья – разряды 24÷I3 и четвертая – младшие разряды I2÷I ячейки. Весь код (2) данной формулы занимает несколько подряд расположенных ячеек, причем первые четыре тетрады этого кода записываются в первую из отведенных ячеек, следующие четыре тетрады – во вторую и т.д. Более подробно вопрос о записи формул в памяти БЭСМ-6 будет обсуждаться в следующем параграфе. В данном параграфе мы опишем основные принципы кодирования элементарных символов Авто-Аналитика.

Напомним, что алфавитом Авто-Аналитика служит множество Г элементарных символов, включающее в себя множество Θ специальных символов, множество Σ связей, множество Ω постоянных букв, множество Φ переменных букв и множество R действительных чисел.

(а) КОДИРОВАНИЕ СПЕЦИАЛЬНЫХ СИМВОЛОВ. Граничный символ Δ, пустой символ $\odot$, раскрывающая скобка (и закрывающая скобка) имеют соответственно коды 0000, 0001, 0002 и 0003.

(*β*) КОДИРОВАНИЕ СВЯЗЕЙ. Для кодирования связей $\sigma\in\Sigma$ отводятся тетрады Q, удовлетворяющие условию $0004 \leqslant Q \leqslant 0777$. Каждая связь кодируется одной такой тетрадой. Установление конкретного соответствия между связями $\sigma\in\Sigma$ и отвечающими им тетрадами Q_σ предоставляется пользователю. Мы будем предполагать лишь, что это соответствие взаимно однозначно и сохраняет отношение порядка, т.е. более слабой связи сопоставлен меньший код. Кроме того, код Q_σ коммутативной или ассоциативной связи σ должен подчиняться неравенству $0020 \leqslant Q_\sigma \leqslant 0077$.

Эти два требования накладывают некоторые ограничения на множества Σ, Σ_0 и Σ_K. А именно, множество Σ не должно содержать более, чем 508, а его подмножество $\Sigma_0 \cup \Sigma_K$ – более, чем 48 элементов.

(c) КОДИРОВАНИЕ ПОСТОЯННЫХ БУКВ. Для кодирования постоянных букв $\lambda \in \Omega$ отводятся тетрады Q, подчиняющиеся неравенству $1000 \leq Q \leq 3577$. Кодом одной постоянной буквы может служить не только одинокая тетрада, но и любая конечная последовательность тетрад, удовлетворяющих данному неравенству. Естественно, соответствие между постоянными буквами и их вообще говоря групповыми кодами должно быть взаимно однозначным. Кроме того, это соответствие должно сохранять отношение порядка. А именно, если из двух постоянных букв λ и μ первая слабее второй, то код $Q_0 Q_1 \dots Q_K$ буквы λ должен быть слабее кода $Q_0' Q_1' \dots Q_\ell'$ буквы μ в лексикографическом смысле, т.е. должно выполняться одно из следующих двух условий:

1. Существует такой индекс i, $i \leq K$ и $i \leq \ell$, что $Q_0 = Q_0'$, $Q_1 = Q_1'$, ..., $Q_{i-1} = Q_{i-1}'$, $Q_i < Q_i'$;
2. $K < \ell$ и $Q_i = Q_i'$ для всех $i = 1, 2, \dots, K$.

Установление конкретного соответствия между постоянными буквами и их групповыми кодами предоставляется пользователю. При этом следует иметь в виду, что некоторые стандартные операторы Авто-Аналитика используют так называемые служебные постоянные буквы с фиксированной кодировкой. Коды, предназначенные для этих букв, нельзя сопоставлять иным постоянным буквам. Например, в задачах, использующих оператор обобщенной коммутативной подстановки, нельзя употреблять постоянную букву с кодом 1000, так как она является служебной постоянной буквой этого оператора. Кроме того, некоторые стандартные операторы накладывают ограничения сверху на длину (т.е. количество тетрад) кода постоянной буквы.

Два подряд расположенных кода постоянных букв можно спутать с одним групповым кодом третьей постоянной буквы. Например, тетрады 3576, 3577 и группа 3576 3577 из двух этих тетрад являются кодами трех различных постоянных букв. Тем не менее при дешифрации кода формулы, в котором коды соседних символов действительно ничем не разделены, такая путаница невозможна. В самом деле, согласно условиям (c) и (d) определения I главы I в формуле две буквы подряд стоять не могут. Поэтому, если в коде $Q_0 Q_1 \dots Q_m Q_0$

формулы Ш некоторая тетрада Q_K удовлетворяет условию 1000 $\leq Q_K \leq$ 3577, то эта тетрада входит в код постоянной буквы формулы Ш, причем полный групповой код этой буквы имеет вид

$$Q_i\, Q_{i+1} \ldots Q_{K-1}\, Q_K\, Q_{K+1} \ldots Q_j ,$$

где $1 \leq i \leq K \leq j \leq m$, $Q_{i-1} \leq$ 0777, $Q_{j+1} \leq$ 0777 и 1000 $\leq Q_\ell \leq$ 3577 для каждого $\ell = i, i+1, \ldots, j$. В частности, тетрада Q_K одна образует код постоянной буквы в формуле Ш тогда и только тогда, когда соседние тетрады Q_{K-1} и Q_{K+1} не превосходят 0777.

(d) КОДИРОВАНИЕ ПЕРЕМЕННЫХ БУКВ. Согласно § 10 главы I множество Φ переменных букв состоит из нескольких попарно не пересекающихся подмножеств $\Phi_0, \Phi_1, \ldots, \Phi_7$. В реализации Авто-Аналитика на машине БЭСМ-6 предполагается, что количество этих подмножеств равно восьми (т.е. $\nu = 7$) и в каждом из них содержится восемь переменных букв.

Тем самым в этой реализации запрещено применение правил, содержащих более восьми переменных букв одного типа. Впрочем данное ограничение по-видимому не слишком обременительно. Во всяком случае авторы не встречали еще ни одного правила в приложениях, содержащего более четырех переменных букв одного типа.

Для кодирования переменных букв отводятся тетрады 3600÷3677, причем тетрады 3600÷3607 закрепляются за типом Φ_0, тетрады 3610÷3617 – за типом Φ_1 и т.д. Каждая переменная буква кодируется одной тетрадой.

Следует однако иметь в виду, что некоторые модификации обобщенной подстановки предполагают несколько иное распределение тетрад 3600÷3677 между различными типами переменных букв. См. по этому поводу описания соответствующих операторов в главе IУ.

(e) КОДИРОВАНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ. Каждое вещественное число представляется в виде последовательности из четырех тетрад Q_1, Q_2, Q_3, Q_4, каждая из которых не меньше 4000, т.е. имеет 1 в своем старшем 12-м разряде. Правило образования кода $Q_1\, Q_2\, Q_3\, Q_4$ числа z следующее. Сначала образуется 48-разрядный нормализованный машинный код

$$\mathfrak{z}_{48}\, \mathfrak{z}_{47} \ldots \mathfrak{z}_2\, \mathfrak{z}_1 \qquad (3)$$

числа z . Разборка этого кода (команда 021) по маске

$$3777\ \ 3777\ \ 3777\ \ 3777 \qquad (4)$$

и последующее логическое сложение с константой

4000 4000 4000 4000

приводит его к виду

$$1\,\xi_{47}\,\xi_{46}\ldots\xi_{37}\,1\,\xi_{36}\ldots\xi_{27}\,1\,\xi_{26}\,\xi_{25}\ldots\xi_{16}\,1\,\xi_{15}\,\xi_{14}\ldots\xi_{5}$$

Старшие I2 **разрядов полученного кода образуют тетраду** Q_1 **, следующие** I2 **разрядов – тетраду** Q_2 **и т.д.**

Ниже приведены примеры на тему кодирования чисел. В левом столбце указаны числа в I0-й системе счисления, в среднем столбце – I6-ти значные восьмеричные константы, отвечающие их 48-разрядным двоичным кодам, в правом столбце – соответствующие четырехтетрадные коды Авто-Аналитика.

0	0000	0000	0000	0000	4000	4000	4000	4000
0,25	3750	0000	0000	0000	5764	4000	4000	4000
0,5	4010	0000	0000	0000	6004	4000	4000	4000
I	4050	0000	0000	0000	6024	4000	4000	4000
2	4110	0000	0000	0000	6044	4000	4000	4000
I,5	4054	0000	0000	0000	6026	4000	4000	4000
3	4114	0000	0000	0000	6046	4000	4000	4000
0,I	3654	6314	6314	6314	5726	5463	4631	6314
0,2	3714	6314	6314	6314	5746	5463	4631	6314
$\sqrt{2}$	4053	2404	7463	1770	6025	6501	4746	5477
π	4114	4417	6652	1041	6046	5103	7665	5042
-0,25	3720	0000	0000	0000	5750	4000	4000	4000
-0,5	3760	0000	0000	0000	5770	4000	4000	4000
-I	4020	0000	0000	0000	6010	4000	4000	4000
-2	4060	0000	0000	0000	6030	4000	4000	4000
-I,5	4064	0000	0000	0000	6032	4000	4000	4000
-3	4124	0000	0000	0000	6052	4000	4000	4000
-0,I	3663	1463	1463	1464	5731	6314	7146	5463
-I/3	3765	2525	2525	2526	5772	6525	5252	6525
-2^{-64}	0020	0000	0000	0000	4010	4000	4000	4000

Для дешифрации кода $\theta_1\,\theta_2\,\theta_3\,\theta_4$ числа z достаточно занести тетрады кода в сумматор и подвергнуть сборке (команда 020) по маске (4). Четыре младших разряда полученного результата будут равны нулю, а 44 старших разряда – те же, что и в машинном коде (3) числа z. Таким образом, наш способ кодирования вещественных чисел является приближенным даже относительно машинного кода (3). Точное представление в виде четы-

рехтетрадного кода допускают лишь те числа, у которых двоичная мантисса содержит не более 36_{10}-ти разрядов, а истинный двоичный порядок p удовлетворяет условию $-64_{10} \leq p \leq 63_{10}$. В частности, таковы все целые числа z, $|z| < 2^{63}$. При $|z| \leq 2^{-65}$ число z отождествляется с нулем и кодируется тетрадами 4000 4000 4000 4000

При $|z| > 2^{63}$ оно не может быть представлено четырехтетрадными кодом даже приближенно. В этом случае, а также во всех тех случаях, когда число нужно выразить точно, а четырехтетрадный код точным не является, приходится обозначать данное число постоянной буквой (например, π, e) или записывать его в виде некоторой подформулы (например, LN*2, 9↑999).

Одним из недостатков описанного способа кодирования чисел является несогласованность естественного порядка между числами и лексикографического порядка между четырехтетрадными числовыми кодами. Тем не менее во всех вопросах, связанных с отношением порядка между элементами алфавита Г, (и, в частности, в операторе коммутирования формулы) мы считаем, что отношение порядка между элементарными символами переходит в лексикографическое отношение порядка между их однотетрадными или групповыми кодами. Тем самым мы предполагаем, что во множестве вещественных чисел задан порядок, отличный от естественного. А именно, число z_1 считается слабее числа z_2, если 44-разрядное целое двоичное число, образованное старшими разрядами 48÷5 машинного кода числа z_1, меньше 44-разрядного целого двоичного числа, полученного аналогичным образом для z_2. Если числа z_1 и z_2 одного знака, то более слабым из них является то число, абсолютная величина которого меньше. Для чисел $z_1 < 0 \leq z_2$ слабее число с меньшим машинным пределом. В случае равенства порядков отрицательное число сильнее положительного.

Четырехтетрадный код вещественных чисел обязателен во всех формулах, которые могут быть подвергнуты обработке оператором Арифметик. В остальных случаях его можно модифицировать. В аналитических расчетах часто встречаются "небольшие" двоично-рациональные и, в частности, целые числа. Четырехтетрадные коды этих чисел отличаются друг отдруга лишь одной-двумя первыми тетрадами; последние тетрады часто совпадают с тетрадой 4000.

Без ущерба для понимания эти последние тетрады можно отбросить. Тогда, например, числа

0, I, 2, 4, $\frac{I}{2}$, $\frac{I}{4}$, $-I\frac{I}{2}$, $-I\frac{2I}{32}$, −63, 4095, 2^{-65}, -2^{-64}, $7329\frac{I}{64}$

будут иметь соответственно следующие коды

4000, 6024, 6044, 6064, 6004, 5764, 6032, 603I5400, 6I504400, 63077774, 4004, 40I0, 632745024I00

(ℓ) ЗАКЛЮЧЕНИЕ. Изложенный способ кодирования элементарных символов по сути дела почти полностью фиксирует алфавит Г в реализации Авто-Аналитика на БЭСМ-6. В самом деле, связями в этой реализации являются тетрады 0004÷0777 и только они. Постоянные буквы суть любые конечные упорядоченные наборы тетрад I000÷3577 и только они. Переменные буквы суть тетрады 3600÷3677 и только эти тетрады. Множество специальных символов и допустимых вещественных чисел также полностью фиксированы.

По воле пользователя могут в некоторых пределах варьироваться лишь множества Σ_a и Σ_κ. А именно, реализация Авто-Аналитика на БЭСМ-6 требует, чтобы код Q_σ любой связи $\sigma \in \Sigma_a \cup \Sigma_\kappa$ подчинялся условию 0020 ≤ Q_σ ≤ 0077, но не предполагает, что всякая тетрада, удовлетворяющая этому условию, является кодом ассоциативной или коммутативной связи. Для указания ассоциативных и коммутативных тетрад пользователь должен задать две 48-разрядные шкалы: шкалу ассоциативных связей в ячейке 00264 и шкалу коммутативных связей в ячейке 00263. Первый справа разряд обеих шкал сопоставлен тетраде 0020, второй разряд – тетраде 002I,..., 48-й разряд – тетраде 0077. Единичный разряд шкалы ассоциативных (коммутативных) связей означает, что тетраду, отвечающую этому разряду, пользователь считает ассоциативной (соотв. коммутативной) связью. Нулевое значение разряда соответствует неассоциативности (некоммутативности) этой связи.

Пользователь должен также по своему усмотрению устанавливать соответствие между его обозначениями элементарных символов и однотетрадными или многотетрадными кодами алфавита Г. Например, он может считать, что символ ϕ является связью и кодируется тетрадой 0642. Однако этого символа нет среди знаков УПП (устройства печати и перфорации). Следовательно, при подготовке программы решения задачи к перфорации пользователю придется самому символ ϕ и другие символы, отсутствующие среди знаков УПП, всюду заменить соответствующими тетрадами.

Что касается знаков УШ, то большинство из них закрепилось за вполне определенными тетрадами (и пользователю не рекомендуется менять это соответствие, так как оно существенно используется в таких важных операторах, как оператор Арифметик, оператор приведения подобных, операторы дифференцирования, интегрирования, решения задач линейной алгебры, исследования системы дифференциальных уравнений по методу Картана и т.п.). В частности, знаки УШ, отнесенные к связям, и соответствующие им тетрады указаны в таблице 5. Среди этих связей знаки +, x, $\vee$, $\wedge$ пользователь всегда должен относить к ассоциативным и коммутативным связям. Заглавные печатные буквы русского и латинского алфавитов естественно отнести к постоянным буквам. Их кодирование указано в таблице 6. Если пользователь в какой-либо формуле запишет подряд группу русских и латинских (заглавных) букв, то данная группа воспримется в качестве одной постоянной буквы из Ω с многотетрадным кодом, причем этот код будет составлен из тетрад, отвечающих входящим в данную группу русским и латинским буквам. Например, постоянные символы АЛЬФА и СОБ будут иметь соответственно коды

II00 III3 II33 II24 II00 и II2I III6 II50

Среди знаков УШ кроме тех, что указаны в таблицах 5 и 6, есть еще десятичные цифры 0,I,2,3,4,5,6,7,8,9 и символы

␣ ( ) [ ] ' , % ◊

Круглые скобки относятся к специальным символам и, какуказано выше, имеют коды 0002 и 0003. Кавычки ' и' используются для записи на бланке некоторых часто употребляемых связей, отсутствующих среди знаков УШ. Например, группа символов 'S' обозначает знак $\int$ интеграла (Выражение $\int F(t)dt$ на бланке принимает вид F*T 'S' T) и кодируется одной тетрадой 0I73. Группа символов 'P' обозначает знак дифференцирования (выражение dF/dt на бланке принимает вид F 'P' T) и кодируется тетрадой 0I75. Группа символов 'Д' обозначает знак дифференциала и кодируется тетрадой 0I76. Группа символов 'D' кодируется одной тетрадой 0I77.

Символ % используется на бланке в качестве признака переменной буквы. А именно, переменные буквы из подмножеств

$\Phi_0, \Phi_1, \ldots, \Phi_7$

обозначаются соответственно символами

%Xd , %Yd , %Zd , %Pd , %Qd , %Rd , %Sd , %Td ,

Знак УШ	Тетрада	Смысловое содержание знака или его наименование
10	0005	Подстрочная десятка
≠	0006	Знак неравенства
—	0007	тире или соединительный знак между левой и правой частями в правилах
=	00II	Знак равенства
<	00I2	Знак неравенства
>	00I3	Знак неравенства
≥	00I4	Знак неравенства
≤	00I5	Знак неравенства
:	00I6	Двоеточие
,	0020	Запятая
≡	0022	Знак логического тождества
+	0024	Знак операции сложения ("плюс")
×	0034	Знак операции умножения
/	0044	Знак операции деления
↑	0050	Знак операции возведения в степень
∨	0070	Знак дизъюнкции ("или")
∧	007I	Знак конъюнкции ("и")
;	0074	Точка с запятой или знак разделения аргументов функции
–	0I0I	Одноместный знак "минус"
¬	0I20	Знак отрицания ("не")
.	0I40	Точка
⊃	0204	Знак включения или импликации
!	02I0	
÷	02I4	
*	0774	Знак функциональной зависимости
‾	0775	Надчерк
_	0776	Подчерк

Таблица 5

Буква	Тетрада	Буква	Тетрада	Буква	Тетрада	Буква	Тетрада
А	II00	М	III4	Ш	II30	*L*	II44
Б	II0I	Н	III5	Щ	II3I	*N*	II45
В	II02	О	III6	Ы	II32	*Q*	II46
Г	II03	П	III7	Ь	II33	*R*	II47
Д	II04	Р	II20	Э	II34	*S*	II50
Е	II05	С	II2I	Ю	II35	*U*	II5I
Ж	II06	Т	II22	Я	II36	*V*	II52
З	II07	У	II23	*D*	II37	*W*	II53
И	III0	Ф	II24	*F*	II40	*Z*	II54
Й	IIII	Х	II25	*G*	II4I		
К	III2	Ц	II26	*I*	II42		
Л	III3	Ч	II27	*J*	II43		

Таблица 6.

где α есть одна из цифр 0, I,..., 7. При этом цифра $\alpha = 0$ может быть опущена. Соответствие между этими трехсимвольными обозначениями переменных букв и их однотетрадными кодами указано в таблице 7.

Назначение прочих символов УШ и дальнейшие сведения о подготовке исходной информации к вводу в машину см. в следующих главах.

Символ	Тетрада	Символ	Тетрада	Символ	Тетрада	Символ	Тетрада
% *X* или % *X0*	3600	% *Z* или % *Z0*	3620	% *Q* или % *Q0*	3640	% *S* или % *S0*	3660
% *X1*	360I	% *Z1*	362I	% *Q1*	364I	% *S1*	366I
...	...	...	...	...	...	...	...
% *X7*	3607	% *Z7*	3627	% *Q7*	3647	% *S7*	3667
% У или % У0	3610	%Р или % Р0	3630	% *R* или % *R* 0	3650	% Т или % Т0	3670
% У1	36II	% Р1	363I	% *R* I	365I	% Т1	367I
...	...	...	...	...	...	...	...
% У7	36I7	% Р7	3637	% *R* 7	3657	% Т7	3677

Таблица 7

§ 3. РАСПРЕДЕЛЕНИЕ ПАМЯТИ

(а) ВВЕДЕНИЕ. Поскольку машина Бэсм-6 предназначена для одновременного счета по нескольким независимым программам, то математик для своей задачи должен заказать (в так называемом паспорте задачи) необходимые ему листы ОЗУ, тракты МБ и номера МЛ. Система математического обеспечения БЭСМ-6 анализирует паспорта разных задач и распределяет ОЗУ и ВЗУ между ними так, что обеспечивается одновременное независимое решение нескольких задач даже в том случае, когда заказанные для них листы ОЗУ, тракты МБ и номера МЛ взаимно перекрываются. Программисту следует лишь иметь в виду, что любое обращение (чтение, запись, передача управления) к незаказанным частям памяти вызывает прерывание счета по его программе.

При использовании Авто-Аналитика пользователь должен заказать и предоставить в его распоряжение не менее четырех листов в начале ОЗУ (т.е. ячейки ОЗУ с адресами 00000÷$A^{\omega}_{ОЗУ}$, где $A^{\omega}_{ОЗУ} \geqslant$ 07777 и $A^{\omega}_{ОЗУ}$ + I делится без остатка на 2000), несколько первых трактов МБ (т.е. ячейки МБ с 20-разрядными адресами от I 000000 по $A^{\omega}_{МБ}$, где $A^{\omega}_{МБ}$ удовлетворяет условию

$$0\ 777777 \leqslant A^{\omega}_{МБ} \leqslant 2\ 777777$$

и $A^{\omega}_{МБ}$ + I без остатка делится на 2000), а также один магнитофон для МЛ, на которой находится весь комплекс программ Авто-Аналитика. Информация о заказанных листах ОЗУ и трактах МБ кроме паспорта задачи указывается еще в так называемых основных параметрах Авто-Аналитика, о которых пойдет речь ниже.

Предоставленная Авто-Аналитику часть ОЗУ делится на шесть частей: библиотека стандартных программ (БСП), оперативное рабочее поле (ОРП), оперативный адресный стол (ОАС), таблица переходов (ТП), специальное рабочее поле (СРП) и массив обмена с барабанами (МО). Память на МБ делится на две части: барабанный адресный стол (БАС) и барабанное рабочее поле (БРП).

(а) БИБЛИОТЕКА СТАНДАРТНЫХ ПРОГРАММ. БСП занимает фиксированное место в ОЗУ начиная с ячейки 00001 и кончая ячейкой $A^{o}_{ОРП}$ - I, $A^{o}_{ОРП} \geqslant$ 04400, и включает в себя программы, оформление которых в виде операторов (см. § 6) либо невозможно, либо существенно снижает скорость работы Авто-Аналитика. Основными

стандартными программами являются программы, связанные с принятым в Авто-Аналитике способом записи формул в ОЗУ ("Чтение", "Запись", "Прописка", "Выдача информации о формуле" и т.п.). В библиотеку стандартных программ включена программная реализация таких важных органов Авто-Аналитика, как система автоматического распределения памяти (Диспетчер) и интерпретирующая система Авто-Аналитика (Ридер). Описание Диспетчера, стандартных программ и Ридера содержится в следующих трех параграфах.

В БСП содержатся также массивы разнообразных восьмеричных констант, массивы основных параметров Авто-Аналитика и так называемых констант настроек. Ряд ячеек БСП используется в качестве рабочих ячеек Авто-Аналитика.

В конце БСП начиная с ячейки 04400 пользователь может размещать свои собственные стандартные программы, рабочие ячейки и константы. Если таковых у пользователя не окажется, то он должен адрес $A^{o}_{ОРП}$ установить равным 04400.

(с) ОПЕРАТИВНОЕ РАБОЧЕЕ ПОЛЕ. ОРП занимает часть ОЗУ начиная с ячейки $A^{o}_{ОРП}$ и кончая ячейкой $A^{\omega}_{ОРП} = A^{o}_{ОАС} - I$. Оно (совместно с БРП) предназначено для хранения всех формул, участвующих в решении задачи пользователя, в том числе исходных формул, формул, являющихся промежуточными или окончательными результатами счета, а также списки всех правил преобразования формул при помощи операторов простой и обобщенной подстановок. ОРП может заполняться как сначала (т.е. начиная с ячейки $A^{o}_{ОРП}$), так и с конца (т.е. с ячейки $A^{\omega}_{ОРП}$) и таким образом распадается на две части – ОРПI и ОРПII. Между этими частями расположен массив свободных ячеек, образующих резерв ОРП. Ячейки резерва ОРП присоединяются к ОРПI или ОРПII по мере записи в них новых формул. В момент времени, когда резерв ОРП исчерпается полностью, производится обращение к Диспетчеру для перераспределения всей памяти и обеспечения нового резерва, достаточного для дальнейшей работы Авто-Аналитика.

ОРП II считается вспомогательной частью ОРП. Сюда для кратковременного хранения заносятся промежуточные формулы, их всевозможные составные части и вообще любая информация, вырабатываемая для собственных целей некоторыми стандартными программами и операторами, и исчезающая из памяти к концу их работы. Структура и способ размещения информации в ОРП II целиком определяются внутренними целями и средствами соответствующих стандартных программ и операторов.

ОРП I является основной частью ОРП. Каждая формула здесь занимает одну или несколько подряд расположенных ячеек. В каждую ячейку пишется по четыре тетрады из кода формулы. Поэтому естественно "элементарной составной частью" ОРП считать группу из I2 разрядов произвольной ячейки. Такие группы мы будем именовать позициями. В каждой ячейке А выделяются четыре позиции с номерами $i = 0, I, 2$ и 3, образованные соответственно разрядами 48÷37, 36÷25, 24÷I3, I2÷I. Шестизначное восьмеричное число iА будет называться адресом i-ой позиции ячейки А. Все позиции ОРП располагаются в естественном линейном порядке. А именно, мы считаем, что за позицией с адресом 0А следует позиция IА, затем идут позиции 2А и 3А. После позиции 3А следует позиция ячейки А + I и т.д. Для всякой позиции iА адрес следующей позиции обозначается через iА + I, а адрес предыдущей – через iА – I. Таким образом

$$i\text{А} + \text{I} = \begin{cases} (i + \text{I})\ \text{А}, & \text{если } i < 3, \\ 0\ (\text{А} + \text{I}), & \text{если } i = 3, \end{cases}$$

и аналогично

$$i\text{А} - \text{I} = \begin{cases} (i - \text{I})\ \text{А}, & \text{если } i > 0, \\ 3(\text{А} - \text{I}), & \text{если } i = 0. \end{cases}$$

По индукции определяются сумма iА + К и разность iА – К для любого натурального К.

Каждая формула в ОРП I занимает группу подряд расположенных позиций по одной тетраде в каждой позиции. Код 0000 начального граничного символа Δ всегда записывается в нулевую позицию некоторой ячейки А, а следующие тетрады Q_1, Q_2, ... кода

$$0000\ Q_1\ Q_2\ \ldots\ldots\ Q_m\ 0000$$

данной формулы заносятся последовательно в позиции IА, 2А и т.д. Если при этом код 0000 конечного граничного символа Δ окажется в нулевой позиции некоторой ячейки В, то начиная с ячейки В в ОРП I размещается следующая формула (и символ Δ в позиции 0В становится одновременно конечным граничным символом для первой формулы и начальным граничным символом для второй). Если же код 0000 конечного граничного символа первой формулы окажется в первой, второй или третьей позиции ячейки В, то оставшиеся позиции ячейки В, если они еще есть, заполняются любыми, чаще всего нулевыми тетрадами, а запись второй формулы начинается с ячейки В + I.

Например, если в ячейках ОЗУ начиная с ячейки 05000 последовательно записать коды формул

Δ (A+B)/(A+C)ΔA*0.5+0ΔCOS*LN*AΔΔ COS*SIN
AΔ%X+%Y2↑(%X0%Y2)ΔA+ARCCTG*AΔ

(пользуясь кодировкой элементарных символов, указанной в таблице 5,6 и 7), то содержимое этих ячеек примет следующий вид (справа даны пояснения):

05000	0000	0002	II00	0024	Δ (A +	первая формула
0500I	II02	0003	0044	0002	B) / (	
05002	II00	0024	II24	0003	A + C)	
05003	0000	II00	0034	6004	Δ A * КОД	вторая формула
05004	4000	4000	4000	0024	числа 0.5 +	
05005	4000	4000	4000	4000	Код числа 0	
05006	0000	II2I	III6	II50	Δ C O S	третья формула
05007	0774	II44	II45	0774	* L N *	
050I0	II00	0000	0000	0000	A Δ Δ Δ	
050II	0000	0000	000I	0000	код пустой формулы	
050I2	0000	II2I	III6	II50	Δ C O S	пятая формула
050I3	0774	II50	II42	II45	* S I N	
050I4	0774	II00	0000	0000	* A Δ Δ	
050I5	0000	3600	0024	36I2	Δ%X0+%Y2	шестая формула
050I6	0050	0002	3600	0034	↑ (%X0 *	
050I7	36I2	0003	0000	0000	%Y2) Δ Δ	
05020	0000	II00	0024	II00	Δ A + A	седьмая формула
0502I	II47	II2I	II2I	II2I	R C C T	
05022	II4I	0774	II00	0000	G * A Δ	

(Ячейка с кодом пустой формулы должна содержать среди последних двух "незначащих" тетрад хотя бы одну ненулевую тетраду. В противном случае эта формула исчезнет при первом же включении Диспетчера).

В принципе можно было бы более плотно упаковывать формулы в ОРП I, всегда совмещая граничные символы соседних формул в полном соответствии с ролью символа Δ . Такая система записи формул для программирования аналитических выкладок была бы даже удобнее, но она привела бы к резкому увеличению времени работы диспетчера.

(d) СПЕЦИАЛЬНОЕ РАБОЧЕЕ ПОЛЕ. СРП занимает часть ОЗУ начиная с ячейки $A^{o}_{СРП}$ и кончая ячейкой $A^{\omega}_{СРП} = A^{o}_{МО} + 377$. Оно (совместно с БРП) предназначено для хранения программ всех операторов Авто-Аналитика, используемых в решении задачи пользователя, а также

программ операторов, составленных пользователем и временно – на период решения его задачи – включенных в библиотеку операторов Авто-Аналитика. Инструкция по составлению операторов будет изложена в § 6. Сейчас мы отметим лишь, что программа каждого оператора на СРП занимает целое число подряд расположенных ячеек; в первой из этих ячеек нулевая позиция должна содержать тетраду 0000.

Формулы и программы операторов, записанные в ОРП и СРП, не остаются неподвижными. Диспетчер в поисках свободного места передвигает их по оперативной памяти и может даже отправить их на барабан. Поэтому возникает необходимость в "информационной службе", которая указывала бы, где в данный момент находится данная формула или программа оператора. Для формул и операторов, находящихся в ОЗУ, такую службу несут три органа Авто-Аналитика: оперативный адресный стол, массив настроек и таблица переходов.

(е) ОПЕРАТИВНЫЙ АДРЕСНЫЙ СТОЛ. ОАС занимает часть ОЗУ начиная с ячейки A^{o}_{OAC} и кончая ячейкой $A^{\sim}_{OAC} = A^{o}_{TП} - I$. ОАС предназначен для хранения информации о месте записи в ОЗУ формул и операторов. Для прописки в ОАС все формулы и операторы снабжаются четырехзначными восьмеричными номерами 0001÷7777. Одной формуле или одному оператору нельзя сопоставлять несколько номеров. Один и тот же номер нельзя сопоставлять двум разным формулам или операторам, а также формуле и оператору. В частности, если в ОЗУ имеется несколько копий одной и той же формулы, то каждая копия должна быть прописана под своим собственным номером.

Последнее правило допускает одно исключение. А именно, формулы, расположенные в ОРП непосредственно друг за другом и образующие единый в смысловом отношении массив (таковы, например, левая и правая части правила; группа правил для решения одной специализированной задачи; группа формул, изображающих отдельные уравнения в какой-либо системе уравнений; формулы, являющиеся членами определителя и т.п.), могут быть прописаны в ОАС под одним номером.

Вообще почти любую последовательность тетрад можно снабдить отдельным номером и прописать в ОАС; требуется лишь, чтобы каждая такая последовательность начиналась тетрадой 0000, занимала в ОРП некоторое количество позиций подряд, начиная с нулевой позиции некоторой ячейки и не содержала в себе полностью обнуленной ячейки. Различные прописываемые в ОАС последовательности тетрад не должны пересекаться.

Номера операторов, постоянно входящих в библиотеку Авто-Аналитика, фиксированы раз и навсегда. Номера формул и операторов, составленных пользователем, задаются самим пользователем. Некоторые дополнительные замечания по этому поводу см. в § 6.

Каждой формуле в ОРП (оператору в СРП) сопоставляется отдельная строка ОАС, занимающая одну ячейку и имеющая вид

$$\ell_N \quad N \quad \pi_N \quad \gamma_N \quad A_N \qquad (5)$$

В этой строке разряды 36÷25 содержат номер N данной формулы (оператора). В разрядах 48÷37 указывается четырехзначное восьмеричное число ℓ_N ячеек ОЗУ, занятых формулой (оператором). Предполагается, что максимальное число ячеек ОЗУ, которые могут быть отведены под запись одной формулы (Оператора), равняется 7777_8. Таким образом, в Авто-Аналитике допустимы формулы, содержащие до I6380 элементарных символов. Адрес A_N первой из ячеек ОЗУ, занятых формулой (оператором) с номером N, заносится в I5 младших разрядов строки (5). 24-й разряд строки (5) равен 0, если эта строка соответствует формуле в ОРП, и I, если она соответствует оператору в СРП.

Оставшиеся восемь разрядов 23÷I6 строки (5) отведены под счетчик обращений за данной формулой (оператором). В этом счетчике хранится число $000 \le \gamma_N \le 377$, операторов, заинтересованных в том, чтобы формула (оператор) N находилась в ОЗУ. Для формулы с номером N равенство $\gamma_N = 000$ означает, что она в данный момент не используется для преобразования иных формул и сама не обрабатывается. Для оператора с номером N равенство $\gamma_N = 000$ означает, что он уже завершил свою работу или еще не приступил к ней. В соответствии с этим пользователь обязан установить значение $\gamma_N = 000$ при вводе исходных формул и собственных операторов с перфокарт, а также при получении новых формул программным путем.

Равенство $\gamma_N = 000$ информирует Диспетчер о том, что формула или оператор с номером N могут быть перенесены на барабан. Значение $\gamma_N \neq 000$ защищает формулу или оператор с номером N от перевода на барабан.

Увеличение числа γ_N на единицу происходит автоматически при каждом обращении к адресному столу (при помощи специальной стандартной программы) за информацией о месте записи формулы или оператора с номером N. Для уменьшения числа γ_N на единицу предусмотрена специальная стандартная программа, к которой следует обращаться для восстановления исходного состояния γ_N всякий раз,

когда информация о месте записи формулы (оператора) N становится ненужной. Если в каком-либо операторе обращение за информацией о формуле (операторе) N происходило несколько раз, то в конце своей работы этот оператор должен столько же раз отказаться от данной информации.

Для ускорения поиска нужной строки в ОАС все его строки располагаются в порядке возрастания номера N и заполняют все ячейки, отведенные под ОАС. Последнее, в частности, означает, что границы $A^{o}_{ОАС}$ и $A^{\omega}_{ОАС}$ в процессе работы Авто-Аналитика непрерывно меняются.

При перемещении формул и операторов по ОЗУ во время работы Диспетчера адресные части A_N соответствующих строк ОАС надлежащим образом корректируются.

(ℓ) КОНСТАНТЫ НАСТРОЕК. Шестизначный восьмеричный адрес i A произвольной позиции ОРП удобно представлять в виде 48- разрядной константы

$$(62 - 6i)\ 00 \quad 0000 \quad 000 \quad A, \qquad (6)$$

где младшие I5 двоичных разрядов заняты адресом A, а номер i-й позиции этой ячейки зашифрован в двух старших восьмеричных цифрах. Разряды 48÷42 машинного порядка константы (6) очевидно содержат число $I44_8 - I4_8 \cdot i$. Сдвиг содержимого ячейки A по константе с таким порядком переводит тетраду из i-й позиции в третью позицию. Таким образом, если адрес ячейки, содержащей константу (6), указать, например, в I6-том индексном регистре, а в ячейке p ОЗУ заготовить константу

0000 0000 0000 7777,

то в результате последовательного выполнения четырех команд

```
I6   23   00000
00   0I0  0000
I6   026  0000
00   0II  p
```

тетрада, записанная в позиции i A, окажется в I2-ти младших разрядах сумматора. Следовательно, константы вида (6) весьма полезны для программы потетрадного чтения кода любой формулы.

Константу вида (6) мы будем именовать константой настройки или просто настройкой на позицию i A. Разряды 4I÷I6 константы настройки всегда должны быть нулевыми. Для хранения настроек в теле БСП выделен специальный массив ячеек 00I70÷00237. Иную

информацию в данные ячейки записывать категорически запрещается. В частности их ни в коем случае нельзя обнулять (Несоблюдение этих правил работы с настройками может расстроить всю информационную службу Авто-Аналитика и "перемешать" в памяти все формулы и операторы)

В период работы Диспетчера адресные части настроек "настроенных" на тетрады сдвигаемых формул и операторов, корректируются должным образом. Это позволяет организовывать все программы обработки формул так, что они "не замечают" того, что время от времени обрабатываемые формулы (и даже сами программы, производящие эту обработку) путешествуют по оперативной памяти.

Настройки 00170÷00177 используются в качестве рабочих ячеек некоторыми стандартными программами, а также Ридером и самим Диспетчером. Настройка 00236 указывает на последнюю занятую позицию ОРП I. Настройка 00237 указывает на последнюю занятую позицию ОРП II. Между этими двумя позициями расположен резерв ОРП. Значит данные две настройки рекомендуется применять исключительно для записи новых формул. Включение Диспетчера происходит "как раз тогда, когда настройки 00236 и 00237 "сталкиваются" друг с другом.

Настройки 0200÷0235 предоставлены пользователю для свободного употребления.

(*g*) ТАБЛИЦА ПЕРЕХОДОВ. ТП занимает часть ОЗУ, начиная с ячейки $A^{o}_{ТП}$ и кончая ячейкой $A^{\omega}_{ТП} = A^{o}_{СРП} - I$, и является рабочим массивом Ридера переменной длины. Подобно ОАС и массиву настроек, ТП состоит из строк; их назначение и структура будет описана в § 6. Сейчас мы отметим лишь, что каждая строка занимает одну ячейку в ТП и в ней I5 младших разрядов содержат адрес некоторой ячейки некоторого оператора. При сдвиге этого оператора в пределах СРП данный адрес должен корректироваться Диспетчером.

(*h*) МАССИВ ОБМЕНА. Массив обмена занимает последние три абзаца последнего из листов ОЗУ, предоставленных в распоряжение Авто-Аналитика. Адрес нулевой ячейки этого листа обозначается через $A_{МО}$. Следовательно, МО занимает часть ОЗУ начиная с ячейки $A_{МО}+400$ и кончая ячейкой $A_{МО}+I777$ (нулевой абзац данного листа присоединен к СРП). Массив обмена предназначен для организации группового обмена информацией между ОЗУ и памятью на магнитных барабанах. При этом первый и второй абзацы МО служат для обмена между ОРП и БРП, а третий абзац – для обмена между ОАС и БАС.

(i) БАРАБАННЫЙ АДРЕСНЫЙ СТОЛ. БАС занимает целое число секторов в начале нулевого МБ первого направления, а именно – ячейки МБ с 20-разрядными адресами от I 000 00I по I 000000 + N_{max} – I, где N_{max} – некоторое четырехзначное восьмеричное число, делящееся без остатка на 400, и предназначен для хранения информации о формулах и операторах, записанных в БРП. Каждой формуле или оператору с номером $N \neq 0$ сопоставляется фиксированная строка БАС, занимающая на МБ ячейку с адресом I 000000 + N, независимо от того, есть формула или оператор с номером N в БРП или нет.

Таким образом, в БАС могут быть прописаны лишь те формулы и операторы, номера которых строго меньше числа N_{max} Все прочие формулы и операторы никогда на барабан не переносятся.

Если формула или оператор с номером N действительно записаны в данный момент в БРП, то соответствующая строка БАС имеет вид

$$\ell_N \quad N \quad \pi_N \quad 0 \quad A^N_{МБ} \qquad (7)$$

где N – номер формулы или оператора (указывается в разрядах 36÷25 строки (7), ℓ_N – количество ячеек МБ, занятых данной формулой или оператором (указывается в разрядах 48÷37), $A^N_{МБ}$ – адрес первой из этих ячеек МБ (указывается в 20-ти младших разрядах строки (7)), π_N – признак формулы или оператора, равный 0, если строка (7) соответствует формуле, и I, если она соответствует оператору (признак π_N занимает в строке (7) 24-й разряд). Разряды 23÷2I строки (7) всегда нулевые.

Если же в БРП нет ни формулы, ни оператора с номером N и $N < N_{max}$, то N-я строка БАС в своих двенадцати старших разрядах содержит нули.

(j) БАРАБАННОЕ РАБОЧЕЕ ПОЛЕ. БРП занимает всю остальную память на МБ, предоставленную в распоряжение Авто-Аналитика, т.е. ячейки МБ с адресами от I 000000 + N_{max} по $A^{\omega}_{МБ}$. Адрес $A^{\omega}_{МБ}$ + I должен быть кратным числу 2000. БРП является продолжением ОРП и СРП и предназначено для хранения формул и операторов, которые в данный момент не используются, но могут потребоваться в будущем. Каждая формула и каждый оператор в БРП занимают целое число подряд расположенных ячеек. Причем в нулевой позиции первой из этих ячеек пишется тетрада, равная номеру данной формулы или

оператора. (При передаче формулы или оператора из ОРП или СРП в БРП начальный граничный символ Δ заменяется номером; при обратной передаче формулы или оператора из БРП в ОРП или СРП происходит восстановление начального символа Δ). Каждая формула и каждый оператор в БРП прописываются в БАС. Диспетчер может передвигать формулы и операторы в БРП. В этом случае он корректирует соответствующие строки БАС.

Отметим еще, что при обмене формулами между ОРП и БРП внутренняя структура формулы не анализируется (но полностью обнуленные ячейки улавливаются и удаляются). Поэтому на БРП могут поступать также массивы формул и иные последовательности тетрад, прописанные под видом формулы.

(*k*) ОСНОВНЫЕ ПАРАМЕТРЫ. Информация о распределении памяти в любой момент времени задается так называемыми основными параметрами Авто-Аналитика, которые занимают ячейки 00263÷ 00277 в теле ЕСП и имеют соответственно следующий вид

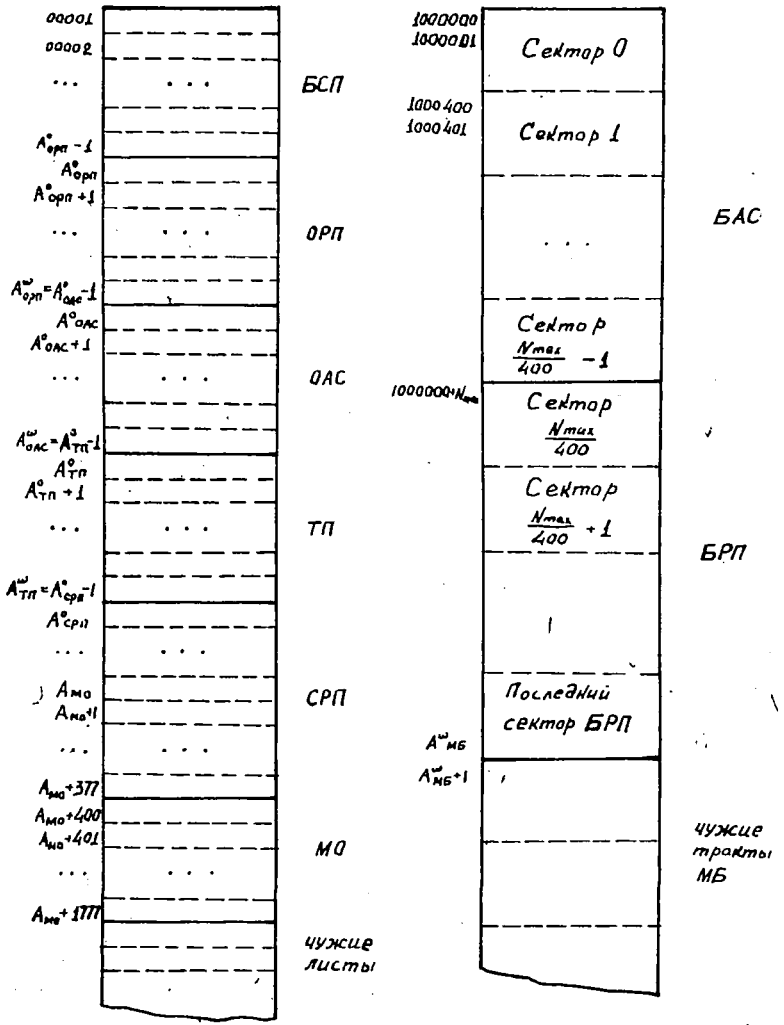
00263	Шкала коммутативных связей			
00264	Шкала ассоциативных связей			
00265	0000	0000	0000	$A^{\circ}_{ОРП}$
00266	0000	0000	000	$A^{\circ}_{ОАС}$
00267	0000	0000	000	$A^{\circ}_{ТП}$
00270	0000	0000	000	$A^{\circ}_{СРП}$
00271	0000	0000	000	$A_{МО}$
00272	0000	0000	0	$A_{БАС}$
00273	Параметр τ			
00274	0000	N_{max}	0000	0000
00275	0000	0000	0	$A_{БРП}$
00276	0000	0000	0	$\ell_{РМБ}$
00277	0000	0000	0	$(A^{\circ}_{МБ} + 1)$

Роль шкал 00263 и 00264 указана в предыдущем параграфе. Параметры

$$A^{\circ}_{ОРП},\ A^{\circ}_{ОАС},\ A^{\circ}_{ТП},\ A^{\circ}_{СРП},\ A_{МО},\ N_{max},\ A_{МБ}$$

задают описанную выше структуру ОЗУ и МБ (Эта структура иллюстрируется также на рис. 5).

Параметр $\ell_{РМБ}$ занимает 20 младших разрядов ячейки 00276 и равен количеству свободных ячеек БРП (= резерв МБ). Параметр $A_{БРП}$ представляет собою 20-разрядный адрес произвольной



Структура ОЗУ Структура МБ

Рис.5. Структура ОЗУ и МБ

свободной ячейки БРП (начиная с этой ячейки Диспетчер записывает новые операторы и формулы, поступающие из ОЗУ).

Параметры $A_{БАС}$ и τ управляют работой с БАС. Эта работа построена так, что обычно копия некоторого сектора БАС представлена в третьем абзаце массива обмена. Если в рассматриваемый момент времени это действительно имеет место, то параметр $A_{БАС}$ совпадает с 20-разрядным адресом нулевой ячейки данного сектора БАС, а параметр τ равен нулю. В этом случае при обращении к тому же сектору БАС (для записи в него новой строки или для чтения старой) никакого обмена между ОЗУ и МБ не происходит; чтение строки сводится к выдаче содержимого соответствующей ячейки третьего абзаца МО, а запись новой строки – к записи ее в эту же ячейку МО. Поэтому иногда содержимое третьего абзаца МО не идентично содержимому соответствующего сектора БАС, причем исправленные строки БАС находятся именно в его "копии". В последнем случае параметр τ принимает ненулевое значение. Если теперь возникает потребность затереть третий абзац МО (например, при обращении к иному сектору БАС), то в случае $\tau \neq 0$ третий абзац МО предварительно переписывается в соответствующий сектор БАС и только потом затирается, а в случае $\tau = 0$ – затирается сразу.

Параметры $A^{o}_{ОАС}$, $A^{o}_{ТП}$, $A^{o}_{СРП}$, $A_{БАС}$, τ, $A_{БРП}$ и $\ell_{РМБ}$ изменяются Диспетчером и некоторыми другими стандартными программами. Остальные параметры постоянны в течение всей работы Авто-Аналитика. Исходное состояние всех основных параметров Авто-Аналитика устанавливается на основании указаний пользователя (см. по этому поводу n^{o} 2 § 3 гл. Ш).

§ 4. ДИСПЕТЧЕР

Задачей Диспетчера является обеспечение требуемого резерва ОРП. Величина этого резерва (т.е. количество ячеек в нем) задается в младших разрядах сумматора в виде целого восьмеричного числа S. В поисках свободного места Диспетчер просматривает ОРП I и ОАС. Все нулевые ячейки ОРП I и "пустые" строки ОАС присоединяются к резерву ОРП. Если объем полученного в результате резерва ОРП больше, чем S, то Диспетчер прекращает свою работу. В противном случае он организует перенесение на БРП всех

тех формул и операторов, для которых эта процедура разрешена. При этом Диспетчер считает, что на БРП можно переписывать те и только те формулы и операторы, которые размещены в ОРП I или СРП, прописаны в ОАС под отличным от нуля номером $N < N_{max}$ и в своем счетчике обращений содержат число γ_N = 000. Освобожденные ячейки ОЗУ присоединяются к резерву ОРП. Если и теперь требуемый резерв ОРП не обеспечен, то Диспетчер прерывает работу Авто-Аналитика из-за недостатка оперативной памяти.

Более подробно работа Диспетчера происходит следующим образом.

ЭТАП I. Обнуляется вспомогательный параметр τ и проверяется неравенство

$$A_{236} + S + 3 \leq A_{237}. \qquad (8)$$

(Здесь и всюду далее, где не оговорено иное, под A_H для 00170 $\leq H \leq$ 00237 понимается адресная часть, т.е. содержимое I5-ти младших разрядов, настройки, хранящейся в ячейке H). Если это неравенство выполнено, то резерв ОРП считается достаточным и осуществляется выход из Диспетчера. В противном случае начинается второй этап.

ЭТАП 2. На этом этапе производится "чистка" ОАС. Из ОАС удаляются все строки с нулевой адресной частью; такие строки в ОАС могут возникнуть в результате работы некоторых стандартных программ и операторов. Например, стандартная программа "Аннулирование формулы в ОЗУ" по указанному ей номеру N отыскивает в ОАС строку

$$\ell_N \; N \; \pi_N \; \gamma_N \; A_N \quad ,$$

заменяет ее строкой

$$0000 \quad N \quad 0000 \quad 0000$$

и обнуляет ячейки с адресами A_N, $A_N + I$, ..., $A_N + \ell_N - I$, занятые формулой или оператором с номером N. Оставшиеся строки ОАС"уплотняются", т.е. собираются и в прежнем порядке записываются в конце ОАС. Верхняя граница $A^{\omega}_{OAC} = A^{o}_{TП} - I$ оперативного адресного стола остается без изменения, а его нижняя граница A^{o}_{OAC} увеличивается на число δ удаленных строк ОАС. В этот момент между ОРП П и ОАС оказывается δ свободных ячеек ОЗУ. На этапе 2 Диспетчер предполагает, что в ОАС имеется хотя бы одна строка; поскольку работа этапа 2 возможна после удаления всех допустимых формул и операторов из ОЗУ в БРП на этапе 6, то приходится также предполагать, что хотя бы одна строка ОАС

соответствует нужной формуле или оператору с номером $N \geq N_{max}$ или с ненулевым числом δ_N .

ЭТАП 3. На данном этапе происходит присоединение к резерву ОРП ячеек, освобожденных в результате работы предыдущего блока. Для этого массив ячеек, образующих ОРП П, сдвигается как единое целое в сторону увеличения адресов до тех пор, пока между ОРП П и ОАС не исчезнут все свободные ячейки. Сдвиг ОРП П осуществляется при помощи стандартной программы "Сдвиг вперед" (Полные описания этой и большинства иных СП, упоминаемых ниже, содержатся в следующем параграфе). При этом надлежащим образом корректируются все те настройки и строки ОАС и ТП, в которых адресная часть А удовлетворяет неравенству

$$A_{237} \leq A \leq A^{\omega}_{ОРП} + I$$

В частности, корректируется настройка 00237 и тем самым увеличивается адрес A_{237}. Адрес A_{236} остается неизменным, поскольку всегда предполагается, что $A_{236} < A_{237}$ (в противном случае Диспетчер не сумеет "развести" настройки 00236 и 00237 на величину требуемого резерва ОРП и произведет прерывание работы Авто-Аналитика). Внутренняя структура ОРП П не анализируется. Значит, если в ОРП П есть обнуленные ячейки, то они сохранят свое положение относительно соседних кодов и к резерву ОРП не присоединяются.

ЭТАП 4. На этом этапе происходит "чистка" ОРП I. Из ОРП I удаляются все обнуленные ячейки, а коды из остальных ячеек ОРП I размещаются в том же порядке в начале ОРП. Настройки и строки ОАС И ТП, нацеленные на сдвигаемые участки ОРП I, корректируются соответствующим образом.В частности, уменьшается адрес A_{236} и тем самым увеличивается резерв ОРП и сокращается объем ОРП I. К концу работы данного блока ОРП I будет содержать в себе ровно столько ячеек, сколько было ненулевых кодов в старом ОРП I (плюс одна нулевая ячейка, если в последней ячейке старого ОРП I был записан нуль).

Более подробно работа Диспетчера на этом этапе заключается в следующем. В счетчик магазина (ИР с номером I7) заносится адрес $A^{o}_{ОРП}$. Как только после серии пустых ячеек встречается первая ячейка с ненулевым кодом, ее адрес временно обозначается через α , а разность $m - \alpha$, где m – содержимое счетчика магазина, – через δ . Ненулевые коды из ячеек α, $\alpha + I$, ...

последовательно записываются в магазин до тех пор, пока не встретится обнуленная ячейка. Адрес предыдущей ячейки (последней в группе необнуленных ячеек α, $\alpha + 1, \ldots$) обозначается через ω (нулевой код из ячейки $\omega + 1$ в магазин не поступает) и включается СП "Общая корррекция". Эта СП кооректирует все те настройки и строки ОАС и ТП, у которых адресная часть А удовлетворяет неравенству $\alpha \leq A \leq \omega$. Затем производится поиск и аналогичная обработка следующей группы ненулевых кодов в ОРП I и т.д. Описанная процедура продолжается до тех пор, пока "читающая головка" данного блока не достигнет последней в ОРП I ячейки A_{236}. При этом возможны два случая в зависимости от того, равно или не равно нулю содержимое ячейки A_{236-1}, предшествующей ячейке A_{236}. В первом случае очередное значение параметров α и ω полагается равным A_{236}, параметр δ полагается равным $m - \alpha$. Код из ячейки A_{236} переносится в магазин (независимо от того, равен он нулю или нет), и включается СП "Общая коррекция". Во втором случае параметры α и δ уже получили в соответствии с описанной процедурой некоторые значения и ненулевые коды из ячеек α, $\alpha + 1$, ..., A_{236-1} последовательно занесены в магазин; в этом случае ω полагается равным A_{236}, код из ячейки A_{236} переписывается в магазин (также независимо от того, равен он нулю или нет) и производит обращение к СП "Общая коррекция".

В связи с описанным способом преобразования ОРП I пользователю нужно иметь в виду следующие замечания.

I) При однократном включении Диспетчера объем получаемого резерва ОРП тем больше, чем больше обнуленных ячеек в ОРП I. Обнуленные ячейки в ОРП I образуются в результате аннулирования формул, появившихся на некоторый период в ОРП I и затем оказавшихся ненужными для дальнейшего счета (это аннулирование производится пользователем либо "вручную", либо при помощи стандартных программ "Аннулирование формулы в ОЗУ", "Аннулирование формулы всюду", а также "Прописка формулы"). Но больший в среднем объем освобождаемого резерва ОРП влечет уменьшение частоты включения Диспетчера, а значит сокращение времени работы Авто-Аналитика по программе пользователя. Кроме того, при больших количествах обнуленных участков ОРП реже возникает необходимость в работе дальнейших блоков Диспетчера, осуществляющих перевод формулы и операторов на барабан. А это также существенно снижает потери времени. Таким образом, пользователю рекомендуется производить

немедленное аннулирование формул и операторов в тот момент времени, когда станет известной их бесполезность для дальнейшего.

2) Ненулевые участки ОРП I обрабатываются Диспетчером одинаково независимо от того, прописаны они в ОАС или нет. Значит, отдельные массивы формул и иных последовательностей элементарных символов, находящихся в ОРП I, можно в ОАС не прописывать. Чтобы не потерять их при срабатывании Диспетчера, достаточно на одну из тетрад каждого такого массива настроить какую-нибудь настройку. Таким образом можно экономить номера формул (что особенно важно при составлении операторов). Следует однако иметь ввиду, что этот прием программирования обладает одним опасным недостатком: каждый массив непрописанной в ОАС информации аннулировать приходится вручную. Ни в коем случае нельзя допускать потерю прописанных массивов в ОРП I. Подобные беспризорные массивы будут находиться в ОРП I до конца решения задачи, увеличивая тем самым частоту включения Диспетчера и обмена с МБ. Небрежность подобного сорта допущенная в цикле, неизбежно приведет к переполнению ОРП и вызовет прерывание работы Авто-Аналитика.

3) Каждый массив информации в ОРП I должен начинаться с нулевой тетрады. Несоблюдение этого правила может привести к путанице. Рассмотрим, например, формулы

$$Ш = \Delta A + B + C + Д\,\Delta\,,\quad Щ = \Delta A + B + C\,\Delta$$

и последовательность тетрад

0034 0002 II00 0024 II02 0003 0000,

образующую самостоятельный информационный массив Ц. Если формулы Ш, Щ и массив Ц записаны в ОРП I подряд начиная с ячейки р + о, то соответствующий участок ОРП I имеет вид

р + о	0000	II00	0024	II02	Δ A + B	Формула Ш
р + I	0024	II2I	0024	II04	+ C + Д	
р + 2	0000	II00	0024	II02	Δ A + B	Формула Щ
р + 3	0024	II2I	0000	0000	+ C Δ Δ	
р + 4	0034	0002	II00	0024	x (A +	Массив Ц
р + 5	II02	0003	0000	0000	B) Δ Δ	

Допустим, что формула Щ оказалась ненужной. После ее аннулирования данный участок ОРП I принимает вид

p + o	0000	II00	0024	II02	ΔA + B ⎫ Формула Ш
p + I	0024	II2I	0024	II04	+ C + Д ⎭
p + 2	0000	0000	0000	0000	
p + 3	0000	0000	0000	0000	обнуленные ячейки
p + 4	0034	0002	II00	0024	x (A + ⎫
p + 5	II02	0003	0000	0000	B) Δ Δ ⎭ Массив Ц.

Если теперь произойдет включение Диспетчера, то формула Ш и массив Ц окажутся записанными рядом:

q + 0	0000	II00	0024	II02	⎫
q + I	0024	II2I	0024	II04	⎭ Бывшая формула Ш
q + 2	0034	0002	II00	0024	⎫
q + 3	II02	0003	0000	0000	⎭ Бывший массив Ц

и в качестве формулы Ш при ее использовании будет выдана формула

A + B + C + Д x (A + B)

4) Константы настроек, указывающие на позиции обнуленных ячеек, естественно не корректируются. Поэтому иногда не будут корректироваться настройки, следящие за конечным граничным символом Δ какой-либо формулы. Например, в ситуации, рассмотренной в предыдущем замечании, настройка на нулевую позицию ячейки p + o, содержащей конечный граничный символ Δ формулы Ш, во время работы Диспетчера не изменится, хотя сама формула Ш возможно уйдет в другое место ОРП I.

ЭТАП 5. На данном этапе Диспетчер предпринимает вторую попытку выхода. Как и в первый раз (на этапе I), проверяется условие (8). Если оно выполнено, то работа Диспетчера заканчивается. В противном случае Диспетчер приступает к переписыванию формул и операторов из ОЗУ в БРП.

ЭТАП 6. На данном этапе Диспетчер отыскивает все формулы в ОРП I и операторы в СРП, которые можно перенести на барабан, и переписывает их в БРП. Попутно Диспетчер "уплотняет" уже имеющуюся информацию в БРП для того, чтобы отыскать там место для формул и операторов, поступающих из ОЗУ. Освобождаемые в результате этого ячейки ОЗУ обнуляются.

Как уже упоминалось выше, для обмена между ОЗУ и МБ в Авто-Аналитике отведен массив обмена, включающий последние три абзаца листа ОЗУ с адресом $A_{МО}$. Первый из этих абзацев служит накопителем кодов, пересылаемых из ОЗУ в БРП, второй является приемником информации, поступающей из БРП в ОЗУ, а третий абзац

МО обслуживает работу Диспетчера и некоторых других стандартных программ с БАС.

Для упрощения алгоритма данной части Диспетчера барабанное рабочее поле программным образом организовано в виде "кольца". А именно, для каждого сектора БРП определен "следующий" сектор. Если исходный сектор включает в себя ячейки $A \div A + 377$ из БРП и $A + 377 < A^{\omega}_{МБ}$, то "следующий" сектор состоит из ячеек $A + 400 \div A + 777$; для последнего сектора $A^{\omega}_{МБ} - 377 \div A^{\omega}_{МБ}$ из БРП в качестве следующего сектора зафиксирован начальный сектор БРП, начинающийся с ячейки $1000000 + N_{max}$

Непосредственную работу с барабаном Диспетчер осуществляет при помощи четырех стандартных программ: "Выдачи строки БАС", "Запись строки БАС", "Чтение кода из БРП" и "Запись кода в БРП".

СП "ВЫДАЧА СТРОКИ БАС". Обращение к этой СП осуществляется командой I5 3I 00511. В качестве исходной информации данная СП использует номер N нужной строки БАС, заданный в первой позиции ячейки 00243 (остальные позиции должны содержать нули). В качестве результата своей работы СП "Выдача строки БАС" записывает в сумматор содержимое ячейки МБ с адресом $1000000 + N$ Кроме того, в момент окончания работы данной СП в третьем абзаце МО будет содержаться копия этого сектора МБ, в который входит ячейка $1000000 + N$, адрес $A_{БАС}$ нулевой ячейки этого сектора окажется в ячейке 00272, а содержимое ячейки 00273 будет равно нулю. Данная СП предполагает, что третий абзац МО и ячейки 00272, 00273 кроме нее самой и кроме СП "Запись строки БАС" никакая другая программа (или оператор) в своей работе не затрагивают. Алгоритм работы Сп "Выдача строки БАС" приведен на рис. 6 (обозначения: N^8 – целое двоичное число, представленное восьмью младшими разрядами номера N; B и q – вспомогательные параметры; C – содержимое сумматора; $(A_{МО} + 1400 + q)$ – содержимое q-й ячейки третьего абзаца МО; $A_{БАС}$, $A_{МО}$ и ξ – основные параметры работы Авто–Аналитика).

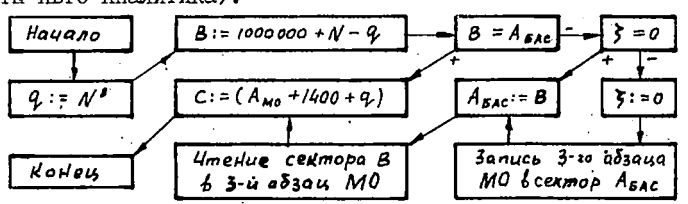


Рис. 6. Выдача строки БАС

СП "ЗАПИСЬ СТРОКИ БАС". Обращение I5 3I 00526. Исходная информация: новая строка БАС в сумматоре и ее номер N в первой позиции ячейки 00243. Результат: новый вид N-й строки БАС записан в копии соответствующего сектора БАС в третьем абзаце МО, параметр τ отличен от нуля.

Алгоритм работы: Начало р : = С; выдача строки БАС; $(A_{МО} + I400 + q)$: = р; τ : = р конец.

СП "ЧТЕНИЕ КОДА ИЗ БРП". Обращение I0 3I 0I023. Исходная информация: в 20-ти младших разрядах ячейки 00245 – адрес β нулевой ячейки барабанного сектора $\angle$, содержащего ячейку МБ с адресом $A'_{БРП}$; в I2-том ИР – величина θ – 400, где θ – число, представленное восьмью младшими двоичными разрядами адреса $A'_{БРП}$ (т.е. относительный адрес ячейки $A'_{БРП}$ в секторе $\angle$); во втором абзаце МО – копия сектора $\angle$. Если $A'_{БРП}$ кратно 400, то вместо указанной информации в ячейку 00245 можно записать число $\beta = A'_{БРП} - 400$ (обозначая тем самым через $\angle$ сектор, предшествующий тому, в котором содержится ячейка $A'_{БРП}$) и I2-й ИР обнулить; содержимое второго абзаца МО в этом случае несущественно (ясно, что в обоих случаях справедливо равенство

$$A'_{БРП} = \beta + И_{I2} + 400,$$

где $И_{I2}$ – содержимое I2-го ИР).

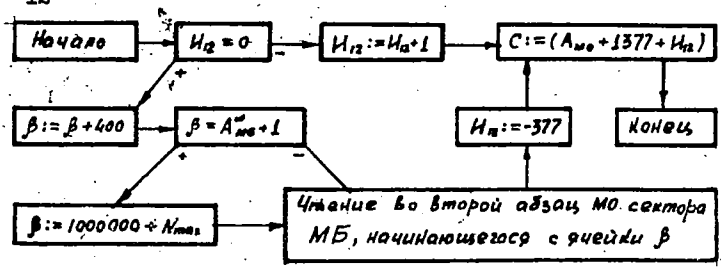


Рис.7. Чтение кода из БРП

Алгоритм работы данной СП изображен на рис. 7. Легко видеть, что этот алгоритм в качестве основного результата выдает в сумматор содержимое ячейки МБ с адресом $A'_{БРП}$. Кроме того, данная СП подготавливает себя к чтению кода из следующей ячейки $A'_{БРП} + I$. Если, не меняя содержимого ячейки 00243, I2-го ИР и второго абзаца МО, обратиться к этой СП вторично, то в сумматор поступит код из ячейки $A'_{БРП} + I$; при третьем обращении в сумматор

поступит содержимое ячейки $A'_{БРП}+2$ и т.д. Если в исходном состоянии $A'_{БРП}$ равно $A^{\omega}_{МБ}$, то при первом обращении к СП в сумматор будет выдан код из ячейки $A^{\omega}_{МБ}$, при втором – из ячейки 1000000 + N_{max} и т.д. Таким образом, данная СП обеспечивает последовательное считывание кодов из ячеек БРП в полном соответствии с нашим восприятием БРП в виде "кольца".

СП "ЗАПИСЬ КОДА В БРП". Обращение IO 3I OIOI4. СП предназначена для последовательной записи кодов в БРП. Код, подлежащий записи, указывается в сумматоре. Адрес $A_{БРП}$ ячейки БРП, куда нужно этот код записать, задается при помощи кода $И_{II} = A^{8}_{БРП} - 377$ в II–ом ИР и кода $\delta = A_{БРП} - A^{8}_{БРП}$ в 20–ти младших разрядах ячейки 00246. СП предполагает, что содержимое ячеек δ, $\delta+1, \ldots, \delta + A^{8}_{БРП} - 1 = A_{БРП} - 1$ сектора МБ, начинающегося с ячейки δ, уже записано в первых ячейках первого абзаца МО.

Алгоритм работы данной СП указан на рис. 8. Анализ этого алгоритма показывает, что СП "Запись кода в БРП" реализует для записи кодов в последовательные ячейки нашего"кольцевого" БРП "режим магазина".

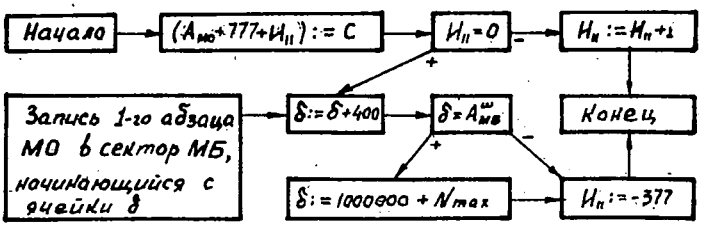


Рис. 8. Запись кода в БРП.

Теперь мы в состоянии описать более подробно алгоритм работы Диспетчера на этапе 6. Этот алгоритм заключается в следующем:

(α) Подготовка к обмену между ОЗУ и БРП. Вспомогательному параметру α присваивается значение, равное адресу $A^{o}_{ОАС}$. Во втором абзаце МО образуется копия сектора БРП, содержащего ячейку $A_{БРП}$. Часть этого же сектора от его начала и по ячейку $A_{БРП}-1$ переписывается в начало первого абзаца МО; оставшаяся часть первого абзаца МО остается пока свободной. Параметрам ρ, δ, $И_{II}$ и $И_{I2}$ присваиваются значения

$$\beta = \delta = A_{БРП} - A^{8}_{БРП},$$

$$И_{I1} = A^{8}_{БРП} - 377, \quad И_{I2} = A^{8}_{БРП} - 400$$

(*б*) Анализ строки ОАС. Исследуется строка

$$\ell_a \; N_a \; \pi_a \; \gamma_a \; A_a \qquad (9)$$

занимающая в ОАС ячейку a. Если в этой строке

$$0000 < \ell_a \leq \ell_{РМБ}, \quad 0000 < N_a < N_{max}, \quad \gamma_a = 000$$

и $A_a \leq A_{236}$ или $A_a \geq A^{o}_{СРП}$, то осуществляется переход на блок (*д*) (формула или оператор с номером N_a допускает перенос на БРП). В случае $N_a \geq N_{max}$ происходит переход на блок (*ѓ*) (строка (9) и строки ОАС, занимающие в нем ячейки $a+1$, $a+2$, ..., $A^{\omega}_{ОАС}$, имеют номера не меньше, чем N_{max}, и значит соответствующую информацию в БРП переписывать нельзя). В случаях $\ell_a > \ell_{РМБ}$ (для формулы или оператора с номером N_a в БРП не хватает свободного места), $\gamma_a > 000$ (формула или оператор с номером N_a используется в данный момент некоторым оператором Авто-Аналитика) или $A_{236} < A_a < A^{o}_{СРП}$ (прописанная под номером N_a информация находится в ОРП П), а также в случаях $\ell_a = 0000$ или $N_a = 0000$ (строка (9) не соответствует никакой формуле и никакому оператору в ОЗУ; что она означает – Диспетчеру неизвестно, но на всякий случай он ее не трогает) происходит переход на блок (*с*)

(*с*) Значение параметра a увеличивается на единицу. Если все еще $a \neq A^{o}_{ТП}$, то переход на блок (*б*). В противном случае переход на блок (*ѓ*).

(*д*) Накопление свободных ячеек БРП для формулы или оператора с номером N_a. При помощи СП "Чтение кода в БРП" просматриваются ячейки БРП начиная с ячейки $A_{БРП}$. Свободные (т.е. обнуленные) ячейки БРП подсчитываются. Встречающиеся формулы и операторы переписываются в БРП при помощи СП "Запись кода в БРП". При этом предполагается, что всякий массив информации, находящийся в БРП и прописанный под некоторым номером N в БАС, начинается с ненулевого кода, в нулевой позиции которого указан номер N, и занимает подряд расположенные ячейки (часть из них может быть обнулена), количество ℓ_N которых указано в строке БАС с номером N. Строки БАС, отвечающие передвигаемым по БРП формулам и операторам, корректируются надле-

жащим образом. Окончание работы данного блока и переход на следующий блок (е) совершается в тот момент, когда СП "Чтение кода в БРП" считает очередной нулевой код из свободной ячейки и количество пройденных данной СП свободных ячеек БРП сравняется с длиной l_a формулы или оператора с номером N_a. В этот момент очевидно

$$A_{БРП} = A'_{БРП} - l_a$$

(е) Перенос формулы или оператора с номером N_a из ОЗУ в БРП: В ячейку 1000000 + N_a, принадлежащую БАС, заносится строка

$$l_a N_a \tau_a \; 0 \; A_{БРП}$$

образованная на основе строки (9) и счетчика $A_{БРП}$ " барабанного магазина", организованного программой "Запись кода в БРП". Затем в сумматор считывается содержимое

$$Q_0 Q_1 Q_2 Q_3$$

ячейки ОЗУ с адресом A_a, нулевая тетрада Q_0 этого кода заменяется тетрадой N_a и результат

$$N_a Q_1 Q_2 Q_3$$

заносится в ячейку $A_{БРП}$ при помощи СП "Запись кода в БРП". Далее с помощью этой же СП в БРП переписываются без изменения коды ячеек ОЗУ с адресами $A_a + I$, $A_a + 2$, ..., $A_a + l_a - I$. Во все ячейки A_a, $A_a + I$, ..., $A_a + l_a - I$ оперативной памяти засылаются нули. Основной параметр $l_{РМБ}$ в ячейке 00276 уменьшается на l_a. Строка (9) в ОАС заменяется строкой

0000 N_a 0000 0000

Параметр τ принимает значение I и осуществляется переход на блок (с)

(д) Этот блок завершает работу Диспетчера по перенесению на барабан формул и операторов из ОЗУ. Согласно счетчику "барабанного магазина" (СП "Запись кода в БРП) устанавливается новое значение основного параметра $A_{БРП}$. Если содержимое II-го ИР не равно нулю, т.е. если в первом и втором абзацах МО остались еще коды, не отправленные на барабан программой "Запись кода в БРП", то все такие коды из второго абзаца МО (они остались еще не "просмотренными" во время работы блока (d) переносятся в свободные ячейки первого абзаца МО и содержимое этого абзаца отправляется в соответствующий (а именно, содержащий ячейку $A_{БРП}$) сектор МБ.

ЭТАП 7. На данном этапе Диспетчер уплотняет СРП, все освобожденные ячейки присоединяет к резерву ОРП и, если в результате этого резерв ОРП увеличивается, придает параметру τ значение I.

Более подробно работа Диспетчера на этом этапе состоит в следующем. Организуется просмотр СРП начиная с его ячейки $A^{o}_{СРП}$. Свободные (т.е. обнуленные) ячейки пропускаются и подсчитываются. Массивы ячеек, занятые операторами, сдвигаются к началу СРП (величина сдвига равняется количеству встретившихся к этому моменту свободных ячеек СРП) и производится соответствующая коррекция настроек и строк ОАС и ТП. При этом предполагается, что каждый оператор, находящийся в СРП, прописан в ОАС под ненулевым номером (иначе произойдет прерывание), начинается с ненулевого кода и занимает массив подряд расположенных ячеек (быть может, частично обнуленных).

Когда просмотр СРП завершается, основные параметры $A^{o}_{ОАС}$, $A^{o}_{ТП}$ и $A^{o}_{СРП}$ увеличиваются на число δ освобожденных (и пока собранных в конце СРП) ячеек; массив ячеек, образованный объединением ОРП И, ОАС, ТП и занятой операторами части СРП как единое целое сдвигается вперед (т.е. в сторону увеличения адресов) на δ ячеек; после завершения сдвига производится коррекция всех настроек и строк ОАС и ТП, нацеленных на позиции сдвигаемого массива (в частности корректируется адрес A_{237}). В заключении данного этапа все коды в СРП еще раз переписываются в те же самые ячейки, но с признаками команд.

ЭТАП 8. Если $\tau = 0$, то на этапах 6 и 7 в ОЗУ не было освобождено ни одной ячейки. В этом случае Диспетчер не в состоянии увеличить резерв ОРП и работа всей системы прерывается из-за переполнения ОРП. В случае $\tau = I$ осуществляется переход на этап I.

На этапе I в этом случае снова произойдет проверка условия (8). Если оно окажется выполненным, то работа Диспетчера закончится. (Следует отметить, что в данном случае в ОРП I и ОАС могут остаться свободные участки, не присоединенные к резерву ОРП. А именно – все те участки ОРП I и строки ОАС, которые были заняты формулами, отправленными на барабан. Присоединение этих ячеек к резерву ОРП произойдет во время следующего включения Диспетчера). Если же условие (8), несмотря на увеличение

резерва ОРП благодаря этапу 7, выполнено не будет, то после неудавшейся попытки выхода на этапе I Диспетчер перейдет к чистке ОАС и ОРП (Этапы 2,3 и 4) и на этапе 5 предпримет еще одну попытку выхода. Если она также не удается, то последующая работа на этапах 6 и 7 не изменит нулевого значения параметра τ (этот параметр обнуляется на первом этапе) и произойдет прерывание на этапе 8.

Блок-схема всего алгоритма Диспетчера приведена на рис. 9. Через СДВ, СДН и ОК на этой блок-схеме обозначены обращения к стандартным программам "Сдвиг вперед", "Сдвиг назад" и "Общая коррекция". Если a есть адрес ячейки ОЗУ, то (a) есть содержимое ячейки a и A_a есть код, записанный в I5-ти младших разрядах ячейки a. Аналогично следует понимать обозначения $(i\,A)$ и $(A_{БРП})$, где $i\,A$ – адрес позиции в ОЗУ и $A_{БРП}$ – адрес ячейки МБ. Через С как обычно обозначено содержимое сумматора, а через С [48÷37] – тетрада в нулевой позиции кода С. Символами

$$\alpha, \beta, \delta, \rho, \tau, a, m, n, A'_{БРП}, N, N_0$$

обозначены вспомогательные параметры Диспетчера.

Диспетчер оформлен в виде стандартной программы с двумя входами. Первому входу соответствует команда обращения I5 3I 00I05. Величина s требуемого резерва ОРП в этом случае задается в младших разрядах сумматора (старшие разряды сумматора должны быть обнулены. Второму входу соответствует команда обращения I5 3I 00700. Величина s в этом случае задается в младших разрядах ячейки 00247.

Диспетчер занимает в БСП ячейки 00700÷0II23. Он использует рабочие ячейки

00I70, 00I72, 00243÷0025I, 00257÷00262, 0III3÷0II23 (I0)

В частности, параметры N, α, β и δ хранятся в ячейках 00243÷00246. Содержимое ячейки 00243 в начале работы Диспетчера запоминается и перед выходом из него восстанавливается. Исходное состояние остальных рабочих ячеек Диспетчера не восстанавливается.

Диспетчер использует также ИР с номерами 0I и 07÷I7. Все эти ИР в конце работы Диспетчера восстанавливают свое исходное состояние. При этом исходные состояния первого и I5-го Ир запоминаются в адресных частях настроек 00I72 и 00I70 и, следовательно, корректируются вместе со всеми настройками. Необходимость коррекции содержимого первого ИР вытекает из способа его использования в Ридере. Необходимость коррекции содержимого

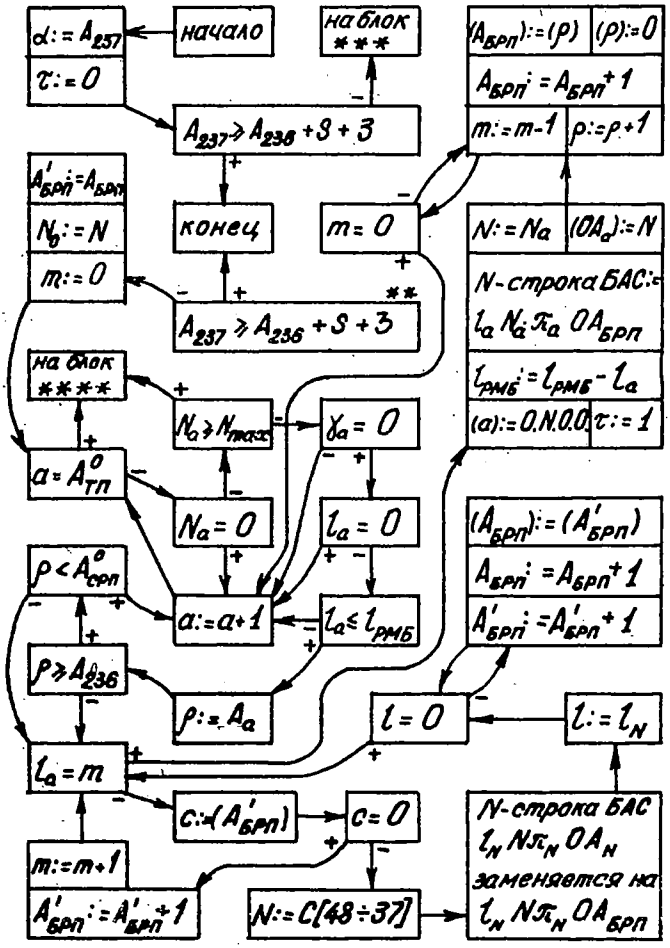

начало
на блок ***
конец
на блок ****
$\alpha := A_{257}$
$\tau := 0$
$A_{237} \geqslant A_{236} + 3 + 3$
$A'_{БРП} := A_{БРП}$
$N_0 := N$
$m := 0$
$m = 0$
$N := N_\alpha$
$(0A_\alpha) := N$
N-строка БАС:=
$l_\alpha N_\alpha \pi_\alpha 0A_{БРП}$
$l_{РМБ} := l_{РМБ} - l_\alpha$
$(\alpha) := 0.N.0.0$
$\tau := 1$
$N_\alpha \geqslant N_{max}$
$\gamma_\alpha = 0$
$\alpha = A^0_{ТП}$
$N_\alpha = 0$
$l_\alpha = 0$
$\rho < A^0_{СРП}$
$\alpha := \alpha + 1$
$l_\alpha \leq l_{РМБ}$
$\rho \geqslant A_{236}$
$\rho := A_\alpha$
$l = 0$
$l := l_N$
$l_\alpha = m$
$c := (A'_{БРП})$
$c = 0$
$m := m + 1$
$A'_{БРП} := A'_{БРП} + 1$
$N := C[48 \div 37]$
N-строка БАС
$l_N N \pi_N 0A_N$
заменяется на
$l_N N \pi_N 0A_{БРП}$
$(A_{БРП}) := (A'_{БРП})$
$A_{БРП} := A_{БРП} + 1$
$A'_{БРП} := A'_{БРП} + 1$
$(A_{БРП}) := (\rho)$
$(\rho) := 0$
$A_{БРП} := A_{БРП} + 1$
$m := m - 1$
$\rho := \rho + 1$

Рис. 9

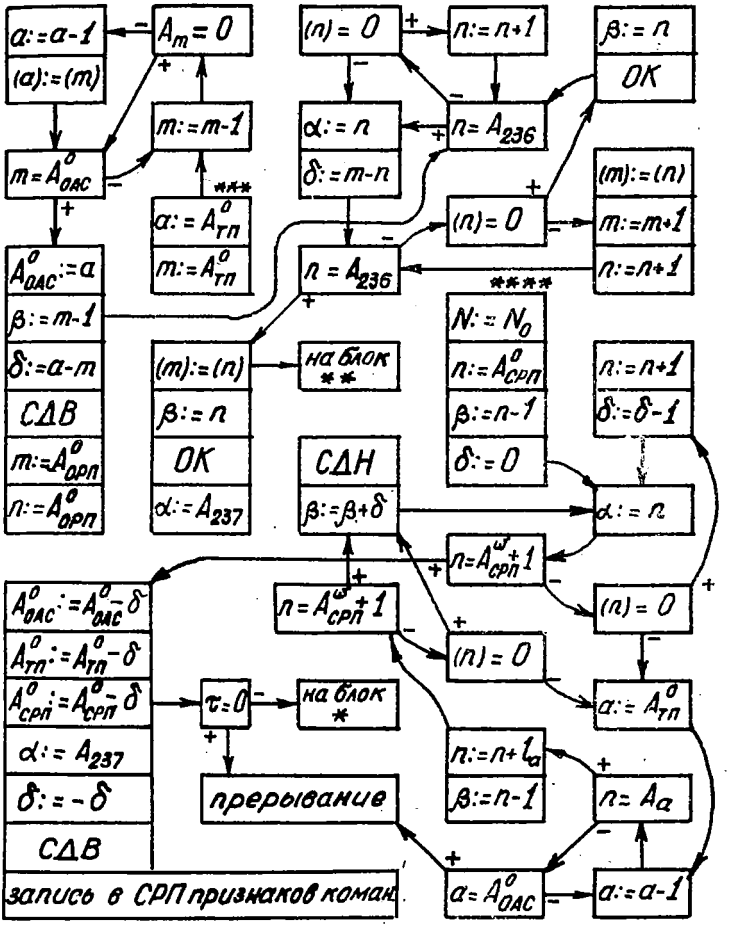

α:=α-1
(α):=(m)
A_m=0
m:=m-1
m=A^0_ОАС
α:=A^0_ТП
m:=A^0_ТП
A^0_ОАС:=α
β:=m-1
δ:=α-m
СДВ
m:=A^0_ОРП
n:=A^0_ОРП
(n)=0
n:=n+1
α:=n
δ:=m-n
n=A_236
(n)=0
n=A_236
(m):=(n)
на блок **
β:=n
ОК
α:=A_237
β:=n
ОК
(m):=(n)
m:=m+1
n:=n+1
N:=N_0
n:=A^0_СРП
β:=n-1
δ:=0
СДН
β:=β+δ
n:=A^ω_СРП+1
n:=A^ω_СРП+1
(n)=0
n:=n+1
δ:=δ-1
α:=n
A^0_ОАС:=A^0_ОАС-δ
A^0_ТП:=A^0_ТП-δ
A^0_СРП:=A^0_СРП-δ
α:=A_237
δ:=-δ
СДВ
τ:=0
на блок *
прерывание
(n)=0
α:=A^0_ТП
n:=n+1_α
β:=n-1
n=A_α
α=A^0_ОАС
α:=α-1
запись в СРП признаков коман

Рис.9а

I5-го ИР очевидна, поскольку в этом ИР содержится адрес возврата из Диспетчера, а обращение к Диспетчеру возможно и из операторов, находящихся в СРП.

Диспетчер корректирует основные параметры A^{o}_{OAC}, $A^{o}_{TП}$, $A_{БАС}$, τ, $A_{БРП}$ и ℓ РМБ. Включение Диспетчера возможно в период работы большинства стандартных программ и операторов.

§ 5. СТАНДАРТНЫЕ ПРОГРАММЫ.

Большинство стандартных программ, включенных в БСП, обусловлено принятым в Авто-Аналитике способом кодирования формул и распределения оперативной памяти. Сюда относятся прежде всего СП чтения и записи тетрад в заданные позиции ОРП, СП, предназначенные для работы с настройками, СП выдачи информации о формуле, аннулирования формулы, прописки формулы в ОАС и т.п. Некоторые СП обслуживают систему обмена между ОЗУ и МБ (четыре из них были упомянуты в описании Диспетчера) и систему ввода исходной информации с перфокарт и загрузки Авто-Аналитика и вывода той или иной информации на печать. В данном параграфе будут описаны лишь стандартные программы первого типа.

В качестве ИР возврата большая часть СП использует I5-й ИР. Поэтому команда обращения к СП в большинстве случаев имеет вид:

I5 3I - $\alpha_{СП}$,

где $\alpha_{СП}$ - адрес начала (или входа) данной СП. Иногда регистром возврата служит I4-й ИР, и тогда команда обращения имеет вид:

I4 3I $\alpha_{СП}$.

Вообще многие из СП используют три индексных регистра с номерами I4, I5 и I6. В ИР с номерами 0I÷I3 исходные коды сохраняются всеми СП.

Почти все СП преобразуют, анализируют или как-либо иначе используют информацию из рабочих ячеек 00240÷00247 и 02046. Поэтому естественно объявить эти ячейки основными рабочими ячейками Авто-Аналитика и условиться об обозначении хранящихся в них кодов. Мы будем применять следующие обозначения:

Л - тетрада в 3-ей позиции ячейки 00240;
М - тетрада в 3-ей позиции ячейки 00241;
П - Тетрада в 3-ей позиции ячейки 00242;

N – тетрада в I-й позиции ячейки 00243;

N_1 – тетрада в I-й позиции ячейки 02046 (остальные позиции в ячейках 00240÷00243 и 02046 всегда предполагаются обнуленными);

S – неотрицательное целое число в I5-ти младших разрядах ячейки 00247;

α – неотрицательное целое число (адрес ячейки ОЗУ) в I5-ти младших разрядах ячейки 00244;

β – неотрицательное целое число (адрес ячейки ОЗУ) в I5-ти младших разрядах ячейки 00245 (старшие разряды 48÷I6 ячеек 00244, 00245 и 00247 всегда должны быть нулевыми);

δ – целое 48-разрядное двоичное число в ячейке 00246, причем если $\delta \geq 0$, то оно задается в прямом коде, а если $\delta < 0$, то – в обратном коде, получающемся из прямого кода числа $|\delta|$ путем поразрядного сравнения с константой

7777 7777 7777 7777 (II)

(в частности, число $\delta = 0$ может быть представлено как чисто нулевым кодом, так и кодом (II); числа δ = I, -I, -2, ... задаются в виде констант

0000 0000 0000 000I
7777 7777 7777 7776
7777 7777 7777 7775

и т.п.);

H – адрес константы настройки, хранящийся в ИР-I6;

i A – адрес позиции, на которую нацелена константа настройки, хранящаяся в ячейке H;

(i A) – тетрада, записанная в позиции с адресом iA;

A_H – адресная часть настройки H (точнее, число, записанное в I5-ти младших разрядах ячейки H);

i – номер позиции в ячейке A_H, на которую нацелена настройка H;

$\varkappa$ – адрес ячейки ОЗУ, записанный в ИР возврата СП в момент обращения к этой СП (как правило, СП, окончив свою работу, передает управление в ячейку $\varkappa$; исключения из данного правила всегда явно оговариваются).

(α) СП "ПРОВЕРКА КОММУТАТИВНОСТИ ИЛИ АССОЦИАТИВНОСТИ". Эта СП проверяет, выполнено или нет одно из соотношений

$A \in \Sigma_K$, $M \in \Sigma_K$, $П \in \Sigma_K$, $C \in \Sigma_K$, $Л \in \Sigma_a$, $М \in \Sigma_a$, $П \in \Sigma_a$, $С \in \Sigma_a$, причем каждому из данных соотношений соответствует особый вход. СП действует на основании шкал 00263 и 00264. Если проверяемое условие выполнено (т.е. исходная тетрада Q равная Л,М,П или С, меньше, чем 0100, и отвечающий ей K-й, $K = Q - 0077_8$, разряд шкалы 00263 или 00264 равен I), то на выходе $\omega = I$. В противном случае значение сигнала ω на выходе полагается равным нулю.

Данная СП посылает исходную тетраду Q в ИР-I4, не изменяет прочие ИР, не имеет рабочих ячеек и не включает Диспетчер. В случае условий $C \in \Sigma_K$ и $C \in \Sigma_a$ тетрада $Q = C$ берется из 3-ей позиции сумматора; старшие разряды 48 ÷ I3 сумматора в этом случае должны быть обнулены. Прочая информация о работе СП содержится в таблице 8.

Проверка условия	Исходная информ.	Обращение		Значения на выходе				
		ИР	α СП	$\omega = 0$	Сумматор	ИР-I4	ИР-I5	ИР-I6
$Л \in \Sigma_K$	Л	I5	00021	$Л \notin \Sigma_K$	Если проверяемое условие выполнено, то 0000 0000 0001 Иначе если исходная тетрада $Q < 0100$, то 0000 0000 0000 0000 если же $Q \geq 0100$, то 4000 0000 0000 0000 ($Q - 0100$)	Л	∞	Сохраняется
$М \in \Sigma_K$	М	I5	00022	$М \notin \Sigma_K$		М	∞	
$П \in \Sigma_K$	П	I5	00023	$П \notin \Sigma_K$		П	∞	
$С \in \Sigma_K$	С	I5	00027	$С \notin \Sigma_K$		С	∞	
$Л \in \Sigma_a$	Л	I5	00024	$Л \notin \Sigma_a$		Л	∞	
$М \in \Sigma_a$	М	I5	00025	$М \notin \Sigma_a$		М	∞	
$П \in \Sigma_a$	П	I5	00026	$П \notin \Sigma_a$		П	∞	
$С \in \Sigma_a$	С	I5	00032	$С \notin \Sigma_a$		С	∞	

Таблица 8

(β) СП ГРУППЫ "ЧТЕНИЕ". К этой группе СП относятся три СП: "Потетрадное чтение", "Чтение вне скобок" и "Чтение числа", каждая из которых обладает несколькими входами. Информация об этих стандартных программах содержится в таблице 9.

СП "ПОТЕТРАДНОЕ ЧТЕНИЕ". Исходной информацией для данной СП служит константа настройки, адрес Н которой указан в ИР-I6. В зависимости от входа данная СП перестраивает настройку Н с по-

зиции i A на следующую позицию i A + I или на предыдущую позицию i A - I. Возможен также вариант без перестройки настройки H. Затем тетрада из позиции, определяемой новым состоянием настройки H, поступает в третью позицию сумматора, а остальные позиции сумматора обнуляются. В конце работы СП содержимое сумматора - в зависимости от входа - может быть записано в одну из ячеек 00240, 00241 и 00242.

Предусмотрено 9 различных входов в СП и в соответствии с этим установилось 9 различных названий данной по сути дела единой СП. А именно, модификация "Чтение вперед Л" сначала перестраивает настройку H на следующую позицию, затем извлекает из этой позиции тетраду и засылает ее в сумматор и в ячейку 00240. Модификации "Чтение вперед М" и "Чтение вперед П" действует аналогично, но вместо ячейки 00240 используют соответственно ячейки 00241 и 00242. Модификация "Чтение вперед С" представляет собой общую часть трех предыдущих модификаций; она ограничивается перестройкой настройки H "вперед" и извлечением тетрады в сумматор. Модификации "Чтение назад Л", "Чтение назад М", "Чтение назад П" и "Чтение назад С" аналогичны предыдущим модификациям с той лишь разницей, что предварительная перестройка настройки H производится не на следующую, а на предыдущую позицию. Наконец, модификация "Чтение без перестройки" ограничивается пересылкой в сумматор тетрады из позиции, на которую указывает настройка H в ее исходном состоянии.

СП "Потетрадное чтение" со всеми ее модификациями хорошо приспособлена для последовательного чтения элементарных символов, составляющих формулу в ОРП, причем имеется возможность "двигаться" вдоль формулы как в прямом, так и в обратном направлении, и само это направление можно менять в любой момент времени. Используя несколько настроек, можно одновременно анализировать и сопоставлять различные участки одной или нескольких формул.

Три модификации данной СП, ограничивающиеся запоминанием результата в сумматоре, в качестве ИР возврата используют ИР-I4, а все остальные - ИР-I5. Значение сигнала ω на выходе равно нулю тогда и только тогда, когда из ОРП считан код граничного символа Δ . СП "Потетрадное чтение" рабочих ячеек не имеет и к Диспетчеру не обращается.

Шифр	Название	Исходная информация	Обращение ИР	Обращение α СП	Значения на выходе $w=0$	Значения на выходе С	Значения на выходе ИР-I4	Значения на выходе ИР-I5	Значения на выходе ИР-I6	Рабочие ячейки	Алгоритм работы
ЧВЛ	Чтение вперед Л	Н	I5	00041	Если из ОЗУ считано Δ или число, равное нулю	Результат	00036	$\varkappa$	Н	нет	$iA:=iA+I;C:=(iA);Л:=C$
ЧВМ	Чтение вперед М	Н	I5	00042			00037	$\varkappa$	Н	нет	$iA:=iA+I;C:=(iA);М:=C$
ЧВП	Чтение вперед П	Н	I5	00043			00040	$\varkappa$	Н	нет	$iA:=iA+I;C:=(iA);П:=C$
ЧВС	Чтение вперед С	Н	I4	00047			$\varkappa$	Сохр.	Н	нет	$iA:=iA+I;C:=(iA)$
ЧНЛ	Чтение назад Л	Н	I5	00044			00036	$\varkappa$	Н	нет	$iA:=iA-I;C:=(iA);Л:=C$
ЧНМ	Чтение назад М	Н	I5	00045			00037	$\varkappa$	Н	нет	$iA:=iA-I;C:=(iA);М:=C$
ЧНП	Чтение назад П	Н	I5	00046			00040	$\varkappa$	Н	нет	$iA:=iA-I;C:=(iA);П:=C$
ЧНС	Чтение назад С	Н	I4	00052			$\varkappa$	Сохр.	Н	нет	$iA:=iA-I;\ C:=(iA)$
ЧТС	Чтение без перестройки	Н	I4	00051			$\varkappa$	Сохр.	Н	нет	$C:=(iA)$
ЧВВС	Чтение вперед вне скобок	Н	I5	01400		Результат	01402	00000	Н	00252, 00253	См. рис.II
ЧНВС	Чтение назад вне скобок	Н	I5	01412			01414	00000	Н	-"-	См. рис.I2
ЧЧВ	Чтение числа вперед	Н	I5	01472			01460	$\varkappa$	Н	00252+00254, 00260, 01423	См.описание в тексте
ЧЧН	Чтение числа назад	Н	I5	01475			01460	$\varkappa$	Н	-"-	См.описание в тексте

Таблица 9

СП "ЧТЕНИЕ ВНЕ СКОБОК". Исходной информацией служит адрес настройки Н. СП имеет две модификации: "Чтение вперед вне скобок" и "Чтение назад вне скобок". СП "Чтение вперед вне скобок" начинает свою работу с обращения к СП "Чтение вперед С". Если считанная тетрада отлична от кода раскрывающей скобки, то она выдается в качестве результата. Иначе настройка Н перегоняется через выражение, стоящее в скобках, и настраивается на соответствующую закрывающую скобку. Если вслед за этой закрывающей скобкой находится тетрада, отличная от кода раскрывающей скобки, то она выдается в качестве результата. В противном случае повторяется перегон настройки Н через новое выражение в скобках и т.д. Алгоритм работы данной модификации изображен на рис.10. СП "Чтение назад вне скобок" работает аналогично. Алгоритм ее работы см. на рис.11.

Обе модификации выдают результат в третьих позициях сумматора и ячейки 00253 (остальные позиции обнуляются), настройку Н нацеливают на ту позицию, из которой извлекли свой результат, и к Диспетчеру не обращаются. Сигнал ω на выходе равен нулю лишь в том случае, когда результат совпадает с кодом граничного символа Δ. Следует отметить, что при неправильной расстановке ско-

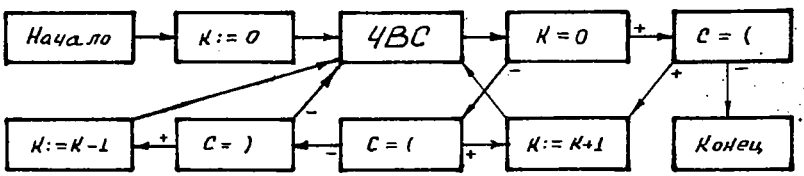


Рис. 10. Алгоритм СП ЧВВС

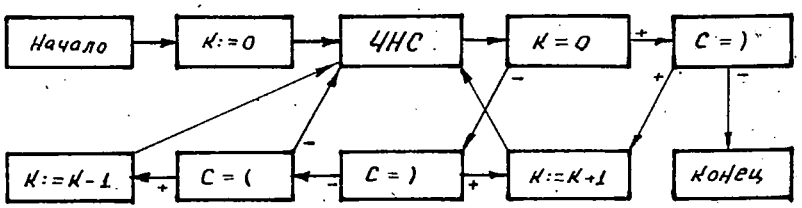


Рис. 11. Алгоритм СП ЧНВС

бок в формуле возможно зацикливание.

СП "ЧТЕНИЕ ЧИСЛА". Исходной информацией служит адрес настройки Н. Имеются две модификации "Чтение числа вперед" и "Чтение числа назад". Первая из них предполагает, что код

$$Q_1\ Q_2\ \ldots\ Q_\Delta \qquad (I2)$$

некоторого числа Ξ начинается с позиции i A + I и занимает подряд $\Delta > 0$ позиций, причем тетрады Q_1, Q_2 и т.д. записаны соответственно в позициях i A + I, i A + 2 и т.д. Вторая модификация предполагает, что код (I2) числа Ξ записан начиная с позиции i A – I "назад", т.е. тетрада Q_1 в позиции i A – I, тетрада Q_2 в позиции i A – 2 и т.д.

Хотя обычно в Авто-Аналитике код числа состоит из четырех тетрад, рассматриваемая СП выдает верный результат и при ином количестве тетрад в коде (I2). Результатом в обеих модификациях является 48-разрядный код, старшие II разрядов которого совпадают с младшими разрядами тетрады Q_1, следующие II разрядов – с младшими разрядами тетрады Q_2 и т.д. В случае $1 \leq \Delta \leq 4$ младшие разряды результата обнуляются. В случае $5 \leq \Delta \leq$ II "лишние" разряды кода (I2) (а именно, семь младших разрядов тетрады Q_5 и все разряды тетрад Q_6, Q_7, ...) отбрасываются. В случае $\Delta \geq$ I2 результат вообще неверен. В случае $\Delta = 0$ (т.е. в том случае, когда в позиции i A + I (соответственно, в позиции i A–I) находится тетрада $Q < 4000$) результат полагается равным нулю.

В любом случае результат поступает в сумматор и в ячейки 00253 и 01423. Константа настройки Н нацеливается на позицию iA + Δ + I, следующую за позицией iA + Δ (соответственно, на позицию iA – Δ – I, предшествующую позиции iA – Δ) с последней тетрадой Q_Δ кода (I2). Диспетчер не включается. Сигнал ω = = 0 свидетельствует о чисто нулевом результате.

(c) СП ГРУППЫ "ЗАПИСЬ". К этой группе относятся четыре СП: "Потетрадная запись", "Запись числа", "Групповая запись первого типа" и "Групповая запись второго типа". Все они предназначены для записи новых тетрад в ОРП и образования из них новых формул. Каждая СП группы "Запись" обладает несколькими модификациями и каждая модификация может вызвать включение Диспетчера. Информация об этих СП содержится в таблице I0. Обозначение РЯД означает "рабочие ячейки Диспетчера".

Шифр	Название	Исходн. информация	Обращение			
			ИР	α СП	α СП^{236}	α СП^{237}
ЗВЛ	Запись вперед Л	Н,Л	15	00061	00561	-
ЗВМ	Запись вперед М	Н,М	15	00062	00562	-
ЗВП	Запись вперед П	Н,П	15	00063	00563	-
ЗВС	Запись вперед С	Н,С	15	00067	00567	-
ЗНЛ	Запись назад Л	Н,Л	15	00064	-	00564
ЗНМ	Запись назад М	Н,М	15	00065	-	00565
ЗНП	Запись назад П	Н,П	15	00066	-	00566
ЗНС	Запись назад С	Н,С	15	00072	-	00572
ЗПС	Запись без перестройки	Н,С	15	00071	00570	00571
ЗЧВ	Запись числа вперед	Н,С	15	01441	01447	-
ЗЧН	Запись числа назад	Н,С	15	01444	-	01452
ГЗВ1	Групповая запись вперед первого типа	Н	15	01147	01145	-
ГЗН1	Групповая запись назад первого типа	Н	15	01150	-	01146
ГЗВ2	Групповая запись вперед второго типа	Н	15	01163	01161	-
ГЗН2	Групповая запись назад второго типа	Н	15	01164	-	01162

Таблица 10

Значения на выходе					Рабочие ячейки	Алгоритм работы
ω = 0	C	ИР-14	ИР-15	ИР-16		
Если было обращение к Диспетчеру и в 17-том ИР содержится нуль	?	?	æ	H	00260+РЯД	iA:=iA+1;(iA):=Л; БОР
	?	?	æ	H	-"-	iA:=iA+1;(iA):=М; БОР
	?	?	æ	H	-"-	iA:=iA+1;(iA):=П; БОР
	?	?	æ	H	-"-	iA:=iA+1;(iA):=С; БОР
	?	?	æ	H	-"-	iA:=iA-1;(iA):=Л; БОР
	?	?	æ	H	-"-	iA:=iA-1;(iA):=М; БОР
	?	?	æ	H	-"-	iA:=iA-1;(iA):=П; БОР
	?	?	æ	H	-"-	iA:=iA-1;(iA):=С; БОР
	?	?	æ	H	-"-	(iA):=С; БОР
= 0 всегда	нуль	?	01433	H	00173 00260 01423 01424 01425 РЯД	Число С кодируется 4-мя тетрадами и эти тетрады пишутся по настройке Н вперед (назад)
	-"-	?	01433	H		
	-"-	01156	01155	00173	00173 00255 00256 00260 РЯД	См. описание в тексте
	-"-	01156	01155	00173		См.описание в тексте
четное число информ. слов	?	H	?	H	00173 00256 00260 01205 РЯД	См. описание в тексте
	?	H	?	H		См. описание в тексте

Таблица 10а

СП "ПОТЕТРАДНАЯ ЗАПИСЬ". Исходной информацией служат адрес настройки Н и тетрада Л, М, П или тетрада С из третьей позиции сумматора. Сначала СП перестраивает настройку Н с позиции i А на следующую позицию i А + I или на предыдущую позицию i А – I (в зависимости от входа; возможен вариант без перестройки настройки Н). Затем тетрада Л, М, П или С записывается в позицию ОРП, адрес которой указывается новым состоянием настройки Н. После этого включается блок обеспечения резерва (БОР). Этот блок сравнивает адреса позиций, на которые нацелены настройки 00236 и 00237. Если выполнено равенство

$$i_{236}\, A_{236} + 4 = i_{237}\, A_{237}, \qquad (13)$$

то производится обращение к Диспетчеру при $S = 0$.

Основным модификациям рассматриваемой СП присвоены следующие имена: "Запись вперед Л", "Запись вперед М", "Запись вперед П", "Запись вперед С", "Запись назад Л", "Запись назад М", "Запись назад П", "Запись назад С" и "Запись без перестройки". Работа первых восьми модификаций ясна из их наименований (см.также таблицу 10). Модификация "Запись без перестройки" пишет тетраду С в позицию i А и передает управление на БОР.

Хотя все СП записи имеют однокомандное обращение

$$15 \quad 31 \quad \alpha_{СП}, \qquad (14)$$

фактически приходится писать две команды

16 24 Н
15 31 $\alpha_{СП}$,

из которых первая есть команда засылки адреса Н в ИР-16. Поскольку запись вперед чаще всего производится при помощи настройки 00236, то для первых четырех модификаций предусмотрены дополнительные входы $\alpha_{сп}^{236}$, при использовании которых предварительная засылка адреса 00236 в ИР-16 не требуется (она производится в ячейке $\alpha_{сп}^{236}$, после чего осуществляется переход на основной вход модификации). Аналогично для записи назад обычно используется настройка 00237 и поэтому в следующих четырех модификациях предусмотрены дополнительные входы $\alpha_{сп}^{237}$, при использовании которых не нужна предварительная засылка адреса 00237 в ИР-16. Модификация "Запись без перестройки" имеет два дополнительных входа $\alpha_{сп}^{236}$, $\alpha_{сп}^{237}$ с аналогичными функциями.

СП "Потетрадная запись" позволяет последовательно записывать в свободное место ОРП тетрады вновь получаемых формул и при использовании настроек 00236 и 00237 избавляет пользователя от необходимости следить за наличием этого свободного места.

Все модификации сохраняют тетрады Л, М и П, но затирают содержимое сумматора. Прочая информация об СП ясна из таблицы 10.

СП "ЗАПИСЬ ЧИСЛА". Исходной информацией служат адрес настройки Н и 48-разрядный код С в сумматоре. Имеются две модификации "Запись числа вперед" и "Запись числа назад", снабженные дополнительными входами $\alpha_{сп}^{236}$ и $\alpha_{сп}^{237}$ для работы с настройками 00236 и 00237 без предварительной фиксации адресов 00236 и 00237 в ИР-16. Обе модификации могут обратиться к Диспетчеру. СП "Запись числа вперед" (соответственно "Запись числа назад") кодирует содержимое сумматора С в виде последовательности $Q_1 Q_2 Q_3 Q_4$ из четырех тетрад по правилу, описанному в § 2, и затем эти тетрады последовательно пишутся по настройке Н при помощи СП "Запись вперед С" (соответственно, "Запись назад С"). Содержимое сумматора не сохраняется. Настройка Н к концу работы СП оказывается нацеленной на позицию ОРП, в которую записалась последняя тетрада Q_4 .

СП "ГРУППОВАЯ ЗАПИСЬ ПЕРВОГО ТИПА". Исходной информацией служит адрес настройки Н. Имеются две модификации: "Групповая запись вперед первого типа" и "Групповая запись назад первого типа", снабженные дополнительными входами для работы с настройками 00236 и 00237 без предварительной фиксации адресов 00236 и 00237 в ИР-16. Обе модификации могут обратиться к Диспетчеру.

Обращение к СП осуществляется командой (14), которая может занимать как левую, так и правую половину ячейки $\alpha - 1$. В первом случае содержимое правой половины ячейки $\alpha - 1$ не играет роли. В ячейках α, $\alpha + 1$, ... располагается последовательность тетрад

$$Q_1 Q_2 \ldots Q_{n-1}, \quad n \geq 1$$

по четыре тетрады в каждой ячейке; первые $n - 1$ тетрад отличны от кода граничного символа Δ, а последняя тетрада Q_n совпадает с кодом 0000. СП пишет по настройке Н при помощи СП "Запись

вперед С" (соответственно, "Запись назад С") тетрады из 0-й, 1-й, 2-й и 3-й позиций ячейки $\varkappa$, затем из 0-й, 1-й, 2-й и 3-й позиций ячейки $\varkappa + 1$ и т.д. до тех пор, пока не встретится тетрада 0000. Эта тетрада по настройке Н не пишется. Возврат из СП осуществляется в ячейку, следующую за той, из которой выбрана тетрада 0000. Настройка Н в результате работы СП нацеливается на ту позицию ОРП, куда записалась последняя ненулевая тетрада a_{n-1}.

СП "ГРУППОВАЯ ЗАПИСЬ ВТОРОГО ТИПА". Исходной информацией служит адрес настройки Н. Имеются две модификации: "Групповая запись вперед второго типа" и "Групповая запись назад второго типа". Для работы с настройками 00236 и 00237 имеются дополнительные входы, при использовании которых предварительная фиксация адресов 00236 и 00237 в ИР-16 не требуется. Обе модификации могут обратиться к Диспетчеру.

Обращение к СП осуществляется командой (14) в некоторой ячейке $\varkappa - 1$. В следующих ячейках $\varkappa$, $\varkappa + 1$, ... располагается последовательность 24-разрядных информационных слов вида

j 32 B или j 33 B

(j занимает четыре разряда, В – пятнадцать разрядов) по два слова в каждую ячейку; если количество информационных слов нечетное, то последнее слово занимает левую половину ячейки. СП просматривает эти информационные слова по порядку и для каждого из них по настройке Н при помощи СП "Запись вперед С" (соответственно "Запись назад С") пишет некоторую тетраду Q. Если информационное слово имеет вид j 32 В, то тетрада Q берется из 12-ти младших разрядов кода

$$B_{исп} = \{B + И_j\} \bmod 2^{15}$$

где $И_j$ – содержимое ИР с номером j. В случае слова j 33В тетрада Q берется из третьей позиции ячейки $B_{исп}$ (содержимое остальных позиций ячейки $B_{исп}$ безразлично). Возврат из СП осуществляется в ячейку, следующую за той, из которой выбрано последнее информационное слово. Настройка Н в результате работы СП нацеливается на ту позицию ОРП, куда была записана последняя тетрада Q.

(d) СП ГРУППЫ "РАБОТА С НАСТРОЙКАМИ". В данную группу включены СП, специально предназначенные для облегчения работы с

константами настроек: "Фиктивное чтение", "Перевод настройки", "Подвод настройки", "Упаковка настроек" и "Восстановление настроек". Информация об этих СП содержится в таблице II.

СП "ФИКТИВНОЕ ЧТЕНИЕ". Исходной информацией служит адрес настройки Н. Имеются две модификации: "Фиктивное чтение вперед" и "Фиктивное чтение назад". СП переводит настройку Н с позиции i А на следующую позицию i А + I (соответственно на предыдущую позицию i А - I). От СП "Чтение вперед С" ("Чтение назад С") отличается меньшим временем работы. Диспетчер не включается даже в том случае, когда Н = 00236 или 00237.

Поэтому для работы с настройками 00236 и 00237 рекомендуются особые входы $\mathcal{L}_{сп}^{236}$ и $\mathcal{L}_{сп}^{237}$. В этом случае после перевода настройки 00236 на следующую (настройки 00237 на предыдущую) позицию происходит обращение к блоку обеспечения резерва, который при необходимости включает Диспетчер.

Следует также отметить, что в качестве ИР возврата используется ИР-I4 при основных входах и ИР-I5 при особых входах.

СП "ПЕРЕВОД НАСТРОЙКИ". Исходная информация - адрес настройки Н и содержимое сумматора С. Код С интерпретируется как неотрицательное целое число K, записанное в младших разрядах сумматора. Старшие разряды должны содержать нули. СП имеет две модификации "Перевод настройки вперед" и "Перевод настройки назад". СП осуществляет перевод настройки Н с позиции i А на позицию i А + K (соответственно на позицию i А - K). Однократное обращение к данной СП по характеру преобразования настройки Н равносильно K -кратному обращению к СП "Фиктивное чтение", однако требует существенно меньшего времени. Обращения к Диспетчеру нет. Поэтому СП "Перевод настройки" опасно применять в случаях Н = 00236 и Н = 00237.

СП "ПОДВОД НАСТРОЙКИ" предназначена для оформления перехода от записи одной формулы в ОРП к записи следующей формулы. Напомним, что каждая формула в ОРП должна начинаться с нулевой позиции какой-нибудь ячейки (в эту позицию записывается начальный граничный символ Δ формулы). Если при этом запись конечного Δ одной формулы произошла в нулевую позицию очередной ячейки, то в ту же позицию заносится начальное Δ следующей формулы. В противном случае позиции между конечным Δ первой формулы и на-

Шифр	Название	Исходная информация	Обращение				Значения на выходе					Рабочие ячейки
			ИР	$\alpha_{СП}$	$\alpha_{СП}^{\varepsilon 36}$	$\alpha_{СП}^{\varepsilon 37}$	$\omega=0$	Сумматор	ИР-I4	ИР-I5	ИР-I6	
ЧВФ	Фиктивное чтение вперед	H	I4	00057	–	–	Всегда	Новое (H)	∝	Сохр.	H	нет
		–	I5	–	00367	–	?	?	00I03	∝	00236	РЯД
ЧНФ	Фиктивное чтение назад	H	I4	00I06	–	–	Всегда	Новое (H)	∝	Сохр.	H	нет
		–	I5	–	–	00371	?	?	00I03	∝	00237	РЯД
A+C	Перевод настройки вперед	H,C	I5	00III	–	–	Всегда	нуль	?	∝	H	00260
A–C	Перевод настройки назад	H,C	I5	00II7	–	–	Всегда	нуль	?	∝	H	00260
ПОДВ	Подвод настройки вперед	H	I5	00356	00355	–	Никогда	?	?	?	H	00I73+ РЯД
ПОДН	Подвод настройки назад	H	I5	00363	–	00362	Никогда	?	?	?	H	–"–
КФI	Конец формулы в ОРП I	–	I5	–	0035I	–	Никогда	?	?	?	00236	–"–
КФ2	Конец формулы в ОРП II	–	I5	–	–	00373	Никогда	?	?	?	00237	–"–
УПАК	Упаковка настроек	H	I5	0II24	–	–	Никогда	0000 0000 A	00000	0II33	H	–"–
РПАК	Восстановление настроек	H	I5	0II37	–	–	Всегда	нуль	?	∝	00000	нет
КН	Коррекция настроек	α,β,δ	I5	00323	–	–	Никогда	?	Сохр.	∝	00000	нет
ОК	Общая коррекция	α,β,δ	I5	003I2	–	–	Никогда	?	Сохр.	∝	00000	нет
СДВ	Сдвиг вперед	α,β,δ	I5	00305	–	–	Никогда	?	Сохр.	∝	00000	нет
СДН	Сдвиг назад	α,β,δ	I5	00300	–	–	Никогда	?	Сохр.	∝	00000	нет
СДВИГ	Сдвиг участка ОРП	α,β,δ	I5	I222	–	–	Никогда	?	Сохр.	∝	00000	нет

Таблица II

чальным Δ второй формулы должны быть пропущены (или заполнены, например, кодом символа Δ).

Запись информации в ОРП П в общем случае не регламентируется, однако и здесь чаще всего используется тот же принцип. Поскольку ОРП П образуется при помощи СП "Запись назад" по настройке 00237, и тетрады формулы как правило размещаются в порядке, противоположном обычному, то естественно предполагать, что начальное Δ произвольной формулы в ОРП П размещается в третьей позиции некоторой ячейки А (а следующие по порядку тетрады формулы пишутся в позиции 2А, 1А, 0А, 3(А-1), 2(А-1) и т.д. Если конечное Δ данной формулы попадает в третью позицию какой-либо ячейки В, то с этой позиции начинается запись "назад" новой формулы в ОРП П. В противном случае оставшиеся позиции ячейки В забиваются тетрадой Δ (или иными тетрадами) и новая формула расписывается начиная с 3-ей позиции ячейки В-1.

Для упрощения этих действий по переходу от записи одной формулы к записи следующей формулы и предусмотрена СП "Подвод настройки". Она имеет четыре модификации "Подвод настройки вперед", "Подвод настройки назад", "Конец формулы в ОРП I" и "Конец формулы в ОРП П".

Первые две модификации в качестве исходной информации используют адрес настройки Н. Для работы с настройками 00236 и 00237 предусмотрены дополнительные входы, при использовании которых предварительная установка Н = 00236 или Н = 00237 не требуется. СП "Подвод настройки вперед" (соответственно "Подвод настройки назад") проверяет условие $i = 0$ (соответственно $i = 3$) для адреса iА позиции, на которую указывает настройка Н. Если это условие выполнено, то производится выход из СП. Иначе по настройке Н при помощи СП "Запись вперед С" ("Запись назад С") пишется тетрада Δ и осуществляется переход на начало СП.

Модификация "Конец формулы в ОРП I" ("Конец формулы в ОРПП") работает только с настройкой 00236 (соответственно, 00237). Поэтому исходная информация для нее не нужна. Данная модификация полагает Н:=00236 (Н:=00237) и считывает тетраду Q при помощи СП "Чтение без перестройки", т.е. из позиции iА. Если $Q \neq \Delta$, то при помощи СП "Запись вперед С" ("Запись назад С") по настройке Н пишется тетрада Δ и затем осуществляется переход на начало СП "Подвод настройки вперед" ("Подвод настройки назад"). Если

же $Q = \Delta$, то указанный переход производится сразу.

Каждая модификация СП "Подвод настройки" может обратиться к Диспетчеру.

СП "УПАКОВКА НАСТРОЕК" И "ВОССТАНОВЛЕНИЕ НАСТРОЕК"

Аппарат настроек представляет собой основное и незаменимое орудие для анализа и преобразования формул в условиях их непрерывного движения по оперативной памяти. Поэтому естественно возникает задача строжайшей экономии настроек. В частности было решено, что во всех операторах Авто-Аналитика корректируемые настройки свободного пользования (т.е. настройки 00176 ÷ 00235) не должны затираться. Но обойтись без настроек операторы не могут. Следовательно, нужен способ как-то запоминать часть настроек в начале работы оператора и восстанавливать их в конце. Поскольку "законсервированные" настройки также должны корректироваться при перемещениях формул по памяти, то запомнить настройки можно лишь единственным способом – присоединить их к ОАС в виде его "фиктивных" строк. Именно для этой цели и был зарезервирован в свое время номер $N = 0000$. В любой момент времени в ОАС могут находиться "упакованные в нем" настройки, причем таковыми являются те и только те строки ОАС, у которых на месте номера N стоит нуль. Поскольку строки ОАС упорядочены по возрастанию их номеров, то все его строки с нулевым номером, т.е. все упакованные в нем настройки занимают некоторое количество подряд расположенных ячеек в начале ОАС. Для того, чтобы упакованные в ОАС настройки не были потеряны в момент работы второго этапа Диспетчера, приходится предполагать, что среди корректируемых настроек свободного пользования нет строк с нулевой адресной частью. В исходном состоянии БСП, поступившем в ОЗУ с магнитной ленты, все корректируемые настройки "забиты" кодами с ненулевой адресной частью и обнуленными разрядами 48 ÷ 16. От пользователя требуется, чтобы ни один блок его собственной программы не обнулял адресную часть ни одной из настроек 00176 ÷ 00235.

Для запоминания настроек и их последующего восстановления составлены две СП: "Упаковка настроек" и "Восстановление настроек". Исходной информацией для обеих СП является адрес настройки H.

СП "Упаковка настроек" упаковывает в ОАС настройки с адресами H, H + I, ... , 00235. Для этого она путем обращения к Диспет-

черу обеспечивает себе резерв ОРП объемом не менее, чем в

$$S = 236_8 - H$$

ячеек. Затем при помощи СП "Сдвиг назад" (см.ниже) сдвигается ОРП II на S ячеек в сторону ОРП I, после чего основной параметр A^0_{OAC} уменьшается на S и содержимое ячеек

$$H, H + I, \ldots, H + S - I = 00235$$

переписывается соответственно в ячейки

$$A^0_{OAC}, A^0_{OAC} + 1, \ldots, A^0_{OAC} + S - 1.$$

СП "Восстановление настроек" восстанавливает исходное состояние настроек с адресами H, H + I, ... , 00235. При этом данная СП предполагает, что соответствующее обращение к СП "Упаковка настроек в ОАС" происходило при том же самом значении H. Работа СП заключается в переписывании первых S = 236 - H ненулевых строк ОАС в ячейки H, H + I, ... , 00235 и последующем обнулении переписанных строк ОАС. Алгоритм работы СП изображен на рис. I2. Включение Диспетчера невозможно.

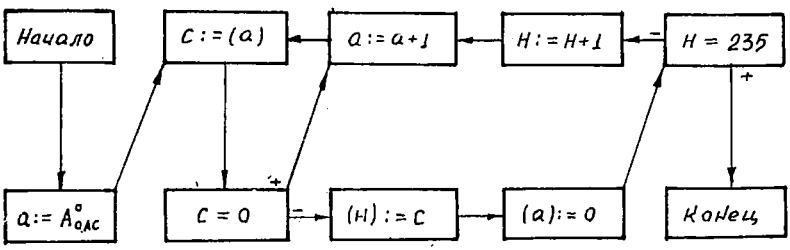


Рис. I2. Алгоритм СП РПАК

Еще раз подчеркнем, что в начале каждого оператора, использующего корректируемые настройки, должно стоять обращение к СП "Упаковка настроек", а в конце - обращение к СП "Восстановление настроек", причем оба обращения должны быть произведены при одном и том же значении H.

(*е*) СП ГРУППЫ "СДВИГ". В данную группу отнесены следующие СП: "Коррекция настроек", "Общая коррекция", "Сдвиг вперед",

"Сдвиг назад" и "Сдвиг участка ОРП". Информация об этих СП приведена в таблице II.

Исходной информацией для всех СП группы "Сдвиг" служат начальный и конечный адреса α и β сдвигаемого массива формул (или иной информации в ОРП), а также величина сдвига δ ; при сдвиге вперед, т.е. в сторону увеличения адресов, δ должно быть положительным и заданным в младших разрядах ячейки 00246 прямым кодом, а при сдвиге назад, т.е. в сторону уменьшения адресов, δ должно быть отрицательным и заданным в младших разрядах ячейки 00246 обратным кодом (см. обозначения в начале параграфа). Правильная работа СП данной группы гарантируется лишь при условии $\alpha \leq \beta$.

Если направление сдвига ("вперед" или "назад") известно, то следует обращаться к соответствующей СП "Сдвиг вперед" или "Сдвиг назад". В тех же довольно редких случаях, когда направление сдвига заранее предсказать нельзя, приходится использовать СП "Сдвиг участка ОРП".

СП "Сдвиг вперед", "Сдвиг назад" и "Сдвиг участка ОРП" переносят содержимое ячеек α , $\alpha + 1$, ... , β соответственно в ячейки $\alpha + \delta$, $\alpha + \delta + 1$, ... , $\beta + \delta$. При этом СП "Сдвиг вперед" начинает переписывание с конца сдвигаемого массива, СП "Сдвиг назад" – с начала сдвигаемого массива, а СП "Сдвиг участка ОРП" определяет знак величины сдвига δ (по содержимому 48-го разряда) и передает управление на одну из первых двух СП. По окончании сдвига обе СП "Сдвиг вперед" и "Сдвиг назад" передают управление на начало СП "Общая коррекция".

СП "Общая коррекция" просматривает ОАС и ТП, вылавливает все их строки, адресная часть А которых удовлетворяет неравенству $\alpha \leq A \leq \beta$, и циклически (команда 013) добавляет к ним содержимое δ ячейки 00246. Затем управление передается на начало СП "Коррекция настроек".

СП "Коррекция настроек" просматривает настройки 00170 ÷ 00237 и к каждой из настроек с адресной частью А, $\alpha \leq A \leq \beta$, циклически добавляет содержимое ячейки 00246. Затем осуществляется выход по 15-му ИР.

Ни одна СП группы "Сдвиг" не обращается к Диспетчеру.

(f) СП ГРУППЫ "РАБОТА С ОАС". В эту группу включены следующие СП: "Поиск строки ОАС", "Аннулирование строки ОАС", "Ан-

нулирование формулы в ОЗУ", "Отказ от формулы", "Выдача информации о формуле", "Аннулирование формулы всюду", "Прописка", "Изменение номера формулы", "Раздвоение формулы", "Приведение формулы к почти нормальному виду". Информация об этих СП содержится в таблице I2.

СП "ПОИСК СТРОКИ ОАС". Исходной информацией для данной СП служит номер N. Если в ОАС есть N-строка, т.е. строка, соответствующая формуле или оператору с номером N (точнее, строка, в первой позиции которой находится тетрада N), то на выходе СП сигнал ω принимает значение единица, в сумматор записывается ненулевой код, а в I6-й ИР посылается адрес A_{OAC} ячейки, содержащей N-строку ОАС. В противном случае ω принимает значение нуль, в сумматор записывается нуль, а в I6-й ИР посылается адрес A_{OAC} последней строки ОАС, соответствующей формуле или оператору с номером $N_1 \leq N$ (если таких строк в ОАС нет вообще, то A_{OAC} полагается равным $A^{0}_{OAC} - I$). Обращений к Диспетчеру нет.

СП "АННУЛИРОВАНИЕ СТРОКИ ОАС". Исходной информацией служит номер N. Если в ОАС есть N-строка, то она заменяется строкой

$$0000 \quad N \quad 0000 \quad 0000, \tag{I5}$$

сигнал ω на выходе принимает значение единица и в сумматор записывается код (I5). Иначе ОАС не изменяется, ω полагается равным нулю и в сумматор записывается нуль. В любом случае в I6-й ИР поступает тот же адрес A_{OAC}, что и в СП "Поиск строки ОАС". Диспетчер не включается.

СП "АННУЛИРОВАНИЕ ФОРМУЛЫ В ОЗУ". Исходной информацией служит номер N. Если в ОАС есть N-строка

$$\ell_N \; N \; \pi_N \; \gamma_N \; A_N \tag{5}$$

(в которой, возможно, $\ell_N = 0$), то она заменяется строкой (I5), в ячейки ОЗУ с адресами

$$A_N, A_N+1, \ldots, A_N+\ell_N-1$$

записываются нули, ω полагается равным единице и в сумматор засылается код (I5). Иначе ОАС, ОРП и СРП не изменяются, ω принимает значение нуль и в сумматор записывается нуль. В любом случае в I6-й ИР поступает тот же адрес A_{OAC}, что и в СП "Поиск строки ОАС". Диспетчер не используется.

Эту СП рекомендуется применять для уничтожения тех формул и

операторов, которые для дальнейшего счета не нужны и заведомо находятся в ОЗУ (например, имеют номер $N \geq N_{max}$ или $\gamma_N \neq$ 000).

СП "ОТКАЗ ОТ ФОРМУЛЫ". Исходная информация - номер N. Если в ОАС есть N - строка (5) и в ней $\gamma_N \neq$ 000, то её адрес $A_{ОАС}$ посылается в ИР-16, γ_N в N - строке ОАС уменьшается на единицу и в сумматор записывается новый вид N -строки ОАС. В противном случае СП прерывает работу Авто-Аналитика, поскольку отказ от формулы, о которой не была затребована информация, не допускается. На выходе $\omega = 0$ тогда и только тогда, когда длина формулы или оператора не меньше, чем 4000 ячеек. Диспетчер не включается.

Назначение данной СП описано в § 3 (e) и § 4.

СП "ВЫДАЧА ИНФОРМАЦИИ О ФОРМУЛЕ". Исходной информацией для данной СП служит номер N оператора или формулы. Если оператор (формула) с номером N находится в ОЗУ и ему (эй) соответствует строка (5) ОАС, то СП в строке (5) увеличивает на единицу γ_N и выдает в сумматор и в ячейку 00251 константу

6200 0000 000 A_N,

т.е. константу настройки на нулевую позицию первой из ячеек ОЗУ, занятых оператором (формулой) N.

Если оператор (формула) N находится в БРП, то он (она) предварительно переносится в ОЗУ. Для этого считывается N -строка БАС

$$e_N \; N \; \pi_N \; 0 \; A^N_{НБ}$$

и при помощи Диспетчера обеспечивается резерв ОРП объемом не менее e_N ячеек. Если $\pi_N = 0$, т.е. если номер N соответствует формуле, то при помощи СП "Конец формулы в ОРП 1" осуществляется подвод настройки 00236 на нулевую позицию и в конец ОРП 1 заносится формула N, поступающая в ОЗУ из БРП. Настройка 00236 в результате этого оказывается нацеленной на нулевую позицию первой ячейки резерва ОРП; в эту позицию вписывается тетрада Δ. Если же $\pi_N = 1$, т.е. если номер N соответствует оператору, то основные параметры $A^{\circ}_{ОАС}$, $A^{\circ}_{ТП}$ и $A^{\circ}_{СРП}$ уменьшаются на e_N, ОРП $\overline{II}$, ОАС и ТП сдвигаются при помощи СП "Сдвиг назад" на e_N ячеек и в освободившиеся ячейки СРП заносится оператор, поступающий в ОЗУ из БРП. Настройка 00236 в этом случае не тро-

Шифр	Название СП	Исходная информация	Обращение ИР	Обращение СП	Значения на выходе: Условие для $\omega=0$	Сумматор	ИР-14	ИР-15	ИР-16	Рабочие ячейки
ПАС	Поиск строки ОАС	N	14	00325	нет N-строки ОАС	?	$\mathcal{Z}$	Сохр.	А	00260,00261
АНСТР	Аннулирование строки ОАС	N	15	01210	-"-	0.0.0.0 или 0.N.0.0	01211		А	-"-
АНОЗУ	Аннулирование формулы в ОЗУ	N	15	00540	-"-	-"-	?		А	-"-
ОТКАЗ	Отказ от формулы	N	15	00340	$\ell_N \geq 4000$	-строка ОАС	00341		А	-"-
ВЫДАЧА	Выдача информации о формуле	N	15	00616	Всегда $\omega = 1$	6200 0000 000A_N	?	?	А	
АНФ	Аннулирование формулы всюду	N	15	00345	формулы N не было в ОЗУ	?	?	?	?	
ПРОПФ	Прописка формулы	N, H', H	15	00377	Всегда = 1	N-строка ОАС	?	?	А	
ПРОПО	Прописка оператора	N, H', H	15	00533	-"-	-"-	?	?	А	
ПРОПН	Прописка по настройке	N, H	15	01213	-"-	-"-	?	?	А	
РФ	Раздвоение формулы	N, N_1	15	02030	Всегда $\omega = 0$	нуль	?	02104	00000	02115+02117, 00176 и р.я. СП "Выдача"
ИНФ	Изменение номера формулы	N, N_1	15	02031	-"-	-"-	?	02045	00000	00176 и р.я. СП "Выдача"
НОРМ	Приведение формулы к почти нормальному виду	N	15	01225	ИР-12 содержит нуль	Содержимое ИР-12	A_{OAC}	?	00255	00176, 00252+00253 и р.я. СП "Выдача"

Таблица 12

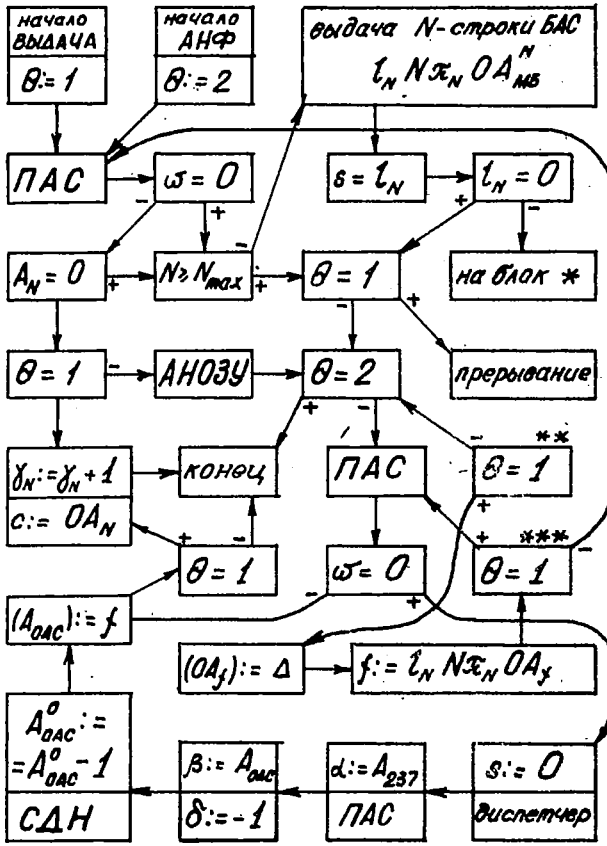

начало ВЫДАЧА
θ:= 1
начало АНФ
θ:= 2
выдача N-строки БАС
ПАС
ω = 0
s = ℓN
ℓN = 0
AN = 0
N > Nmax
θ = 1
на блок *
θ = 1
АНОЗУ
θ = 2
прерывание
γN:= γN + 1
конец
ПАС
θ = 1 **
c:= OAN
θ = 1
ω = 0
θ = 1 ***
(AOAC):= f
(OAf):= Δ
f:= ℓN NxN OAf
A°OAC:= = A°OAC - 1
СДН
β:= AOAC
δ:= -1
α:= A237
ПАС
s:= 0
диспетчер

Рис. 13

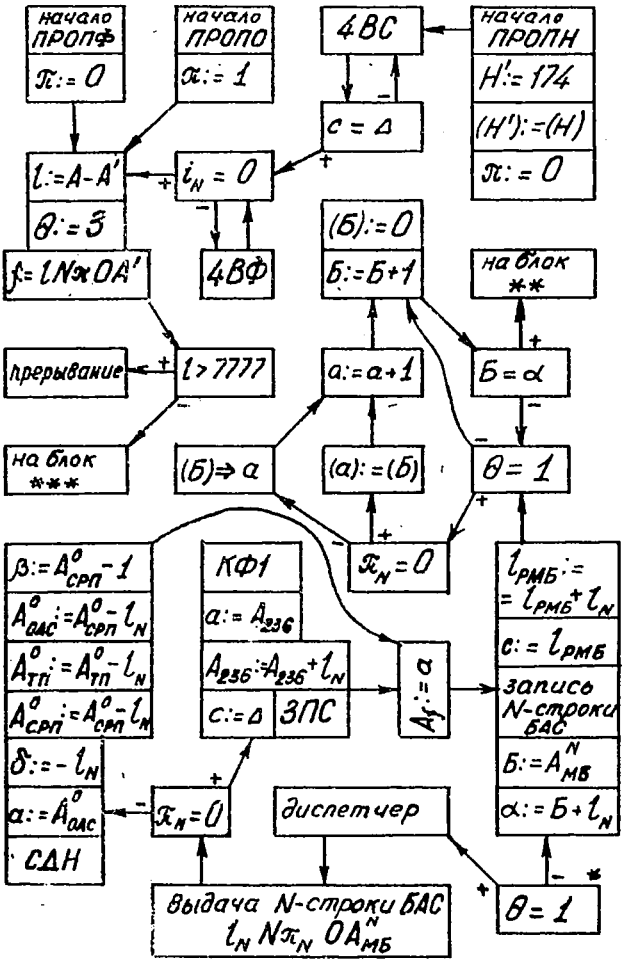

начало ПРОПФ
π:=0
начало ПРОПО
π:=1
4ВС
начало ПРОПН
H':=174
(H'):=(H)
π:=0
c=Δ
l:=A-A'
θ:=3
i_N=0
(Б):=0
Б:=Б+1
на блок **
прерывание
l>7777
a:=a+1
Б=α
на блок ***
(Б)⇒a
(a):=(Б)
θ=1
π_N=0
β:=$A^0_{СРП}$-1
$A^0_{ОАС}$:=$A^0_{СРП}$-l_N
$A^0_{ТП}$:=$A^0_{ТП}$-l_N
$A^0_{СРП}$:=$A^0_{СРП}$-l_N
δ:=-l_N
a:=$A^0_{ОАС}$
СДН
КФ1
a:=A_{236}
A_{236}:=A_{236}+l_N
c:=Δ
ЗПС
A_f:=a
$l_{РМБ}$:=$l_{РМБ}$+l_N
c:=$l_{РМБ}$
запись N-строки БАС
Б:=$A^N_{МБ}$
α:=Б+l_N
π_N=0
диспетчер
θ=1
Выдача N-строки БАС
l_N Nπ_N $OA^N_{МБ}$

Рис. 13а

гается. В обоих случаях нулевая позиция из ячеек ОЗУ, принявших в себя оператор или формулу N принудительно обнуляется (т.е. восстанавливается начальное Δ). Затем оператор или формула N прописывается в ОАС, его или ее копия в БРП аннулируется (т.е. ячейки БРП с адресами

$$A_{НБ}^{N},\ A_{НБ}^{N}+1,\ \ldots,\ A_{НБ}^{N}+\ell_N-1$$

обнуляются, а в БАС в качестве N -строки вписывается ненулевой код с нулевыми старшими разрядами) и управление передается на начало СП.

СП "Выдача информации о формуле" посылает в ИР-16 адрес $A_{ОАС}$ N-строки ОАС. Если формула или оператор N находится в ОЗУ, то Диспетчер не включается. В противном случае его включение неизбежно. Если формулы или оператора с номером N нет ни в ОЗУ ни в БРП, то СП прерывает работу Авто-Аналитика. Данная СП соединена с двумя следующими СП ("Аннулирование формулы всюду" и "Прописка"). Алгоритм этой объединенной СП приведен на рис.13. Обозначения на этой блок-схеме те же, что и на предыдущих блок-схемах. Отметим лишь, что через f обозначено содержимое ячейки 00175, через A_f - адресная часть кода f , (Б) $\Rightarrow a$ означает "содержимое ячейки Б переписывается в ячейку a с признаком команды.

СП "АННУЛИРОВАНИЕ ФОРМУЛЫ ВСЮДУ". Исходной информацией служит номер N . СП аннулирует формулу или оператор с номером N . При этом рассматриваются два случая. Если формула (оператор) N находится в ОЗУ, то работа данной СП эквивалентна СП "Аннулирование формулы в ОЗУ". Если же формулы или оператора N в ОЗУ нет, то N -строка БАС заменяется отличным от нуля кодом с нулевыми разрядами 48 + 37, а ячейки БРП, занятые этой формулой или этим оператором, обнуляются. В том случае, когда формулы или операторы N в ОЗУ или в БРП нет вообще, СП сразу, как только узнает об этом, осуществляет выход. К Диспетчеру обращений нет. Значительная часть этой СП совмещена с СП "Выдача информации о формуле" (см. рис.13).

СП "ПРОПИСКА". Данная СП имеет три модификации "Прописка формулы", "Прописка по настройке" и "Прописка оператора". Все эти модификации в программном отношении совмещены друг с другом и с двумя предыдущими СП. (См. рис.13).

СП "Прописка формулы" и "Прописка оператора" в качестве исходной информации используют номер N и адреса Н′ и Н двух констант

настроек, записанные соответственно в 14-м и 16-м ИР. Предполагается, что адресная часть настройки H′ содержит адрес $A_{H'}$ первой из ячеек ОЗУ, занятых формулой или оператором N, а адресная часть настройки H – адрес A_H ячейки ОЗУ, следующей за последней ячейкой ОЗУ, занятой данной формулой или данным оператором (таким образом, количество ячеек ОЗУ, занятых формулой или оператором N, равно $A_H - A_{H'}$). Рассматриваемые СП формируют N-строку ОАС вида

$$\ell_N \; N \; \pi_N \; \gamma_N \; A_N \qquad (5)$$

где

$$\ell_N = A_N - A_{H'}, \quad A_N = A_{H'}, \quad \gamma_N = 0$$

а $\pi_N = 0$, если прописывается формула (т.е. обращение произведено на вход "Прописка формулы"), и $\pi_N = 1$, если прописывается оператор (т.е. обращение произведено на вход "Прописка оператора").

Более подробно работа данных СП протекает следующим образом. Сначала формируется и запоминается в ячейке 00175 константа (5). (В случае

$$A_{H'} - A_H > 7777$$

происходит прерывание работы Авто-Аналитика). Затем при помощи СП "Аннулирование формулы всюду" аннулируется "старая" формула или "старый" оператор с номером N. (Таким образом, обращение к рассматриваемым СП возможно и в том случае, когда под номером N уже прописана некоторая информация и возникает необходимость в прописке под тем же номером новой информации; прежняя информация в этом случае автоматически исчезает из памяти). Затем при помощи СП "Поиск строки в ОАС" ищется последняя ячейка ОАС, в первой позиции которой записана тетрада $N_1 \leq N$, и ее адрес обозначается через A_{OAC}. Если $N_1 = N$ (в этом случае ячейка A_{OAC} содержит константу (15)), то в ячейку A_{OAC} переписывается содержимое ячейки 00175 и осуществляется выход из СП. В противном случае при помощи Диспетчера обеспечивается резерв ОРП объемом в 1 ячейку (S:=0), параметр A_{OAC} уменьшается на 1, ОРП $\overline{II}$ и начало ОАС (по ячейку A_{OAC}) сдвигается на 1 ячейку назад (т.е. α :=A_{237}, β:=A_{OAC}, δ :=–1, СП "Сдвиг назад"), в освободившуюся ячейку A_{OAC} вписывается содержимое

ячейки 00175 и осуществляется выход из СП.

На выходе в ИР-16 сохраняется адрес $A_{ОАС}$ новой N -строки ОАС, копия этой N -строки заносится в Сумматор. Константы настроек Н и Н′ не изменяются.

СП "Прописка по настройке" предназначена для прописки формул в ОАС и отличается от СП "Прописка формулы" тем, что в качестве исходной информации кроме номера N использует только адрес Н настройки на любую позицию первой из ячеек ОЗУ, занятых прописываемой формулой. Адрес Н хранится в ИР-16.
Данная СП полагает Н : = 00174, (Н):=(Н) и организует просмотр формулы по настройке Н при помощи СП "Чтение вперед С" до тех пор, пока не будет найдена тетрада Δ . Если эта тетрада окажется в нулевой позиции некоторой ячейки, то осуществляется переход на начало СП "Прописка формулы". В противном случае производится фиктивное чтение вперед по настройке Н до тех пор, пока параметр i_H не окажется равным нулю, и только после этого включается СП "Прописка формулы".

В процессе работы СП "Прописка по настройке" адрес $A_{ОАС}$ новой N -строки ОАС посылается в ИР-16. Копия этой N -строки поступает в сумматор; копия исходного состояния настройки Н записывается в ячейке 00174; в ячейке Н формируется настройка на нулевую позицию первой из ячеек ОЗУ, следующих за ячейками, занятыми прописываемой формулой. Возможно включение Диспетчера.

Все модификации СП "Прописка" посылают копию новой N -строки ОАС не только в Сумматор, но и в ячейку 00175.

СП "ИЗМЕНЕНИЕ НОМЕРА ФОРМУЛЫ". Исходной информацией для данной СП служат два номера N и N_1 , хранящиеся в виде констант

0000 N 0000 0000 (15)

0000 N_1 0000 0000 (16)

в ячейках 00243 и 02046 соответственно.

СП работает следующим образом. При помощи СП "Выдача информации о формуле" формула (оператор) с номером N вызывается в ОЗУ; на базе соответствующей N -строки ОАС формируется константа

$$\ell_N \; N_1 \; \Pi_N \; 0 \; A_N, \qquad (17)$$

в ячейку A_{OAC} в качестве новой N -строки ОАС записывается код (15); код (16) пересылается в ячейку 00243; при помощи СП "Аннулирование формулы всюду" из памяти удаляется формула (или оператор), имевшая в момент обращения к рассматриваемой СП номер N_1; код (17) записывается в ОАС в качестве новой его N_1 -строки (при этом особым образом используется СП "Прописка").

Иначе говоря, данная СП аннулирует старую формулу (оператор) с номером N_1 ; присваивает формуле (оператору), имевшей ранее номер N , новый номер N_1 ; затирает прежнюю N -строку ОАС или БАС; изменяет содержимое ячейки 00243, полагая его равным коду (16).

Возможно включение Диспетчера. Если в момент обращения к СП в памяти нет ни формулы, ни оператора с номером N , то СП вызовет прерывание работы Авто-Аналитика.

СП "РАЗДВОЕНИЕ ФОРМУЛЫ". Исходная информация - константы (15) и (16) в ячейках 00243 и 02046. СП образует в ОРП 1 копию формулы N и прописывает ее под номером N_1. Алгоритм работы СП приведен на рис.14. Содержимое ячеек 00243 и 02046 сохраняется. Возможно включение Диспетчера. Если под номером N был прописан оператор ($\mathcal{T}_N = 1$) или под номером N в ОАС и БАС ничто не было прописано, то СП вызовет прерывание работы Авто-Аналитика.

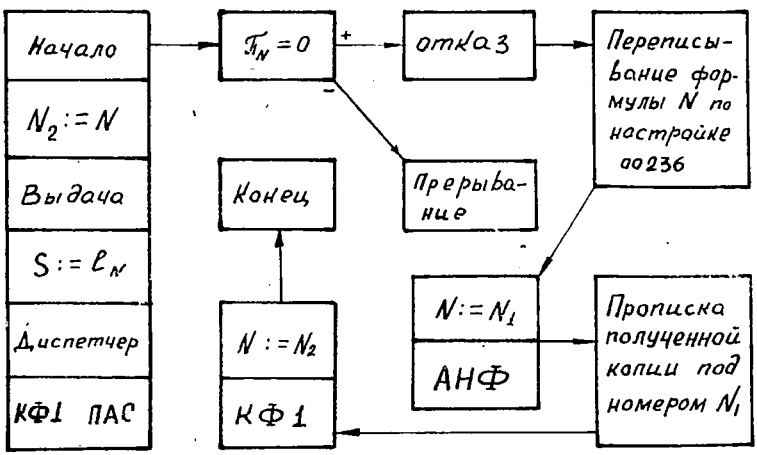


Рис. 14. Алгоритм СП РФ

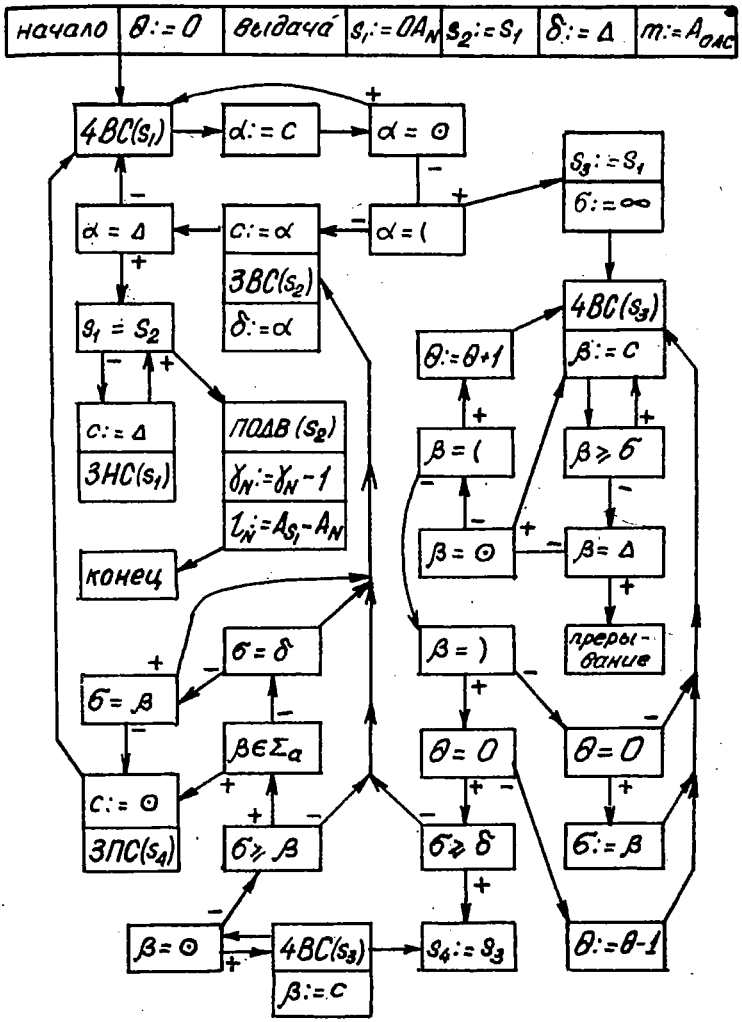


Рис. 15. Алгоритм СП НОРМ

СП "ПРИВЕДЕНИЕ ФОРМУЛЫ К ПОЧТИ НОРМАЛЬНОМУ ВИДУ". Исходной информацией служит номер формулы N . СП переписывает формулу с номером N , пропуская коды скобок, входящие в устранимые пары скобок, и коды пустых символов, если таковые по каким-либо причинам появились среди тетрад кода формулы N . Результат - почти нормальный вид исходной формулы N - прописывается в ОАС под прежним номером N .

Подробный алгоритм работы СП изображен на рис.15. Символами θ , δ , m , λ, σ, β обозначены внутренние параметры алгоритма. Символы ϑ_1 , ϑ_2, ϑ_3, ϑ_4 означают константы настроек, расположенные вне массива корректируемых настроек. Обозначение ЧВС (ϑ_1) означает обращение к СП "Чтение вперед С" при H = ϑ_1 . Обозначения

ЗВС (ϑ_2), ЧВС (ϑ_3), ЗНС (ϑ_1), ЗПС (ϑ_4), ПОДВ(ϑ_2)

следует понимать аналогичным образом. Прочая символика рис.15 та же, что и на предыдущих рисунках.

При использовании данной СП нужно иметь ввиду следующее. Поиск исходной формулы N осуществляется при помощи СП "Выдача информации о формуле". Если формула N находится в БРП, то она переносится в ОРП 1 и в этом случае возможно включение Диспетчера. Если же формула N находится в ОРП, то обращения к Диспетчеру нет. Получаемый в качестве результата почти нормальный вид исходной формулы N записывается в ОЗУ на месте этой исходной формулы. Освободившиеся позиции обнуляются и собираются в единый массив после конечного граничного символа (предполагается, что формула N расписана "вперед").

СП вызовет прерывание работы Авто-Аналитика, если в памяти нет информации, прописанной под номером N . То же самое произойдет в случае, когда после некоторой раскрывающей скобки в "формуле" N символ Δ встретится раньше, чем соответствующая закрывающая скобка. Прочие случаи неправильной расстановки скобок СП не заметит. Если под номером N прописан массив формул, то СП приведет к нормальному виду и пропишет под номером N лишь одну первую формулу этого массива. Прочие формулы данного массива из памяти не исчезнут, но окажутся нигде не прописанными и разыскать их будет трудно. Наконец, если под номером N прописан оператор, то СП преобразует этот оператор по своему алгоритму и,

тем самым, безнадежно испортит его. С вероятностью около 1/2 в подобном случае произойдет прерывание работы Авто-Аналитика.

§ 6. ИНТЕРПРЕТИРУЮЩАЯ СИСТЕМА (РИДЕР).

Основу интеллекта Авто–Аналитика составляет его библиотека стандартных операторов (БСО). Каждый оператор представляет собой запись алгоритма для решения одной или нескольких однотипных задач. В БСО имеются, например, операторы Коммутирование, Некоммутативная обобщенная подстановка, Коммутативная обобщенная подстановка, Арифметик, Приведение подобных членов, Программы линейной алгебры, Дифференцирование, Интегрирование, Метод внешних форм Картана и многие иные. Конкретное описание каждого оператора из БСО см в гл. IУ. Пользователь всегда может составить новые операторы и включить их в БСО.

Каждому оператору из БСО присвоен номер N, удовлетворяющий условию

$$0040 \leq N \leq 0777$$

Если в программе математика прямо или косвенно (т.е. через посредство иных операторов) используется оператор N, то номер N нельзя сопоставлять никакому другому массиву информации и, в частности, никакой формуле. Кроме номера почти каждому оператору ставится в соответствие неотрицательное целое число K_N, указывающее, сколько тетрад занимает исходная информация для данного оператора (т.е. сколько исполнительных аргументов должно содержать обращение к данному оператору).

Обращение к оператору с номером N в общем случае имеет вид

$$\pi N Q_1 Q_2 \ldots Q_m, \qquad (I8)$$

где $Q_1, Q_2, \ldots, Q_m$ – тетрады, именуемые действительными аргументами обращения (I8), а π – признак формирования исполнительных аргументов. Признак π занимает три двоичных разряда. Поскольку старшая восьмеричная цифра в N всегда равна нулю, то она в обращении (I8) заменяется признаком π. Таким образом, выражение πN в (I8) предствляет собой одну тетраду. (Например, если первая тетрада в обращении (I8) равна I725, то (I8) есть обращение к оператору 0725 с признаком $\pi = I$).

Признак π определяет способ формирования исполнительных аргументов

$$Q_1', Q_2', \ldots, Q_{K_N}'$$

на основе действительных аргументов $Q_1, Q_2, \ldots, Q_m$ обра-

щения (18). Если $\pi = 0$, то исполнительные аргументы отождествляются с соответствующими действительными аргументами; ясно, что в этом случае m должно совпадать с K_N.

При $\pi = 1$ каждый исполнительный аргумент Q_i' формируется из трех действительных аргументов Q_{3i-2}, Q_{3i-1} и Q_{3i}. Поэтому в случае $\pi = 1$ количество m действительных аргументов в обращении (18) должно совпадать с $3K_N$. Обозначим через q_1, q_2, q_3 и q_4 первую, вторую, третью и четвертую восьмеричные цифры тетрады Q_{3i-2}. Исполнительный аргумент Q_i' при $\pi = 1$ вычисляется по формуле

$$Q_i' = \{ Q_{3i} + И_{q_4} + \{ (И_{q_3}) \} \bmod 2^{12} + И_{q_2} + \\ + \{ (\{ Q_{3i-1} + И_{q_1} \} \bmod 2^{15}) \} \bmod 2^{12} \} \bmod 2^{12}, \qquad (20)$$

где $И_q$ – содержимое ИР с номером q, (A) для $A = И_{q_3}$ или

$$A = \{ Q_{3i-1} + И_{q_1} \} \bmod 2^{15} \qquad (19)$$

означает содержимое ячейки с адресом A и через

$$\{a\} \bmod 2^{12}$$

обозначен код, представленный 12-тью младшими разрядами двоичного кода a (выражение (19) понимается аналогично).

Значения $\pi = 2,3,\ldots,7$ пока не используются и составляют резерв для возможных дополнений в будущем.

Кроме стандартных операторов, включенных в БСО, можно использовать так называемые внутренние операторы с номерами 0000÷0004 (Вообще для внутренних операторов отведены номера 0000÷0037) однако в настоящее время операторов с номерами 0005÷0037 еще нет). Внутренние операторы реализованы в виде стандартных программ и включены в БСП. Обращение к внутреннему оператору также имеет вид (18), однако признак здесь не играет никакой роли. Исполнительные аргументы внутренних операторов всегда отождествляются с действительными аргументами.

Программа каждого оператора из БСО представляет собой последовательность обращений к иным операторам, т.е. имеет вид

$$\Delta\ \pi_1 N_1 Q_{11} Q_{12} \ldots Q_{1m_1}\ \pi_2 N_2 Q_{21} Q_{22} \ldots Q_{2m_2} \ldots \pi_s N_s Q_{s1} \ldots Q_{sm_s}\ \Delta \qquad (21)$$

В СРП или БРП оператор (21) занимает целое число подряд расположенных ячеек. В каждую ячейку пишется в естественном порядке по четыре тетрады оператора (21). Последняя ячейка содержащая в одной из своих позиций конечное Δ, считается занятой оператором (21) даже в том случае, когда эта позиция нулевая. Таким об-

разом, при записи операторов в СРП конечное Δ одного оператора нельзя совмещать с начальным Δ другого оператора. В записи оператора (2I) могут присутствовать полностью обнуленные ячейки. Каждый оператор из БСО прописывается в ОАС или БАС под своим номером N, причем признак $\mathcal{T}_N$ в соответствующей N строке ОАС или БАС полагается равным I.

Внутренние операторы нигде не прописаны. Поэтому номера 0001÷0037 можно использовать для обозначения формул (но не новых операторов, включаемых математиком в БСО).

Работа операторов вида (2I) осуществляется в режиме интерпретации при помощи специальной интерпретирующей программы "Ридер". Ридер занимает фиксированное место в БСП. Кроме стандартных операторов Ридер интерпретирует (прокручивает) так называемые операторные участки программы математика. Каждый такой участок представляет собой последовательность обращений к стандартным и внутренним операторам, лишенную однако начального и конечного симовлов Δ.

Обращение к Ридеру осуществляется командой

I5 3I 0I523 (22)

Ридер предполагает, что операторный участок начинается с нулевой позиции ячейки $\mathcal{Ж}$, где $\mathcal{Ж}$ берется из ИР-I5 (т.е. фактически $\mathcal{Ж} = p + I$, если команда (22) расположена в ячейке p). Ридер организует просмотр операторного участка

$$\mathcal{T}_1 N_1 Q_{11} Q_{12} \dots Q_{1m_1} \mathcal{T}_2 N_2 Q_{21} Q_{22} \dots Q_{2m_2} \dots \dots \quad (23)$$

и последовательно исполняет (т.е. интерпретирует) каждый из операторов, обращения к которым встречаются в последовательности (23). Естественный порядок выполнения операторов нарушается лишь в том случае, когда встречается оператор перехода.

В общих чертах работа Ридера состоит в следующем. Основным параметром Ридера служит настройка 00I7I. Будем именовать ее "читающей головкой". В исходный момент времени читающая головка настраивается на чтение тетрады из нулевой позиции ячейки $\mathcal{Ж}$. Иначе говоря, она настраивается на чтение тетрады $\mathcal{T}_1 N_1$ операторного участка (23). Назовем шагом Ридера ту часть его работы, в результате которой читающая головка перейдет от тетрады $\mathcal{T}N$ одного обращения вида (I8) к тетраде $\mathcal{T}N$ другого обращения. Допустим, что после некоторого числа шагов Ридера его читающая головка добралась до некоторого обращения вида (I8) в операторном участке Я(23) или, возможно, в некотором операторе

(2I) из БСО. Работа Ридера на следующем шаге зависит от номера N в обращении (I8) и определяется следующими правилами:

(а) Если $N \geq$ 0040 (т.е. (I8) есть обращение к оператору из БСО), то Ридер запоминает свое текущее состояние (т.е. номер прокручиваемого оператора (2I) и то место в нем, где встретилось обращение (I8)) в виде отдельной строки ТП, Затем Ридер при помощи СП "Выдача информации о формуле" вызывает в СРП оператор N и настраивает свою читающую головку на чтение его первой тетрады.

(б) Если $N \leq$ 0037 (т.е. (I8) есть обращение к внутреннему оператору), то Ридер производит обращение к стандартной программе в БСП, реализующей данный внутренний оператор N.

Опишем внутренние операторы Ридера.

(I) ОПЕРАТОР ВЫХОДА. Номер N = 0000. Аргументов не имеет. Употребляется только в операторах, включенных в БСО. Применение оператора 0000 в операторном участке программы математика приводит к непредсказуемым последствиям (прерывание, перемешивание памяти, интерпретация наобум выбранного оператора и т.п.). Если обращение к оператору 0000 встретилось в операторе (2I), то Ридер прекращает выполнение оператора (2I) и по соответствующей строке ТП осуществляет возврат к продолжению интерпретации того оператора или операторного участка, из которого было обращение к оператору (2I).

Для выхода из оператора (2I) в пределах его кодовых участков (см.ниже) можно поставить команду

00 30 0I624 (24)

прямой передачи управления на программу, реализующую оператор 0000. Если выход из оператора (2I) оформлен через оператор 0000, расположенный где-то внутри оператора (2I) или при помощи команды (24), то конечный граничный символ Λ в записи оператора (2I) может быть опущен.

(2). ПУСТОЙ ОПЕРАТОР. Номер N = 0001. Аргументов не имеет. Встречая оператор 0001, Ридер пропускает его и настраивает свою читающую головку на чтение следующей тетрады в операторе (2I) или в операторном участке (23) программы математика. Оператор 0001 может употребляться для разметки программы. Он полезен также в некоторых иных случаях.

(3) ОПЕРАТОР ВЫДАЧИ АРГУМЕНТОВ. Номер N = 0002. Употребляется только в операторах включенных в БСО. В рабочей

программе математика применение оператора 0002 не имеет смысла. Обращение к оператору 0002 имеет вид

$$0002 \qquad 000I \qquad K_M , \qquad (25)$$

где M – номер оператора (2I), в котором встретилось данное обращение, и K_M – количество исполнительных аргументов в обращении

$$\pi M Q_1 Q_2 \ldots Q_m \qquad (26)$$

к оператору M. Допустим, что последняя тетрада K_M обращения (25) занимает некоторую позицию ячейки A. Содержимое следующих позиций ячейки A не играет роли. Ячейки A + I, A + 2, ..., A + K_M, следующие в записи оператора (2I) за обращением (25), должны быть свободными. Оператор 0002 по соответствующей строке ТП отыскивает в памяти обращение (26) к оператору (2I), извлекает из этого обращения дополнительные аргументы.

$$Q_1' \; Q_2' \ldots Q_{K_M}'$$

(по описанным выше правилам) и записывает их в третьи позиции ячеек A + I, A + 2, ..., A + K_M соответственно. Попутно оператор 0002 корректирует соответствующую строку ТП для того, чтобы обеспечить правильное срабатывание оператора выхода из оператора (2I). В заключении своей работы оператор 0002 настраивает читающую головку Ридера на чтение тетрады из нулевой позиции ячейки A + K_M + I.

Если оператор (2I) не требует никаких аргументов, т.е. если K_M = 0, то оператор (2I) не должен содержать ни одного обращения к оператору 0002. Если $K_M > 0$, то оператор (2I) должен иметь равно одно обращение к оператору 0002 и это обращение должно срабатывать равно один раз. Естественно располагать это обращение в самом начале оператора (2I). Поэтому обычно начало оператора (2I), требующего фиксированное количество $K_M \neq 0$ исполнительных аргументов, имеет вид

A + 0	0000	0002	000I	K_M	Обращение к оператору 0002;
A + I	0000	0000	0000	0000	ячейка первого аргумента;
A + 2	0000	0000	0000	0000	ячейка второго аргумента;
A + K	0000	0000	0000	0000	ячейка K – того аргумента
A + K + I	$\pi_2 N_2$	Q_1	Q_2	...	начало основной части оператора

. . . . . .

. . . . . .

(4) ОПЕРАТОР "КОДОВЫЙ УЧАСТОК" Номер N = 0003. Обращение к оператору 0003 может произойти как в стандартном операторе, так и в операторном участке программы математика. Ридер считает, что ячейки

следующие за той ячейкой А, в одной из позиций которой встретилась тетрада 0003, заняты некоторой программой в машинных командах. Поэтому выполнение оператора 0003 сводится к передаче управления в ячейку А + I. Таким образом, оператор 0003 эквивалентен команде

00 30 A + I

Содержимое позиций ячейки А, следующих за тетрадой 0003, не играет роли. Исполнив оператор 0003, Ридер временно прекращает свою деятельность до тех пор, пока в центральный процессор БЭСМ-6 не поступит команда (22) обращения к Ридеру.

Все операторы из БСО сдвигаются в моменты работы Диспетчера, и вместе с операторами сдвигаются их кодовые участки. Но в кодовых участках часто встречаются так называемые внутренние адреса, т.е. адреса ячеек, включенных в тело оператора (2I) и перемещающихся синхронно с ним. Таковы адресные части команд переходов в пределах одного кодового участка или из одного кодового участка в другой кодовый участок того же оператора, а также адреса констант и рабочих ячеек, включенных в тело оператора. Все внутренние адреса зависят от того, где в ОЗУ записан рассматриваемый оператор, и должны модифицироваться при его сдвиге. Проблема модификации внутренних адресов в Ридере решается при помощи ИР-I. А именно, предполагается, что никакие органы Авто-Аналитика кроме Ридера и Диспетчера (и в том числе программы и операторы, составленные математиком) не могут изменить содержимое ИР-I. Диспетчер рассматривает содержимое ИР-I как адресную часть некоторой настройки и корректирует его соответствующим образом. Работа Ридера построена так, что в момент перехода на исполнение кодового участка в операторе (2I) ИР-I содержит адрес первой из ячеек, занятых в ОЗУ оператором (2I) Поэтому все внутренние адреса в кодовых участках оператора (2I) заменяются их относительными адресами и модифицируются при помощи ИР-I. (Здесь под относительным адресом ячейки, входящей в тело оператора (2I) понимается разность А – В, где А – ее фактический адрес в ОЗУ, а В – адрес первой из ячеек ОЗУ, занятых оператором (2I), т.е. адрес, записанный в ИР-I. Ясно, что относительные адреса не зависят от места записи оператора в ОЗУ).

(5) ОПЕРАТОР "КОНЕЦ ЗАДАЧИ". Номер N = 0004. Аргументов не имеет. По результату работы эквивалентен макрокоманде

00 074 0000

(см. описание БЭСМ-6 в работе [21]).

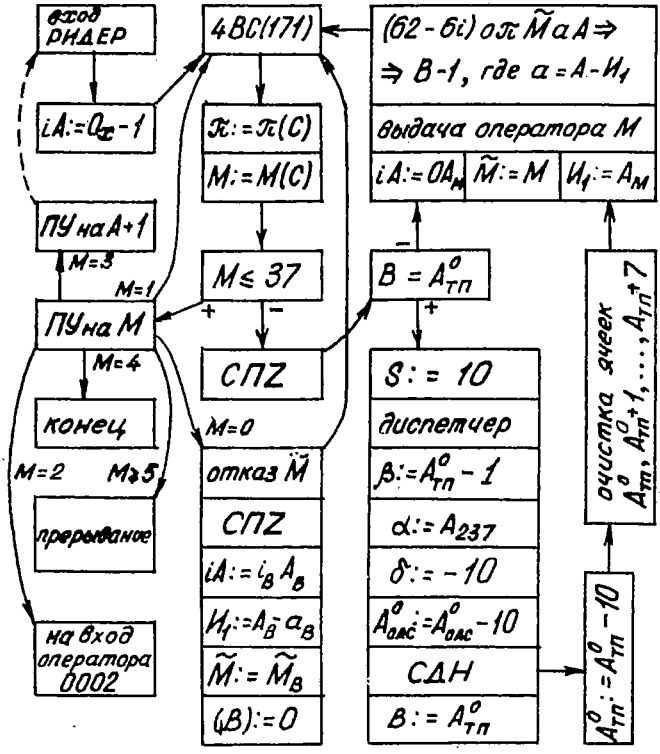

вход РИДЕР
4ВС(171)
(62-6i) оπ M̃ a A⇒
⇒ B-1, где a=A-И1
выдача оператора M
iA:=0Aм
M̃:=M
И1:=Aм
iA:=Qπ-1
π:=π(C)
M:=M(C)
ПУ на A+1
M=3
M=1
M≤37
B=A°тп
очистка ячеек A°тп, A°тп+1, ..., A°тп+7
ПУ на M
M=4
СПZ
S:=10
диспетчер
конец
M=0
β:=A°тп-1
M=2
M≥5
отказ M̃
α:=A237
прерывание
СПZ
δ:=-10
iA:=iв Aв
A°вас:=A°вас-10
на вход оператора 0002
И1:=Aв-aв
M̃:=M̃в
СДН
A°тп:=A°тп-10
(B):=0
B:=A°тп

Рис.16

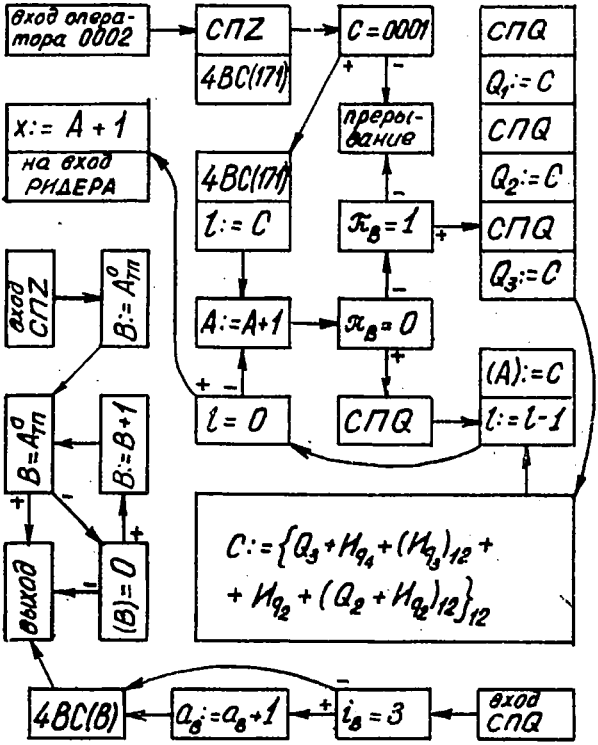

вход оператора 0802
СПZ
4ВС(171)
C=0001
СПQ
Q1:=C
прерывание
x:=A+1
на вход РИДЕРА
4ВС(171)
l:=C
xв=1
СПQ
Q2:=C
СПQ
Q3:=C
вход СПZ
B:=A0пп
A:=A+1
xв=0
B=A0пп
B:=B+1
l=0
СПQ
(A):=C
l:=l-1
C:={Q3+Иq4+(Иq3)12+
+Иq2+(Q2+Иq2)12}12
выход
(B)=0
4ВС(В)
aв:=aв+1
iв=3
вход СПQ

Рис. 16а

Внутренних операторов с номерами 0005 ÷ 0037 пока еще нет. Если в каком-либо операторе или операторном участке встречается обращение к оператору с таким номером, то Ридер вызывает прерывание работы всей системы.

Опишем алгоритм Ридера более подробно.

(А) ВХОД. В ячейке 00171 формируется настройка на третью позицию ячейки $\varkappa - 1$, где $\varkappa$ берется из ИР-15. (Подготовка к чтению вперед начиная с нулевой позиции ячейки $\varkappa$).

(Б) ЧТЕНИЕ. По настройке 00171 при помощи СП "Чтение вперед" считывается одна тетрада; из трех старших двоичных разрядов этой тетрады выделяется признак π, а из девяти младших разрядов - номер оператора М.

(В) ПЕРЕКЛЮЧАТЕЛЬ. Если $M \geqslant 0040$, то переход на блок (Г); если М равно 0000, 0001, 0002, 0003 или 0004, то переход соответственно на блоки (Ж), (Б), (З), (И) и (К). В случае $0005 \leqslant M \leqslant 0037$ прерывание.

(Г) КОНСЕРВАЦИЯ ОПЕРАТОРА $\widetilde{M}$. Поскольку $M \geqslant 0040$, то встретилось обращение к оператору из БСО. Для прокручивания оператора М прежде всего требуется законсервировать текущее состояние Ридера. В связи с этим ищется последняя свободная, т.е. обнуленная ячейка ТП (Таблица переходов ТП заполняется с конца). Если свободных ячеек в ТП нет, то переход на блок (Е). Иначе в последнюю свободную ячейку ТП заносится код (= строка ТП)

$$(62 - 6i)\quad 0\quad \pi\quad \widetilde{M}\quad \alpha\quad A,$$

где i A - адрес позиции, на которую нацелена настройка 00171 (блок (Б) только что извлек из этой позиции тетраду πM), $\widetilde{M}$ - девятиразрядный номер оператора, который прокручивается Ридером в данный момент (т.е. оператора, на одну из позиций которого нацелена настройка 00171; если эта позиция входит в операторный участок рабочей программы математика, то М условно считается равным нулю), α есть двенадцатиразрядный относительный адрес ячейки А в операторе М (т.е. $\alpha = A - И_1$, так как содержимое ИР-1 согласно описанию блока (Д) совпадает с адресом первой из ячеек ОЗУ, занятых оператором $\widetilde{M}$; если математик самовольно не изменяет ИР-1, то в моменты прокрутки операторных участков его рабочей программы значе-

ние $И_1$ полагается равным нулю).

(Д) ПЕРЕХОД НА ОПЕРАТОР М. При помощи СП "Выдача информации о формуле" оператор с номером М вызывается в СРП. Константа настройки

6200 0000 000 A_M

на нулевую позицию ячейки A_M – первой из ячеек ОЗУ, занятых оператором М, засылается в ячейку 00171. Адрес A_M заносится также в ИР-1. Номер $\widetilde{M}$ полагается равным М. Переход на блок (Б).

(Е) РАСШИРЕНИЕ ТП. Переход на данный блок возможен лишь из блока (Г) и лишь в том случае, когда в ТП нет свободных ячеек. Данный блок расширяет ТП на 10 ячеек за счет резерва ОРП. А именно, производится обращение к Диспетчеру при $s = 10$, сдвигаются назад на 10 ячеек ОРП II и ОАС, уменьшаются на 10 параметры A_{237}, $A^{\circ}_{ОАС}$ и $A^{\circ}_{ТП}$ и, наконец, обнуляются новые ячейки ТП, имеющие адреса

$$A^{\circ}_{ТП},\ A^{\circ}_{ТП}+1,\ \ldots,\ A^{\circ}_{ТП}+7.$$

Переход на блок (Г).

(Ж) ВОЗВРАТ ПО ТП (М = 0000). Оператор $\widetilde{M}$ закончил свою работу и Ридер должен вернуться к продолжению интерпретации того оператора, из которого произошло обращение к оператору М. Для этого при помощи СП "Отказ от формулы" уменьшается на единицу счетчик γ в $\widetilde{M}$ -строке ОАС, отыскивается первая ненулевая строка ТП

$$(62-6i_в)\ 0\ \pi_в\ \widetilde{M}_в\ a_в\ A_в \qquad (27)$$

(т.е. та строка, которая была сформирована позднее всех прочих ненулевых строк ТП), на ее основе константа настройки

(62 – 6 $i_в$) 00 0000 000 A_B

посылается в ячейку 00171, разность $A_B - a_в$ посылается в ИР-1, номер $\widetilde{M}$ полагается равным $\widetilde{M}_B$, ячейка ТП с адресом В, содержащая строку (27), обнуляется. Переход на блок (Б).

(З). ВЫДАЧА АРГУМЕНТОВ (М = 0002). По настройке 00171 считывается одна тетрада С. Если она отлична от 0001, то работа Авто-Аналитика прерывается. (Эта тетрада есть первый аргумент обращения к оператору 0002; он определяет тип формирования исполнительных аргументов; пока реализован лишь тип 0001). Иначе по

настройке 00171 считывается еще одна тетрада. Она обозначается через ℓ (Это второй аргумент оператора 0002, указывающий количество исполнительных аргументов). Затем ищется первая ненулевая ячейка В в ТП. Она содержит некоторую строку вида (27). Согласно описанию блока (Г) позиция, адрес $i_в A_в$ которой указан в строке (27), содержит первую тетраду обращения к интерпретируемому в данный момент оператору $\widetilde{M}$, а признак π в этом обращении совпадает с параметром $\pi_в$ строки (27). Если $\pi_в \geqslant 2$, то работа Авто-Аналитика прерывается (обращение к стандартному оператору с непредусмотренным способом формирования исполнительных аргументов). В противном случае строка (27) воспринимается в качестве константы настройки, т.е. ее адрес В посылается в ИР-16, и по этой константе настройки при помощи СП "Чтение вперед" считывается ℓ или 3ℓ (в зависимости от значения $\pi_в$) тетрад. Из этих тетрад по описанному в начале текущего параграфа правилу формируется ℓ исполнительных аргументов, которые по одному заносятся в третьи позиции ячеек A + 1, A + 2, ... , A + ℓ, где A есть адресная часть настройки 00171. В процессе чтения по настройке В параметр $a_в$ строки (27) увеличивается на 1 каждый раз, когда настройка В переходит на чтение позиций из новой ячейки. В заключение работы блока (З) адрес A + ℓ заносится в ИР-15 в качестве α и осуществляется переход на блок (А).

(И). КОДОВЫЙ УЧАСТОК (М = 0003). Передача управления в ячейку A + 1, где A – адресная часть настройки 00171.

(К) КОНЕЦ ЗАДАЧИ (М = 0004). Выполняется макрокоманда

00 074 0000

по которой БЭСМ-6 прекращает счет по программе математика.

Подробная блок-схема алгоритма работы Ридера приведена на рис. 16. Все обозначения кроме СПZ и СПQ объяснены выше. Через СПZ и СПQ обозначены обращения к двум стандартным программам специального вида. Алгоритмы работы этих СП также указаны на рис. 16.

Отметим в заключение, что Ридер не меняет содержимое основных рабочих ячеек Авто-Аналитика с адресами 00240 ÷ 00243, затирает рабочие ячейки Диспетчера и СП "ВЫДАЧА". Содержимое ИР-15 и ИР-16 также не сохраняется.

ГЛАВА III

ВХОДНОЙ ЯЗЫК

Разработка входного языка вызвана необходимостью облегчить общение пользователя с системой программирования Авто-Аналитик. Язык позволяет упростить процесс подготовки исходной информации, а также программирование новых операторов системы.

§ I. ПРЕДВАРИТЕЛЬНЫЕ ПОНЯТИЯ

Вся информация, предназначенная для ввода в ЭЦВМ, записывается при помощи символов, представленных в таблице 6, в виде последовательности строк. В построении строк участвуют слова, числа и константы.

ОПРЕДЕЛЕНИЕ I. Словом называется конечная последовательность букв и цифр, начинающаяся с буквы. Слово идентифицируется по первым шести символам, остальные безразличны. Примерами слов могут служить следующие последовательности символов

A273
СУМ19Е
AI
ЕСЛИ
П БУКВА.

Числа делятся на вещественные и условные. Признаком вещественного числа служит наличие в его изображении точки или подстрочной десятки. В условных (целых) числах эти символы, а также знак минус отсутствуют.

ОПРЕДЕЛЕНИЕ 2. Вещественным числом называется последовательность $\alpha_1 \alpha_2 \dots \alpha_n$ цифр и трех знаков — $\cdot$ $_{10}$ $^{\cdot}$, удовлетворяющая условиям:

(а). В последовательности $\alpha_1 \alpha_2 \dots \alpha_n$ имеется по крайней мере одна цифра и по крайней мере один из символов $\cdot$ и $_{10}$.

(б). Каждый из символов $\cdot$ и $_{10}$ участвует в последовательности $\alpha_1 \alpha_2 \dots \alpha_n$ не более одного раза; если оба эти знака содержатся в данной последовательности, то знак : располо-

жен левее знака $_{10}$.

(в). Если знак $_{10}$ содержится в последовательности $\alpha_1 \alpha_2 \ldots \alpha_n$, то после него в этой последовательности есть хотя бы одна цифра.

(г). Со знаком минус в последовательности $\alpha_1 \alpha_2 \ldots \alpha_n$ может совпадать лишь ее первый член и тот ее член, который следует за знаком $_{10}$.

Примерами записи вещественных чисел могут служить последовательности

$$5.4_{10}1 \qquad 9.91 \qquad _{10}5 \qquad 0.$$

При трансляции текста с входного на внутренний язык системы вещественные числа переводятся в восьмеричную систему и обычно записываются в отдельные ячейки в виде нормализованного машинного кода. Отсюда вытекают естественные ограничения на абсолютную величину вводимых чисел и количество значащих цифр в них. Следует иметь в виду, что числа, входящие в формулы, кодируются особым образом (см. § 3).

ОПРЕДЕЛЕНИЕ 3. Условным (целым) числом называется непустая конечная последовательность цифр, первая из которых отлична от нуля. Последовательность, состоящая из одной цифры 0, также объявляется условным числом.

При трансляции условные числа переводятся в восьмеричную систему. Код каждого условного числа записывается в младшие разряды отдельных ячеек (старшие разряды обнуляются). Например условное число I запишется единицей в младшем разряде соответствующей ячейки.

При практическом программировании часто оказывается необходимым ввод определенных констант, не требующих перевода и занимающих целую ячейку.

ОПРЕДЕЛЕНИЕ 4. Константой называется последовательность $|\alpha_1 \alpha_2 \ldots \alpha_n|$, где $n \geq 1$ и α_i для каждого i есть либо восьмеричная цифра, либо элементарный символ, принадлежащий одновременно внутреннему и входному языку Авто-Аналитика.

Транслятор освобождает константу от прямых скобок $|\cdot|$, заменяет элементарные символы соответствующими кодами (таблица 6), дополняет полученный код слева нулевыми разрядами (либо выбрасывает из него слева лишние цифры). Построенная таким обра-

зом константа записывается в отдельную ячейку. Примерами констант могут служить последовательности

I0I I7777I IABI IA7I I525. + IБ22,, I

§ 2. ВЫРАЖЕНИЯ

Понятие выражения аналогично понятию формулы и служит для записи алгоритма вычисления тех или иных параметров задачи чисто арифметическим путем. Роль букв в выражениях играют слова, числа и константы, а в роли связей выступают символы

$$\vee \wedge + - \times / \uparrow *$$

причем каждая из этих связей сильнее предыдущей и слабее последующей. Выражение определяется следующим рекурсивным образом.

ОПРЕДЕЛЕНИЕ 5. (а). Слово, число или константа есть выражение с главной связью ∞.

(б). Если некоторая последовательность символов ω есть выражение, то последовательности

$$(\omega),\ \underline{У}(\omega),\ \underline{Я}(\omega),\ \underline{Ф}(\omega),\ \underline{В}(\omega)$$

являются выражениями с главной связью ∞.

(в). Если σ есть один из четырех символов $\{- / \uparrow *\}$, τ и ω – выражения, главные связи которых сильнее связи σ, то последовательность $\tau\sigma\omega$ есть выражение с главной связью σ.

(г). Если σ есть один из четырех символов $\{\wedge \vee + \times\}$, а τ и ω – выражения, главные связи которых не слабее связи σ, то последовательность $\tau\sigma\omega$ есть выражение с главной связью σ.

Примерами выражений могут служить последовательности

1) $\underline{В}(5.+6.+A)$
2) $\underline{Ф}(\underline{У}+|4000|)*A+Б$
3) $\underline{У}(\underline{Я}2100+5.+Б*C)$
4) $\underline{В}(SUM1+2*EXP*X\uparrow 1)$
5) $A-(B-C)$
6) $(A-B)-C$

7) $A+C-B$

8) $\underline{B}\,((5/6)/A-C)-\underline{B}\,((A\uparrow X)\uparrow Y)+\underline{B}\,(SIN*(COS*X))$

Например, последовательности вида

1) $A-Б-C$

2) $5/6/A$

3) $A\uparrow X\uparrow Y$

4) $SIN*COS*X$

выражениями не являются.

Аналогично понятиям пары скобок, подформулы и аргументов связи формулы можно ввести понятия пары скобок, подвыражения и аргументов связи в выражении. Так, например, в выражении

$$A1\times B+EXP*X$$

имеются подвыражения

$$A1\Delta\ B\Delta\ EXP\Delta\ X\Delta\ A1\times B\Delta\ EXP*X\Delta\ A1\times B+EXP*X$$

Аргументами связи + являются подвыражения $A1\times B$ и $EXP*X$; аргументами связи $\times$ являются подвыражения $A1$ и B ; аргументами связи $*$ являются подвыражения EXP и X .

Для того, чтобы зафиксировать характер арифметических действий в выражениях, определим понятия внешней связи и связи вещественного характера.

ОПРЕДЕЛЕНИЕ 6. Связь α_i , входящую в выражение $\alpha_1\alpha_2\ldots$ $\ldots\alpha_n$, будем называть внешней связью этого выражения, если не существует такой пары скобок $\{\alpha_j,\alpha_k\}$ в этом выражении, что $j<i<K$ и символ α_{j-1} есть одна из букв $\underline{Я}$, $\underline{В}$, $\underline{Ф}$, $\underline{У}$.

Например, в выражениях

$$\underline{Ф}\,(|40001+K\times L)\Delta\ \underline{Я}2100\Delta\ \underline{B}\,(5.\times 6.)\Delta$$
$$\underline{У}\,(\underline{Ф}4000+1+\underline{Я}141)$$

внешних связей нет.

В выражении $|4000|+K\times L$ внешними связями являются + и $\times$. В выражении $\underline{Ф}\,(|4000|+У+K)\times\underline{B}\,(5.+6.)/A$ внешними свя-

зями являются × и / .

ОПРЕДЕЛЕНИЕ 7. Будем говорить, что связь α_i выражения $\alpha_1 \alpha_2 \dots \alpha_n$ носит вещественный характер, если она отлична от символов $\vee$ и $\wedge$ и существует такая пара скобок $\{\alpha_j, \alpha_k\}$, что $j < i < k$, α_{j-1} есть буква $\underline{B}$ и связь α_i является внешней связью подвыражения $\alpha_{j+1} \alpha_{j+2} \dots \dots \alpha_{k-1}$, заключенного в скобки α_j и α_k. Будем говорить, что прочие связи данного выражения имеют условный характер.

В выражении $\underline{B}\,(5.\times 6.)$ связь × имеет вещественный характер. В выражении $\underline{B}\,(\underline{\Phi}\,(A\times B + 140001)/5.)$ связи × и + имеют условный характер, связь / имеет вещественный характер. В выражении $A + \underline{B}\,(A+B)$ первая слева связь + имеет условный характер, вторая – вещественный.

Транслятор заменяет каждое выражение программой вычисления его значения. Под значением выражения понимается двоичный код, занимающий целую ячейку памяти. В зависимости от контекста этот код может интерпретироваться как вещественное число, условное число, константа, адрес ячейки или номер формулы. Определим значение выражения через значения его подвыражений следующим рекурсивным способом.

ОПРЕДЕЛЕНИЕ 8. (а) Под значением выражения, состоящего из одного числа (константы) понимается само число (константа). Значение слова $\underline{S}\alpha$ совпадает со значением выражения $|\alpha|$.

(б) Значение слова $\underline{Я}\alpha$, где α – целое восьмеричное число, есть содержимое ячейки с адресом α.

(в) Значение слова $\underline{S}(\alpha)$, где α – целое восьмеричное число, есть код элементарного символа, дополненный слева нулями, который записан в позиции, указанной константой настройки в ячейке с адресом α.

(г) Значение слова $\underline{\Phi}\alpha$, где α – целое восьмеричное число, есть результат численной реализации формулы с номером α.

(д) Значение любого слова с подчеркнутой первой буквой и отличного от перечисленных выше, есть номер оператора, идентификатором которого является данное слово.

(е) Значение любого слова, являющегося меткой, есть адрес

этой метки.

(ж) Если слово α отлично от всех имен массивов (см. § 3), от слов SIN, ENTR, MOD, ATG, COS, EXP, LN, ASIN, TG и от слов, значения которых указаны выше, то значение слова есть содержимое ячейки, сопоставленной транслятором данному слову.

(з) Значение выражений (τ), $\underline{У}(\tau)$, $\underline{В}(\tau)$ совпадает со значением выражения τ.

(и) Значение выражения $\underline{Ф}(\tau)$ есть результат численной реализации формулы, номер которой задан двенадцатью младшими двоичными разрядами значения выражения τ.

(к) Значение выражения $\underline{Я}(\tau)$ есть содержимое ячейки, адрес которой задан пятнадцатью младшими разрядами значения выражения τ.

(л) Значения выражений $ASIN*\tau$, $ENTR*\tau$, $MOD*\tau$, $ATG*\tau$, $COS*\tau$, $EXP*\tau$, $LN*\tau$, $SIN*\tau$, $TG*\tau$, где τ есть выражение с главной связью ∞, есть числа $\arcsin t$, $entire\ t$, $mod\ t$, $\cos t$, e^{τ}, $\ln t$, $\sin t$, $tg\ t$. Вещественное число t задается нормализованным кодом значения выражения τ.

(м) Значение выражения $\mu*\tau$, где μ есть имя массива и τ – выражение с главной связью ∞, есть К-й элемент массива μ, где целое число K задается значением выражения τ.

(н) Если главная связь + подвыражения $\tau_1 + \tau_2$ выражения ω носит в ω вещественный (условный) характер, то значением этого подвыражения является вещественное нормализованное (соответственно условное) число, полученное в результате арифметического (соответственного циклического) сложения вещественных нормализованных (условных) значений выражений τ_1 и τ_2.

(о) Значения подвыражений $\tau_1 - \tau_2$, $\tau_1 \times \tau_2$, τ_1 / τ_2 и $\tau_1 \uparrow \tau_2$ выражения ω определяются аналогично. При делении условных чисел в качестве результата выбирается ближайшее целое.

(п) Связи $\vee$ и $\wedge$ всегда выполняются как поразрядное логическое сложение и умножение.

В связи с особенностями реализации Авто-Аналитика на

БЭСМ-6 некоторым словам сопоставляются фиксированные ячейки. Соответствие задается таблицей I3.

Слово	Ячейка	Комментарии
Л	00240	Ячейки стандартных программ чтения и записи
М	00241	
П	00242	
N	00243	номер формулы
ТЭТА	02047	признак из операторов подстановки
N1	02046	служебный номер формулы
ЭПС	02100	точность округления до нуля в операторе численной реализации формул арифметик

Таблица I3

§ 3. СТРУКТУРА ИСХОДНОЙ ИНФОРМАЦИИ

Вся исходная информация, вводимая в ЭЦВМ, представляет собой совокупность строк.

ОПРЕДЕЛЕНИЕ 9. Строкой называется последовательность символов входного языка Авто-Аналитика, имеющая вид

<имя строки>: <тело строки> ↑

Здесь и далее признаком конца строки служит символ ↑, имеющий на БЭСМ-6 код I75. Именем строки может быть любая последовательность символов, отличных от символов : и ↑. При идентификации учитываются только первые шесть символов имени, прочие безразличны. Телом строки может быть в принципе любая последовательность символов, отличных от символа ↑. В системе Авто-Аналитик предполагается, что набор строк может модифицироваться пользователем. В настоящее время предусмотрена трансляция следующих видов строк.

I. Формулы.
2. Параметры системы.
3. Массивы.
4. Ручные программы.
5. Операторы.

Рассмотрим перечисленные типы строк подробнее.

I. ФОРМУЛЫ

Строка, задающая формулу, имеет вид

[<номер формулы>] ; <формула>

Номером формулы может быть любое восьмеричное число от I до 7777. Можно указывать только значащие цифры номера. Сама формула записывается без граничных символов после знака: . Окаймление ее символами Δ происходит автоматически при трансляции. Условия определения I.I не проверяются. Поэтому в качестве формул можно вводить любые последовательности элементарных символов.

Вещественные и условные числа в формулах записываются в соответствии с определениями § I, причем условные числа должны находиться в интервале $0,I, \ldots, 2047_{IO}$. Вещественные числа транслируются в четыре символа внутреннего языка, целые – в один. Допускается запись вещественных чисел с двойной точностью, признаком которых служит буква Д на месте подстрочной десятки. Вещественные числа с двойной точностью транслируются в восемь символов внутреннего языка. Для записи символов, не имеющих представления во входном языке, необходимо использовать прямые скобки |...| . Внутри прямых скобок может быть записано целое число кодов символов внутреннего языка (для БЭСМ-6 – восьмеричных тетрад), разделенных запятыми. Транслятор переписывает содержимое прямых скобок в формулу, опуская запятые.

Приведем несколько примеров задания формул.

(а) Пусть требуется под номером 0020 ввести формулу, состоящую из одной буквы А . Для этого достаточно расписать строку

[0020]: А ↑ .

или

[20]: А ↑

(б) Под номером I000 ввести формулу $А \times 5. + Б$

[I000]: А×5. + Б ↑

(в) Требуется ввести сумму трех букв внутреннего языка с кодами 3400, 2405, 2402, не имеющих аналогов в алфавите входного языка, под номером 7000.

[7000]: |3400| + |2405| + |2402| ↑

или, что то же

[7000]: |3400, 0024, 2405, 0024, 2402| ↑

Связь + имеет код 0024.

(г) Требуется ввести под номером 1100 многочлен

$$a_1 \times 5. + a_2 \times 10. + a_3 \times 576000 - a_4 - a_5 - 3,14159257254$$

В качестве признака индекса в Авто-Аналитике обычно используются связь ¯ (надчерк).

[1100]: A¯1×5.+A¯2×$_{10}$1+A¯3+0.576$_{10}$6+-1.×A¯4+
-1.×A¯5+-0.314159257254≥1 ↑

Можно вводить последовательность формул, прописанных под одним номером. В этом случае формулы последовательности (или, иначе говоря, массива формул) разделяются парой символов ◊◊ . При трансляции символы ◊◊ заменяются граничным символом Δ и каждая формула последовательности расписывается, начиная с нулевой позиции очередной ячейки.

Пусть требуется под номером 1000 прописать массив формул

А, Б, В, Г+А

[1000]: А◊◊Б◊◊В◊◊Г+А ↑

2. ПАРАМЕТРЫ СИСТЕМЫ

Под параметрами системы будем понимать совокупность констант, задающих распределение памяти, признаки коммутативных и ассоциативных связей, а также указания на способ запуска системы.

Распределение памяти задается при помощи следующих строк.

АОРП: <адрес> ↑

ΔОРП: <количество ячеек> ↑

Например:

АОРП: 6000 ↑ ΔОРП: 20000 ↑

По заданным строкам транслятор установит начало рабочего поля в оперативной памяти с ячейки 06000, длина рабочего поля – 20000 ячеек. Адрес и количество ячеек задаются в восьмеричной системе.

Строка

NMAX: <номер формулы> ↑

задает верхнюю границу номеров формул, которые можно записывать на рабочее поле внешних запоминающих устройств.

Коммутативные и ассоциативные связи задаются простым перечислением в виде строк с именами ША (Шкала Ассоциативных связей) и ШК (Шкала Коммутативных связей). Связи перечисляются через запятую. Коды связей на внутреннем языке заключаются в прямые скобки. Например:

ША: +, x, ; ↑

ШК: +, x ↑

То же самое можно записать в виде строк

ША: |0024| , |0034| , , |0074| ↑

ШК: |0024| , |0034| ↑

Указание на способ запуска системы производится при помощи строк вида

ВХОД: <адрес> ↑

или

READ: <последовательность предложений> ↑

При наличии указателя ВХОД после окончания трансляции управление передается на ячейку с заданным адресом. При наличии указателя *READ* начинает выполняться последовательность предложений, заданных в строке и транслированных аналогично предложениям операторов(см.ниже).

3. МАССИВЫ

Входной язык Авто-Аналитика допускает ввод одномерных массивов. Транслятор не производит автоматического распределения памяти для массивов, поэтому строка, задающая некоторый массив, должна иметь вид

МАССИВ: <имя массива> , <адрес> , <первый элемент массива> , ..., <К-й элемент массива> ↑

Массив занимает в МОЗУ К ячеек, начиная с указанного адреса. Адрес указывается в восьмеричной системе. Элементами массива могут быть вещественные и условные числа, и также константы, записанные в соответствии с определениями первого параграфа. Именем массива может быть любое слово кроме *SIN*,

COS, TG, CTG, ASIN, ACOS, ATG, ACTG, EXP, LN

ENTR, MOD , включающее в себя не более шести символов.

ПРИМЕР. Пусть требуется ввести массив констант, состоящий из элементов 5., IO., I5, |I400| , А + Б, 5А с ячейки 06400. Присвоим массиву название КОНСТ и запишем

МАССИВ: КОНСТ, 6400, 5., IO., I5, |I400|, |А + Б| , |5А| ↑

4. РУЧНЫЕ ПРОГРАММЫ

Система допускает ввод в произвольное место МОЗУ программ, составленных в кодах машинного языка БЭСМ-6. Ручные программы вводятся в виде строк

ПРОГРАММА: <кодовой участок> ↑

Рассылка кодов в МОЗУ осуществляется в соответствии с вводными словами.

5. ОПЕРАТОРЫ

Программа, расписанная на входном языке Авто-Аналитика, транслируется в оператор БСО. Строка, задающая оператор, имеет вид

ОПЕРАТОР: <имя оператора>; <предложение> ;<предложение> ; ...; <предложение>; ↑

Именем оператора может быть любое слово (идентифицируются только первые шесть символов имени) или любой восьмеричный номер от 100 до 777. Собственно программа, реализующая данный оператор, расписывается в виде последовательности предложений, разделенных символом ; . Дальнейшие параграфы посвящены подробному описанию способа построения операторов.

§ 4. МЕТКИ, ТИПЫ ПРЕДЛОЖЕНИЙ

Во избежание путаницы в терминологии будем называть последовательность предложений, реализующих данный оператор, программой оператора (стандартным описанием) или просто программой. Каждому предложению программы может быть соотнесена метка. В качестве метки может выступать любое слово не длиннее шести символов, отделенное от соответствующего предложения двоеточием. Таким образом, предложение с меткой имеет вид

<МЕТКА> : <предложение>;

Если в предложениях оператора встречается метка, то при трансляции она заменяется соотнесенным ей относительным адресом.

В соответствии с выполняемыми функциями будем различать следующие типы предложений.

I. ПРЕДЛОЖЕНИЕ ПЕРВОГО РОДА. Всегда начинаются именем оператора, вслед за которым через запятую перечисляются его аргументы. Исключение составляет предложение с оператором := , речь о котором пойдет ниже. Предложение первого рода имеет вид <имя оператора> ,<аргумент> , ..., <аргумент>; Допускаются пустые предложения с меткой или без нее. При трансляции пустому предложению соотносится нулевая ячейка.

2. ПРЕДЛОЖЕНИЯ ВТОРОГО РОДА. Предназначены для ввода формул в программу оператора. Предложения второго рода всегда начинаются меткой, первые три буквы которой – ФОР – фиксированы. После букв ФОР удобно ставить порядковый номер вводимой в программу формулы (например, ФОРI, ФОРI9), хотя никаких ограничений на те три символа, которые можно включить в метку после букв ФОР, не накладывается. Приведем простой пример. Пусть требуется ввести формулы $А, Б, С$ с меткой ФОРI7. Запишем предложение

ФОРI7: $А \Diamond\Diamond Б \Diamond\Diamond С$;

Отметим, что в силу специфической организации ряда операторов Авто-Аналитика символ внутреннего языка с кодом 7777 в предложения второго рода вводить нельзя.

3. ПРЕДЛОЖЕНИЯ ТРЕТЬЕГО РОДА. Предназначены для ввода кодов машинного языка БЭСМ-6 в программу оператора и имеют вид

КОД, <кодовый участок>;

Адресная часть команд кодового участка может включать в себя произвольные слова и метки, заключенные в круглые скобки. Следует помнить о том, что эти адреса следует промодифицировать первым индексным регистром. Портить первый регистр в программе запрещается. Ячейки кодового участка, на которые необходимо передать управление, либо осуществлять некоторые преобразования (короче говоря, ячейки, номера которых нужно

включить в адресную часть какой-либо команды), необходимо помечать. При росписи кодового участка вместо номера ячейки пишется соответствующая метка. В какой-то степени указанный прием аналогичен программированию в условных адресах. При исполнении транслированной программы управление передается на первую ячейку кодового участка.

Рассмотрим примеры задания предложений третьего рода. Пусть требуется сложить условные числа 1,2,5 и прервать счет. Программа может иметь вид

```
      . . . . . .
      . . . . . .;
      КОД,
   01     010   (R1)      складываются числа 1,2,5 и результат
   01     013   (R2)      записывается в ячейку
   01     013   (R3)
   01     000   (SUM)
   01     300   (STOP)    выход из кодового участка на метку
   00     000    0000

R1:   C  0000   0000 }
         0000   0001 }
R2:   C  0000   0000 }   исходные числа
         0000   0002 }
R3:   C  0000   0000 }
         0000   0005 }

SUM: Ч0;                 ячейка результата
STOP:        СТОП;       оператор останова СТОП с меткой STOP
```

При росписи кодовых участков следует иметь в виду, что установлен режим блокировки округления и нормализации. Этот же режим должен быть и на выходе из кодового участка.

Эта же программа может быть записана следующим образом

```
        КОД,
 К   00  0I0   0I4I        сложение констант 0000 0000 0000
     00  0I3   0I42        0001 и 0000 0000 0000 0002, записан-
 К   0I  0I3   (R5)        ных в ячейках БСП
     0I  000   (SUM)
 К   0I  300   (STOP)
     00  000   0000
R5: С    0000  0000
         0000  0005
SUM: Ч0;
STOP:    СТОП;
```

Приведем еще пример. Пусть найдены два условных числа A_1 и A_2. В кодовом участке требуется найти максимальное из них и присвоить его значение слову МАХ с последующим остановом.

```
        КОД,
  К     0I    0I0    (AI)
        0I    005    (A2)
  К     0I    260    (M2)
        0I    0I0    (A2)
  К     0I    000    (MAX)
        0I    300    (MI)
M2: К   0I    0I0    (AI)
        0I    000    (MAX);
MI:     СТОП;
```

§ 5. ЛОГИЧЕСКИЕ ОПЕРАТОРЫ АВТО-АНАЛИТИКА

Логика программы организуется при помощи следующих операторов

I. : =	оператор присвоения
2. ЕСЛИ	оператор условного перехода
3. НА	оператор безусловного перехода
4. ВЫХОД	оператор выхода из подпрограммы
5. ЦИКЛ	оператор цикла

6. КОНЕЦ — оператор выхода из программы
7. СТОП — оператор останова
8. АРГ — оператор выдачи аргумента

Рассмотрим подробно работу и способ использования этих операторов.

ОПЕРАТОР ПРИСВОЕНИЯ

Оператор присвоения в Авто-Аналитике имеет обычный смысл, хотя его возможности и расширены по сравнению, например, с алгольным оператором присвоения. Можно присваивать значения:

а)СЛОВАМ; слева от оператора : = находится произвольное слово, справа - произвольное выражение;

б)ЯЧЕЙКАМ; слева от оператора : = находится слово $\underline{Я}\alpha$, где α - целое восьмеричное число, либо выражение $\underline{Я}(\tau)$. Число α либо значение выражения τ задают адрес ячейки, в которую заносится значение выражения, расположенного справа от оператора присвоения;

в)ЭЛЕМЕНТАМ МАССИВА; слева от оператора : = последовательность символов вида $\mu * \tau$, где μ - имя массива, τ - - выражение, значение которого задает номер элемента массива;

г)ФОРМУЛАМ; слева от оператора : = расположена последовательность символов $\underline{Ф}\alpha$ (восьмеричное число α задает номер формулы) либо $\underline{Ф}(\tau)$ (значение выражения τ задает номер формулы). Выражение справа от оператора присвоения обрабатывается особым образом и прописывается как формула с заданным номером. При этом подвыражения вида $\underline{У}(\omega)$ (ω - выражение) или $\underline{Я}\alpha$ (α - целое восьмеричное число) заменяются своими значениями, задающими символы внутреннего языка Авто-Аналитика. Символы с нулевым кодом опускаются. Подвыражения вида $\underline{Ф}\alpha$ (α задает номер формулы) или $\underline{Ф}(\omega)$ (значение выражения ω задает номер формулы) заменяются соответствующими формулами. Подвыражение $[\underline{Ф}\alpha]$ или $[\underline{Ф}(\omega)]$ заменяются соответствующими формулами после их численной реализации, то есть после проведения в них всех возможных арифметических действий над числами. Прочие символы выражения справа от оператора : = переписываются в результирующую формулу без изменений. Константы в прямых скобках в выражении справа записываются как в формулах, если они не входят в вычисляемое подвыражение. Подвыражения $[\omega]$ заменяются условным значением ω .

д)и, наконец, если слева от оператора : = находится последовательность символ вида $\underline{S}(\alpha)$, где α - целое восьмиричное число, то выражение справа обрабатывается как и в предыдущем случае и переписывается по заданной настройке (из ячейки с адресом α).

Рассмотрим несколько примеров использования данного оператора.

I.Слову SUM требуется присвоить значение суммы числа 5 и результата реализации формулы с номером I000. Записываем

$$SUM := \underline{B}\ (5. + [\Phi\ 1000]);$$

2.В ячейку с адресом 02I00 (точность для Арифметика в БСП) требуется заслать число 10^{-6}.

$$\underline{Я}\ 2100 := {}_{10}-6;$$

Того же результата можно добиться при помощи предложения

$$ЭПС := {}_{10}-6;$$

3.Элементу с номером К + 20 массива констант КОНСТ требуется присвоить значение (условное) К х 5 + I

КОНСТ * (К + 20) : = К х 5 + I;

Так как условное число I записано в ячейке 00I4I из БСП, то можно переписать предложение следующим образом

КОНСТ * (К + 20) : = К х 5 + ЯI4I

4.В некотором кодовом участке программы меткой MI помечена рабочая ячейка. Ей требуется присвоить значение $3,I4I592_{10}$ I;

$$\underline{Я}\ (MI) : = 0.3I4I592_{10}\ I;$$

5.Под номером I000 требуется прописать последовательность символов (А + Б).

Для этого достаточно записать

$$\underline{\Phi}I000 : = (A + Б);$$

Так как коды открывающей и закрывающей скобок хранятся соответственно в ячейках 00I42 и 00I43, то можно записать

$$\underline{\Phi}I000 : = \underline{Я}142\ A + Б\ \underline{Я}143;$$

Код связи + равен 0024. Можно записать

$$\underline{\Phi}I000 : = \underline{Я}I42\ A \quad |0024| \quad Б\ \underline{Я}\ I43;$$

Константа в прямых скобках расписана как при кодировании формул.

6.Пусть формулу $K \times L$ требуется образовать младшими дву-

мя тетрадами величины A .

$$\underline{\Phi}\,(\,К\times\angle\,) := [\,A \wedge |7777\quad 7777|\,]\ ;$$

Значением условного выражения справа от оператора присвоения являются младшие тетрады (24 двоичных разряда) из ячейки, соотнесенной букве А. Условное выражение вычисляется, поэтому константа в нем записана в соответствии с определениями § I.

7.Пусть формулу II00 требуется прописать также под номером II0I

$$\underline{\Phi}\ \text{II0I} := \underline{\Phi}\text{II00}$$

8.Под номером 2000 требуется прописать формулу, образованную суммой формул, номер первой из которых записан в ячейке А, номер второй равен I777.

$$\underline{\Phi}\ 2000 := \underline{\Phi}\ (A) + \underline{\Phi}\ 1777;$$

9.По настройке в ячейке 00236 требуется записать последовательность символов, заданную значением произведения 5 х 6 плюс букву А, плюс численную реализацию формулы II00, плюс произведение 5 х 6, плюс букву А с индексом, образованным значением условного выражения А х Б + 5.

$$\underline{S}(236) := \underline{B}(5.\times 6.) + A + [\underline{\Phi}\text{II00}] + 5.\times 6. + A^{-}\ [\text{АхБ+5}];$$

ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА

Общий вид обращения к оператору условного перехода ЕСЛИ, <выражение> $\overset{\leq}{=}$ <выражение> , ..., <выражение> $\overset{\leq}{=}$ <выражение> , ↄ , <метка> ;

Производится переход на заданную метку, если выполнена вся последовательность условий. Проверка условий $\leq$ и $<$ производится при помощи арифметического вычитания, если хотя бы одно выражение, входящее в неравенство, является вещественным, и циклическим вычитанием в противном случае. При помощи знака равенства можно производить проверку совпадения тех или иных символов, а также проверку вхождения элементов алфавита в подмножества алфавита. Для проверки вхождений выделены следующие служебные слова:

СВЯЗЬ	– подмножество Σ ;
БУКВА	– подмножество букв (постоянных и переменных);
Ф БУКВА	– подмножество переменных букв Ф;
Х БУКВА	– подмножество Φ_0 ;
У БУКВА	– подмножество Φ_1;

Z БУКВА – подмножество Φ_2;
P БУКВА – подмножество Φ_3;
Q БУКВА – подмножество Φ_4;
R БУКВА – подмножество Φ_5;
S БУКВА – подмножество Φ_6;
T БУКВА – подмножество Φ_7;
K СВЯЗЬ – коммутативные связи $\Sigma_к$;
A СВЯЗЬ – ассоциативные связи $\Sigma_а$;
П БУКВА – подмножество постоянных букв Ω ;
ЧИСЛО – подмножество чисел R ;
ДЕЛЬТА – граничный символ.

Оператор ЕСЛИ проверяет вхождение символа, код которого задан младшими разрядами значения выражения в левой части равенства, в множество, заданное служебным словом в правой части равенства.

Приведем примеры обращений к оператору условного перехода.

1. ЕСЛИ, $\underline{S}$ (236) = К СВЯЗЬ, то , МI;
2. ЕСЛИ, $\underline{\Phi}$1000 $\leq$ $\underline{B}$(5.), $\underline{S}$(210)= ДЕЛЬТА, то, выход;
3. ЕСЛИ, А х 5 < Б, $\underline{B}$(SIN*X + I.) < I.5, то, М;
4. ЕСЛИ, А = | (| ,то МI ;
5. ЕСЛИ, А = | Б | , $\underline{\Phi}$1000 $\leq$ $\underline{\Phi}$1001 + ЭПСИЛОН, ЭПСИЛОН < ЭПС, то МЕТКА;
6. ЕСЛИ, $\underline{S}$(235)= |, |, $\underline{S}$(234)= ДЕЛЬТА, |,| = К СВЯЗЬ, АВС < $\underline{\Phi}$1260, то, АВОСТ;

ОПЕРАТОР БЕЗУСЛОВНОГО ПЕРЕХОДА

Обращение к оператору безусловного перехода НА имеет вид

НА, <метка> ;

При помощи оператора НА организуется также обращение к подпрограммам. Метка в этом случае должна иметь вид Ш α , где α – целое восьмеричное число. Работа подпрограммы должна завершаться предложением

ВЫХОД ;

ОПЕРАТОР ЦИКЛА

Обращение к оператору имеет вид

ЦИКЛ, <метка> , <слово> , <выражение> ;

Данный оператор повторяет последовательность предложений, сле-

дующих за ним, до предложения с заданной меткой включительно. Цикл повторяется К раз, где К – значение условного выражения в обращении к оператору ЦИКЛ. При этом слово, указанное в обращении, принимает значение I, ..., К (условные числа). В теле цикла можно изменять значение слова, что соответственно скажется на числе повторений цикла. Изменение значений слов, входящих в выражение, на число повторений цикла не влияет.

Примеры.

ЦИКЛ, МI, К , 5;

ЦИКЛ, ЦИКЛ I, I, К×L+1 ;

ЦИКЛ, МЕТКА, СЛОВО, КОНСТ;

ОПЕРАТОР ВЫХОДА ИЗ ПРОГРАММЫ

Последнее исполняемое предложение программы оператора должно иметь вид

КОНЕЦ;

Это предложение служит признаком выхода из программы оператора.

ОПЕРАТОР ОСТАНОВА

Прерывание работы системы осуществляется предложением вида СТОП, <текст> ;

На печать выдается информация:

РАБОТА СИСТЕМЫ ОКОНЧЕНА ОПЕРАТОРОМ СТОП.

Далее печатается текст из обращения. Текст имеет произвольный вид без символов ; и Ⓕ.

Если использовать оператор STOP, то прерывание осуществляется запрещенной командой, после которой выдается стандартная сбойная информация Авто-Аналитика.

ОПЕРАТОР ВЫДАЧИ АРГУМЕНТА

Особую роль, связанную с операторной организацией программ в Авто-Аналитике, играет оператор выдачи аргумента АРГ. Обращение к нему имеет вид

АРГ, <слово> , <слово> , ..., <слово> ;

Указанное предложение обычно ставится в начале программы оператора. Словам, перечисленным в обращении, присваиваются значения выражений, образующих аргументы реализуемого оператора. Короче говоря, слова в обращении к АРГ являются исходной информацией для программы оператора.

Значение выражений из обращения к оператору разбивается на тетрады. Каждому слову обращения к оператору АРГ сопостав-

ляется одна тетрада обращения. Если слово начинается с буквы $\mathcal{Z}$, четыре последовательные тетрады объединяются в ячейку (слева направо) и сопоставляются (задают значение) данному слову.

§ 6. БИБЛИОТЕКА ВНУТРЕННИХ ОПЕРАТОРОВ

Под внутренними операторами входного языка системы Авто-Аналитик будем понимать операторы, которые реализируются одной из стандартных программ ЕСП. Предложение, задающее обращение к внутреннему оператору, транслируется в обращение к соответствующей

№	Название	Аргументы	Назначение
I.	ЧВ	настройка, слово	чтение вперед
2.	ЧН	-"-	чтение назад
3.	ЧТ	-"-	чтение без перестройки
4.	ЧВФ	настройка	$iA + I$
5.	ЧНФ	настройка	$iA - I$
6.	ЗВ	настройка, слово	запись вперед
7.	ЗН	настройка, слово	запись назад
8.	ЗП	настройка, слово	запись без перестройки
9.	ЗПВ	слово	запись вперед по $\underline{S}$ 236
I0.	ЗПН	слово	запись назад по $\underline{S}$ 237
II.	ЧЧВ	настройка, слово	чтение вперед числа
I2.	ЧЧН	-"-	чтение назад числа
I3.	ЗЧВ	-"-	запись вперед числа
I4.	ЗЧН	-"-	запись назад числа
I5.	ДИСПЕТЧЕР	выражение	диспетчер для резерва ОРП
I6.	ПРОПФ	выражение, настройка, настройка	прописка $\pi = 0$
I7.	ПРОПО	-"-	прописка $\pi = I$
I8.	ВЫДАЧА	выражение, настройка	выдача информации о формуле
I9.	УПАК	настройка	запоминание настроек
20.	РПАК	настройка	восстановление настроек

2I.	АНСТР	выражение	аннулирование строки ОАС
22.	ОТКАЗ	-"-	отказ от формулы
23.	АНФ	-"-	аннулирование формулы
24.	СДВИГ	настройка,настройка, выражение	сдвиг участка ОРП
25.	КФI	настройка	конец формулы РПI
26.	КФ2	-"-	конец формулы РП2
27.	ПАС	выражение	поиск строки ОАС
28.	АНОЗУ	выражение	аннулирование формулы в ОЗУ
29.	РФ	выражение,выражение	раздвоение формулы
30.	ИНФ	-"-	изменение номера формулы
3I.	НОРМ	выражение	приведение формулы к почти нормальному виду
32.	ПРОПН	выражение,настройка	прописка по настройке
33.	ЧВВС	настройка,слово	чтение вперед вне скобок
34.	ЧНВС	настройка,слово	чтение назад вне скобок

Таблица I4.

стандартной программе. Список внутренних операторов задан таблицей I4. Описание работы этих операторов здесь мы приводить не будем, так как соответствующие описания приведены для стандартных программ.

§ 7. ПРИМЕРЫ ПОСТРОЕНИЯ ПРОГРАММ НА ВХОДНОМ ЯЗЫКЕ

I. Из массива чисел 5., I0., I.5, 6, требуется выбрать максимальное, прервать работу системы, если оно больше нуля и выйти из оператора в противном случае.

Массив чисел введем с ячейки 6000 и назовем ЧИСЛА.

МАССИВ: ЧИСЛА, 6000, 5., I0., I.5,6.④

Оператору присвоим имя МАХ. Максимальное число запишем в ячейку 02I00

```
0.          ОПЕРАТОР:  МАХ ;
1.          Я2100 : = ЧИСЛА*1    ;
2.          ЦИКЛ, ЦИКЛ, К, 4 ;
3.          ЕСЛИ, В(Я2100)  ≤  ЧИСЛА  *  (К + 2),то,ЦИКЛ ;
4.          Я2100 : = ЧИСЛА  *  (К+1)  ;
5. ЦИКЛ:    ;
6.          ЕСЛИ, В(Я2100)  ≤  0.,то; ВЫХОД  ;
7.          СТОП, число ⌣ меньше ⌣ нуля ;
10. ВЫХОД: КОНЕЦ ; ⓕ
```

Предложения пронумерованы в восьмеричной системе. Нумерация не перфорируется и в ЭЦВМ не вводится. При распечатке программы нумерация осуществляется автоматически.

II. В заданной формуле *N1* требуется найти главную связь и прописать ее как отдельную формулу с заданным номером *N2* . Предполагается, что связь ∞ кодируется символом 7777.

```
                    ОПЕРАТОР: ГЛ СВ ;
                    АРТ, № I, № 2 ;
                    УПАК,       ≤ 235 ;
                    ВЫДАЧА, № I,   ≤ 235 ;
                    РО : = |7777| ;
              М1 :  ЧВВС,  ≤ 235,  РI  ;
                    ЕСЛИ,  РI = СВЯЗЬ,  РI<РО,ТО, М2;
                    НА, М1 ;
              М2 :  ЕСЛИ,   РI = ДЕЛЬТА, ТО, М3 ;
                    РО : = РI ;
                    НА, М1 ;
              М3  :  РПАК, ≤235 ;
                    ОТКАЗ, № I ;
                    Ф(№2) : = [РО]  ;
                    КОНЕЦ ;
```

III. В заданную формулу № вместо заданной буквы Б вписан букву БI.

```
ОПЕРАТОР : ВПИСАТЬ;
АРТ, №, Б, БI ;
УПАК, ≤ 235 ;
ВЫДАЧА, № ,≤235 ;
```

```
M :     ЧВ,  S 235,   Б2 ;
        ЕСЛИ,  Б2 = Б, ТО, М1 ;
        ЕСЛИ,  Б2 ≠ ДЕЛЬТА, ТО, М2 ;
M1 :    ЧНФ,  S 235 ;
        S(235) : = У(Б1) ;
        НА, М ;
M2 :    РПАК, S 235 ;
        ОТКАЗ, № ;
        КОНЕЦ ;
```

ГЛАВА IУ

ОСНОВНЫЕ ОПЕРАТОРЫ АВТО-АНАЛИТИКА

Основную практическую ценность системы программирования Авто-Аналитик составляет библиотека операторов. В настоящей главе приведено описание основных операторов, включенных в библиотеку операторов Авто-Аналитика. Сведения об алгоритмах, положенных в основу операторов, необходимы для их правильного использования. Общий список операторов приведен в таблице I5.

§ I. ОПЕРАТОРЫ ПОДСТАНОВКИ

Рассмотрим операторы подстановки, наиболее часто используемые при решении задач в системе Авто-Аналитик.

Оператор ПОДСТ (OI77). Обобщенная коммутативная постановка

Данный оператор является, пожалуй, самым сложным оператором системы. Отметим сразу, что его использование на практике весьма ограничено. Причины этого будут изложены ниже.

Обращение к оператору имеет вид

ПОДСТ, A^-, A^+, B, C;

Здесь A^- – номер формулы, задающей левую часть правила;

A^+ – номер формулы, задающей правую часть правила, по которому обрабатывается формула B. Результат подстановки прописывается под номером C.

Оператор реализует операцию обобщенной коммутативной подстановки. Приведем описание алгоритма, положенного в основу данного оператора. Мы полагаем, что это описание может способствовать пониманию сложности тех проблем, которые стоят перед разработчиком программ общих аналитических преобразований, а также пониманию специфичности аналитических выкладок на ЭЦВМ по сравнению с численными или логическими.

Алгоритм, реализующий операцию обобщенной подстановки, включает в себя общую схему выделения аналога формулы A^- в B, подстановку вместо него соответствующего аналога формулы A^+ и элементы упорядочения возникающего перебора. В целом алгоритм можно рассматривать как попытку промоделировать на ЭЦВМ действия, совершаемые человеком при проведении формальных преобразо-

ваний по заданному правилу.

Основной проблемой является выделение в формуле В аналога A^- и аналогов входящих в A^- переменных букв. Для решения этой проблемы разработана рекуррентная процедура, которая осуществляет последовательное деление формул В и A^-, проверяя предварительные условия возможности представления части из В аналогом формулы A^-. Вначале рассмотрим общее описание алгоритма.

Этап I. Последовательная выборка всех частей формулы В, которые могут быть представлены аналогом A^-.

Этап II. Последовательное деление A^- на подформулы. Фиксация части из В, в которой следует произвести дальнейший перебор на этапе I. Переход к этапу I, если деление A^- осуществлено.

Этап III. Если деление A^- невозможно, проверяется совпадение буквы из A^- и выделенной части в В. Если буква в A^- переменная, то соответствующая часть в В полагается ее аналогом. Для деления на этапе II фиксируется очередная подформула в A^-. Производится переход к этапу II. Если проверка не удовлетворяется, переход к этапу I.

Этап IV. Если установлено полное соответствие между A^- и некоторой частью из В, производится непосредственно подстановка. Результат подстановки получается путем занесения в В аналога формулы A^+ вместо аналога формулы A^-. При этом переменные буквы из A^+ заменяются своими аналогами, выделенными на предыдущих этапах. Полученной формуле присваивается номер С. Алгоритм работу заканчивает.

Этап V. Если выбраны все части из В, и ни одна из них не является аналогом A^-, подстановка невозможна. Алгоритм заканчивает работу без формирования формулы С.

Приведенная схема весьма приближенно описывает алгоритм, реализующий операцию обобщенной подстановки. Ниже приведено более детальное описание алгоритма. Ясно, что алгоритм должен удовлетворять некоторым предварительным требованиям или пожеланиям.

Алгоритм должен включать в себя элементы упорядочения перебора при решении комбинаторной задачи, возникающей на первых трех этапах. Полный перебор даже для некоторых простых

преобразований практически неосуществим. Обычным приемом, которым пользуется человек при проведении аналитических выкладок, является выделение характерных признаков в правиле и обрабатываемой формуле и осуществление предварительного перебора по этим признакам без полного анализа формул A^{-} и B. Естественным будет использование этого же приема и в алгоритме обобщенной подстановки.

Алгоритм должен быть достаточно абстрактным и допускать реализацию на различных типах ЭЦВМ.

Отдельные блоки программы, реализующей алгоритм, должны быть четко выделены и организованы для того, чтобы облегчить отладку и иметь возможность реализовать различные модификации обобщенной подстановки без существенной переработки всей программы.

Можно разработать различные алгоритмы обобщенной подстановки, удовлетворяющие перечисленным требованиям. Доказать оптимальность того или иного из них повидимому невозможно, в силу сложности и разнообразия обрабатываемых объектов. Приведенный ниже алгоритм был принят после более или менее завершенной практической реализации нескольких вариантов.

Рассмотрим детальное описание алгоритма обобщенной подстановки, допускающего реализацию на любой достаточно мощной ЭЦВМ.

Несколько упрощает алгоритм предположение, что формулы A^{-} и B не имеют пустых вхождений. Это предположение не накладывает ограничений на структуру обрабатываемых формул, так как любая формула при помощи определенного служебного символа (в дальнейшем будем обозначать его буквой ξ) может быть приведена к виду без пустых вхождений. Использовать служебный символ при записи исходных формул нельзя. Во всех практических случаях это преобразование не сказывается на результате, из которого служебный символ убирается, однако полностью корректным оно станет лишь в предположении, что пустая подформула может быть аналогом тех и только тех типов переменных букв, что и символ ξ .

Приведем простой пример преобразования формулы к виду без пустых вхождений. Пусть формула имеет вид

$$\Delta + A + - Б \uparrow Ж * (SIN * Ж +) \Delta .$$

После преобразования получим

$$\Delta\ \xi + A + \xi - B \uparrow x \times (SIN * x + \xi)\ \Delta .$$

В дальнейшем будем предполагать, что формулы $\bar{A}$ и Б не имеют пустых вхождений.

Существенную роль в формулах играют пары скобок. Поэтому на некоторых этапах работы алгоритма анализ формул А и В ведется раздельно на каждом скобочном уровне.

ОПРЕДЕЛЕНИЕ 4.I. Пусть задана формула

$$A = \Delta\ \alpha_1\ \alpha_2 \ldots \alpha_n\ \Delta$$

Скобочным уровнем N ($N = 0,1, \ldots$) в формуле А назовем множество всех α_i, $i = 1, \ldots, n$, отличных от скобок и граничного символа Δ и таких, что в последовательности $\alpha_1 \alpha_2 \ldots \alpha_{i-1}$ раскрывающих скобок на N меньше, чем закрывающих. Аналогичным образом понятие скобочного уровня определяется для любой части из А. Ясно, что любой скобочный уровень может быть пустым. Легко также доказать следующую теорему.

ТЕОРЕМА 4.I. Если в формуле А скобочный уровень N является пустым, а скобочный уровень ($N+1$) не пуст, то в формуле А можно устранить пару скобок так, что все символы уровня ($N+1$) войдут в уровень N.

Практически важным является следствие из этой теоремы.

СЛЕДСТВИЕ. Если в почти нормальной формуле А пуст скобочный уровень N, то все последующие скобочные уровни также являются пустыми.

Пользуясь этим следствием, мы можем ограничивать процесс поскобочного анализа формул.

Как уже указывалось выше, в основе алгоритма лежит рекуррентная процедура, осуществляющая последовательное деление формул $\bar{B}$ и $\bar{A}$ на подформулы и сравнение этих подформул. Для фиксации деления перед началом работы основной части алгоритма расписываются специальные разметочные формулы $\bar{A}_I$ и B', длина которых равна соответственно длине формул $\bar{A}$ и $\bar{B}$.

ОПРЕДЕЛЕНИЕ 4.2. Пусть задана формула

$$B = \Delta\ \beta_1\ \beta_2 \ldots \beta_n\ \Delta$$

и соответствующая ей разметочная формула

$$B' = \Delta\ \beta'_1\ \beta'_2 \dots \beta'_n\ \Delta$$

Слоем i в формуле В назовем ее подпоследовательность

$$\beta_{i_1}\ \beta_{i_1+1} \dots \beta_{i_1+K_1+1}\ \beta_{i_2}\ \beta_{i_2+1} \dots \beta_{i_2+K_2+1} \dots \beta_{i_m}\ \beta_{i_m+1} \dots \beta_{i_m+K_m},$$

где

$$i_1, i_1+1, \dots, i_1+K_1,\ i_2, i_2+1, \dots, i_2+K_2, \dots,\ i_m, i_m+1, \dots, i_m+K_m$$ –

все без исключения номера членов β'_i формулы B′, равные i.

Пример. Пусть формула В имеет вид

$$B = \Delta\ A + Б + (x - y) + C\ \Delta\ .$$

Разметочная формула

$$B' = \Delta\ i\ K\ K\ K\ i\ i\ i\ i\ i\ K\ m\ \Delta.$$

Тогда слоем i в В будет подпоследовательность

$$A + (x - y),$$

а слоем К – подпоследовательность

$$+ Б + (\ +\ .$$

ОПРЕДЕЛЕНИЕ 4.3. В работе алгоритма участвует переменная величина ГА$^-$, именуемая границей в А$^-$ и представляющая собой адрес позиции (или, что то же, порядковый номер) в формуле А$^-$. Адрес соответствующей позиции в разметочной формуле A^-_I обозначим через ГА^-_I. Подпоследовательность вида $\alpha_k\ \alpha_{k+1} \dots \alpha_{k+m-1}$ из А$^-$ называется слоем в А$^-$, если символ α_k записан в позиции ГА$^-$ и $\alpha'_{k+m} = \Delta$, либо α'_{k+m} совпадает со специальным разметочным символом ∇ в A^-_I, а все $\alpha'_k, \alpha'_{k+1}, \dots, \alpha'_{k+m-1}$ отличны от Δ и ∇.

ОПРЕДЕЛЕНИЕ 4.4. Деление в А$^-$ суть процедура расстановки символов ∇ в A^-_I.

Деление слоя i в В производится выделением некоторой подпоследовательности в слое i формулы В и разметкой ее в В′ символом $(i + I)$.

Определим далее основные понятия, использованные при опи-

сании алгоритма упорядочения перебора.

ОПРЕДЕЛЕНИЕ 4.5. Узкими местами для данных слоев в A^- и B назовем элементы множества $\{\alpha_i\}_{i\in I}$, состоящего из элементарных символов, входящих в рассматриваемый слой A^- и таких, что выполнены следующие условия:

(а). Для каждого α_i $(i \in I)$ отношение кратности в слое A^- к кратности в слое B не меньше отношения кратности в слое A^- к кратности в слое B любого символа α_κ $(\kappa \notin I)$ из того же слоя A^-.

(б). α_i не является ни скобкой, ни переменной буквой, ни главной связью σ_β рассматриваемого слоя в B.

Вопрос о выборе числа узких мест (а значит и индексного множества I) на каждом слое зависит от конкретной реализации Авто-Аналитика.

ОПРЕДЕЛЕНИЕ 4.6. Минимально необходимыми условиями возможности подстановки (МНУ) назовем условие вида:

(*). Для каждого слоя A^-, включающего в себя переменные буквы, кратность любого символа α, отличного от переменной буквы, $\leq$ кратности того же символа α в соответствующем слое B.

(**). Для каждого слоя A^-, не включающего в себя переменных букв, кратность любого символа α = кратности того же символа α в соответствующем слое B.

ОПРЕДЕЛЕНИЕ 4.7. Проверкой МНУ назовем проверку условия (*) или (**).

Простой проверкой МНУ назовем проверку (*) или (**) только для узких мест.

Простая проверка МНУ позволяет произвести предварительную оценку возможности подстановки, которая уточняется далее проверкой МНУ. Предварительная оценка несколько сокращает время, необходимое для проведения перебора и составляет один из этапов работы алгоритма упорядочения перебора.

Приведенные определения позволяют описать основную структуру алгоритма (блок-схема 4.I.). Прочие необходимые понятия будут вводиться по мере надобности при описании конкретных блоков алгоритма.

(а). Предварительный блок

формулы

$$A^{-} = \Delta\ \alpha_1\ \alpha_2\ \dots\ \alpha_k\ \Delta$$

и

$$B = \Delta\ \beta_1\ \beta_2\ \dots\ \beta_n\ \Delta$$

преобразуются к виду без пустых вхождений указанным выше способом.

Расписываются разметочные формулы

$$A_1^{-} = \Delta\ \alpha_1'\ \alpha_2'\ \dots\ \alpha_k'\ \Delta$$

и

$$B' = \Delta\ \beta_1'\ \beta_2'\ \dots\ \beta_n'\ \Delta$$

где $\alpha_i' = 1$, $i = 1, \dots, k$, $\beta_j' = 1$, $j = 1, \dots, n$. Заводится переменная величина i, которая в дальнейшем будет соответствовать номеру рассматриваемого слоя в B. Первоначально $i := 2$.

Адрес позиции α_1 формулы A^{-} заносится в $ГА^{-}$, адрес позиции символа α_1' формулы A_I^{-} заносится в $ГА_I^{-}$.

(I). Выделение О-слоя

Для организации перебора производится предварительное разбиение формулы B на подформулы, которые именуются О-слоями. Опишем процедуру выборки О-слоев. При первом обращении к данному блоку О-слоем объявляется вся формула B. Повторные обращения производятся при неудачном выборе О-слоя, когда в основном цикле среди главных частей О-слоя не удалось выделить аналога A.

ОПРЕДЕЛЕНИЕ 4.8. Пусть задана формула

$$A = \Delta\alpha_1\alpha_2\dots\alpha_{i_1-1}\alpha_{i_1}\alpha_{i_1+1}\dots\alpha_{i_2-1}\alpha_{i_2}\alpha_{i_2+1}\dots\alpha_{i_k-1}\alpha_{i_k}\alpha_{i_k+1}\dots\alpha_n\Delta,$$

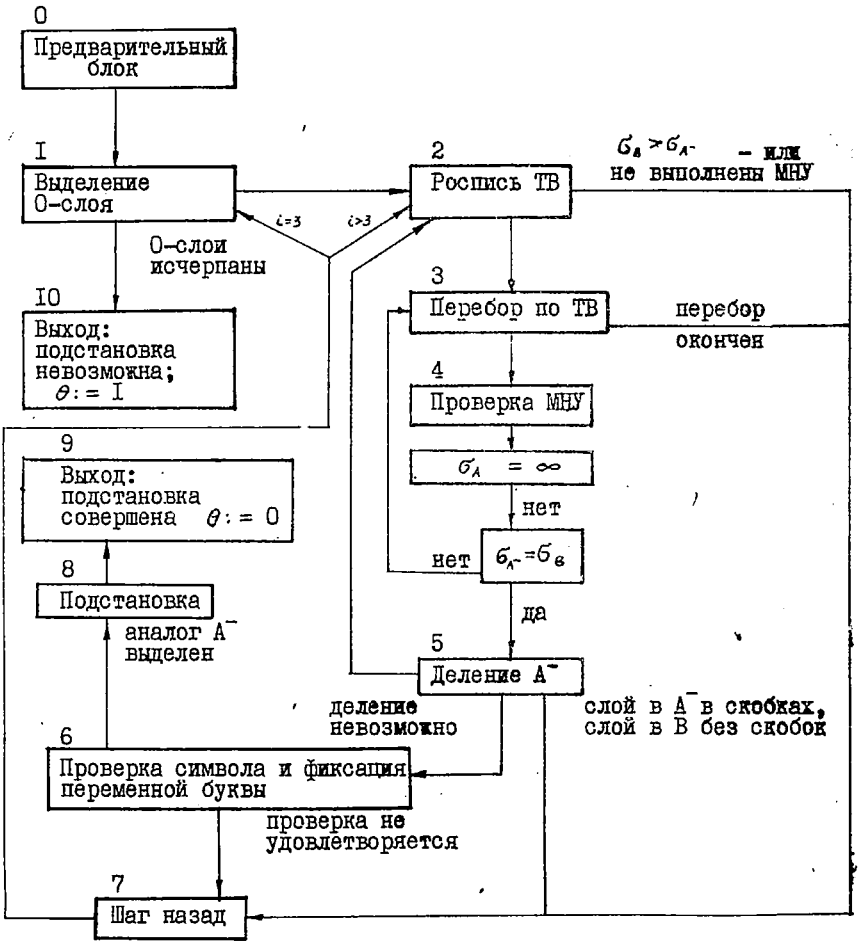
в которой $\alpha_{i_1} = \alpha_{i_2} = \dots = \alpha_{i_k}$ – все без исключения главные связи формулы A. Подформулы

$$\Delta\ \alpha_1\alpha_2\dots\alpha_{i_1-1}\ \Delta\ \alpha_{i_1+1}\ \dots\alpha_{i_2-1}\ \Delta\ \dots\ \Delta\ \alpha_{i_k+1}\ \dots\ \alpha_n\ \Delta$$

будем называть главными подформулами формулы A.

Произвольную часть формулы A, составленную из главных подформул, будем называть главной частью формулы A.

Аналогично определяется понятие главной подформулы и главной части для произвольной части из A.



Блок-схема 4.I.

Блок (I) выбирает и объявляет последовательно О-слоями все подформулы формулы В, которые не являлись главной частью ни одного из прежних О-слоев. О-слой различается в B' символом $i = 2$. Позиции B', не соответствующие О-слою, заполняются единицами.

Перебор О-слоев производится без упорядочения. Однако их число сравнительно невелико и перебор занимает незначительное время, что следует из теоремы 4.2.

Теорема 4.2. Число О-слоев не превосходит n, где n - суммарное количество букв, связей и пар скобок в В.

Доказательство. Пусть в В нет скобок. Очевидно, О-слоем являются все главные подформулы формулы В, число которых не превосходит числа главных связей в В. Далее можно провести деление выбранных О-слоев, причем число главных подформул в них снова не превосходит числа связей. Продолжая рассуждение, получим доказательство теоремы.

Если в формуле В имеются пары скобок, то следует учесть, что О-слоем может быть как выражение в скобках, так и внутренность этих скобок. Очевидно, теорема остается справедливой.

(2). Роспись ТВ

Для организации перебора в очередном слое расписывается специальная таблица, в которую заносятся некоторые результаты анализа слоя в В и соответствующего ему слоя в A^-. Эта таблица именуется таблицей вариантов (ТВ). Каждому слою i ($i = 2,3,...$) в ТВ расписан i-й массив. Массив состоит из участка ТВ и заголовка участка ТВ.

Перед росписью ТВ выделяются узкие места в рассматриваемом слое. Определяются σ_B - главная связь слоя в В и σ_{A^-} - главная связь слоя в A^-. При выделении узких мест легко произвести проверку МНУ. Если МНУ не выполняются или $\sigma_B > \sigma_{A^-}$, то производится переход к блоку (7) "шаг назад", так как вариант слоя выбран неудачно. В противном случае расписывается массив ТВ. Слой в В делится на главные подформулы. Подсчитывается длина слоя в A^- и каждой главной подформулы слоя в В. Каждой главной подформуле соответствует строка массива ТВ. В строку заносится следующая информация:

а) кратность каждого из узких мест в данной подформуле;

б) позиция начала подформулы;

в) длина подформулы;

г) в специальный разметочный разряд заносится нуль.

Далее расписывается заголовок участка, который включает в себя:

а) σ_B – главную связь слоя в B;

б) кратность узких мест слоя A^-;

в) типы переменных букв, входящих в слой A^-;

г) кратность главной связи в слое A^-;

д) число строк в участке ТВ;

е) длину слоя A^-;

ж) тип перебора.

Тип перебора определяется на основе анализа формул A^- и B указанным в следующей таблице способом.

$i=2,\ \sigma_{A^-} = \sigma_B \in \Sigma_K$	–	тип 2
$i=2,\ \sigma_{A^-} = \sigma_B \in \Sigma_a \setminus \Sigma_K$	–	тип 4
$i=2,\ \sigma_{A^-} = \sigma_B \in \Sigma_a$	тип 6	тип 5
$i=2,\ \sigma_{A^-} > \sigma_B$	тип 6	тип 3
$i>2,\ \sigma_{A^-} = \sigma_B \in \Sigma_K$	тип 2	тип 2
$i>2,\ \sigma_{A^-} = \sigma_B \in \Sigma_a \setminus \Sigma_K$	тип I	тип I
$i>2,\ \sigma_{A^-} = \sigma_B \notin \Sigma_a$	тип I	тип I
$i>2,\ \sigma_{A^-} > \sigma_B \in \Sigma_K$	тип 2	тип 3
$i>2,\ \sigma_{A^-} > \sigma_B \in \Sigma_a \setminus \Sigma_K$	тип I	тип 0
$i>2,\ \sigma_{A^-} > \sigma_B \notin \Sigma_a$	тип 0	тип 0

В первой графе с указанием типов выделен особый случай, когда слой в A^- состоит из одной переменной буквы, преобразующей произвольную часть формулы. Прочие случаи описываются второй графой. Смысл введенных типов перебора будет раскрыт ниже, в описании блока перебора по ТВ.

После росписи ТВ $i := i + 1$ и производится переход к перебору по ТВ.

(3). Перебор по ТВ

На последнем участке ТВ приступаем к перебору подформул в соответствии с типом перебора, указанным в заголовке. Перебор упорядочивается на основании информации, занесенной в массив ТВ. Для каждого выбранного варианта производится простая про-

верка МНУ. Кроме того, сравнивается длина выбранных подформул в слоя в A^-. Если в слое A^- нет переменных букв с нефиксированной длиной аналога, то длина слоя в A^- должна совпадать с суммарной длиной выбранных в В подформул. В противном случае длина слоя в A^- не менее длины подформул из В. Выбранная совокупность главных подформул слоя В, для которой перечисленные условия выполняются, фиксируется путем занесения единиц в разметочные разряды строк участка ТВ, соответствующих этим подформулам.

Опишем способ проведения перебора каждого типа.

Тип 0. Все варианты исчерпываются выбором первой главной подформулы.

Тип I. Пусть К - число главных подформул в слое А и n - число главных подформул в слое В. Перебор типа I выбирает первые К подформул, если в слое A^- нет переменных букв, изображающих произвольную часть формулы. В противном случае последовательно выбираются К, К + I, ..., n первых главных подформул.

Тип 2. Выбираются все варианты по К главных подформул. Если в слое A^- есть переменные буквы, изображающие произвольную часть формулы, выбираются все варианты по К, К + I, ..., n главных подформул.

Тип 3. Выбирается одна произвольная главная подформула.

Тип 4. Выбирается подряд по К (по К, К + I, ... при наличие в слое A^- переменной буквы, изображающей произвольную часть формулы) главных подформул, начиная с произвольной главной подформулы.

Тип 5. Если в В число главных подформул не равно К, то переход к блоку "шаг назад", иначе перебор по типу I.

Тип 6. Выбираются все главные подформулы из В.

Тип I'. Выбирается только К первых главных подформул, независимо от наличия переменных букв, изображающих произвольную часть формулы.

Разумно организовать перебор так, чтобы на некоторых этапах перебора отдельные главные подформулы исключались из дальнейшего рассмотрения. После каждого такого исключения производится так называемая проверка на остаток. Проверка на остаток заключается в простой проверке МНУ и проверке длины сразу для всех подформул, которые остаются для дальнейшего перебора.

Если проверка на остаток не выполняется либо исчерпаны все варианты, предусмотренные реализуемым типом перебора, производится переход к блоку "шаг назад".

После выбора какого-либо варианта размечаем входящие в него подформулы как i-й слой (разметка в формуле B′) и переходим к проверке МНУ.

Отметим, что упорядочение перебора, совершаемое в данном блоке, производится без чтения формул A^- или B. Тем самым достигается определенное повышение быстродействия.

(4). Проверка МНУ

Производим проверку минимально необходимых условий возможности подстановки в соответствии с определением 4.7. Если проверка дает отрицательный результат, возвращаемся к перебору по таблице вариантов, аннулировав фиксацию в B′ и ТВ последнего выбранного блоком (3) варианта. В противном случае требуем, чтобы совпадали главные связи слоев в A^- и B ($\sigma_{A^-} = \sigma_B$), если $\sigma_{A^-} \neq \infty$. При невыполнении этого требования также переходим к блоку "шаг назад". Если перечисленные требования удовлетворены, производится деление A^-.

(5). Деление A^-

В момент обращения к данному блоку в формуле B выбран и зафиксирован в B′ и ТВ слой, который предполагается аналогом слоя из A^-. Первым шагом работы блока деления A^- является выборка главной связи слоя A^-. При этом возможны следующие случаи:

1) найдена единственная главная связь;

2) главных связей несколько;

3) слой A^- состоит из одной буквы;

4) слой в A^- является выражением в скобках.

В первом случае производится деление A^- путем вписывания в позицию A^-_I, соответствующую позиции главной связи в слое A^-, разметочного символа ∇ . После этого производится переход к блоку (2).

Если главных связей в слое A^- несколько, то повидимому разумно выбирать последнюю из них для установки границы ∇ . Есть основания полагать, что перебор в этом случае будет сходиться несколько быстрее, так как сокращается число возвратов по ТВ.

Если слой в A^- состоит из одной буквы (главная связь слоя равна бесконечности), обращаемся к блоку проверки символа и фиксации переменной буквы.

Пусть далее слой в A^- является выражением в скобках. Если i -й слой в B не есть также выражение в скобках, производится переход к блоку "шаг назад". В противном случае $ГА^-$ и $ГА_I^-$ сдвигаются на позицию, а в позицию, соответствующую закрывающей скобке, в A_I^- вписывается ∇ . Скобки в B фиксируются.

ОПРЕДЕЛЕНИЕ 4.9. Будем полагать, что номер слоя i не превосходит некоторого i_{max} . Тогда всякий символ β_k (K = = I,2, ..., n) из формулы

$$B = \Delta\, \beta_1\, \beta_2 \ldots \beta_n\, \Delta$$

считается фиксированным, если в разметочной формуле

$$B' = \Delta\, \beta_1'\, \beta_2' \ldots \beta_n'\, \Delta$$

символ $\beta_k' > i_{max}$

Таким образом, фиксация некоторого символа из B производится пометкой его в B'. После фиксации скобок в B вновь обращаемся к делению A^-.

(6). Проверка символа и фиксация переменной буквы

При входе в настоящий блок установлено соответствие между буквой из A^- и слоем из B. Возможны два случая:

I) слой в A^- состоит из постоянной буквы или числа;

2) слой в A^- состоит из переменной буквы.

В случае I проверяется совпадение слоев. Если они не совпадают, переход к блоку "шаг назад". При совпадении слоев i -й слой в B фиксируется. Выделяется следующий слой в A^-. Для этого $ГА^-$ и $ГА_I^-$ сдвигаются вправо до ближайшего ∇ и затем через весь массив подряд записанных ∇ .

ТЕОРЕМА 4.3. При сдвиге $ГА^-$, $ГА_I^-$ вправо полученный в A^- слой соответствует (i – I) -у слою в B.

Доказательство. Пусть $ГА^-$ и $ГА_I^-$ переведены через единственный разделительный символ ∇ . Рассматривая процедуру расстановки ∇ (блок деления A^-), легко видеть, что ∇ установлено в позицию, соответствующую позиции главной связи некоторого слоя в A , которому был соотнесен (i – I)-й слой в B. После деления слоя A по главной связи левая подформула соотнесена

i -у слою в В. Правая подформула, следовательно, соответствует остатку (i -I)-о слоя.

Пусть $ГА^{-}$ и $ГА_{I}^{-}$ переведены через несколько подряд записанных ∇ . Очевидно, имеется несколько записанных подряд закрывающих скобок, после которых стоит либо связь σ , помеченная ∇ , либо граничный символ Δ . Последний случай не рассматривается, так как после сдвига $ГА^{-}$, $ГА_{I}^{-}$ нового слоя в A^{-} не получаем.

При установке ∇ на связь σ слой влево от σ суть выражение в скобках, соответствующее i -у слою в В, подформула вправо от σ (до ближайшего ∇ или Δ) соответствует (i - - I)-у слою. При дальнейшем делении A^{-} пара скобок зафиксировалась и деление продолжалось во внутренности скобок, которая также является i -м слоем. При этом слева получался слой, соответствующий (i + I)-у слою в В, вправо - подформула, соответствующая i -у слою в В. Продолжая рассуждение, получим, что слой (буква) перед массивом закрывающих скобок соответствует i -у слою в В. При переводе $ГА^{-}$ вправо получаем, таким образом, вправо от σ слой в A^{-}, соответствующий (i -I)-у слою в В. Теорема доказана.

ОПРЕДЕЛЕНИЕ 4.IO. Процедуру расстановки ∇ и сдвига $ГА^{-}$ и $ГА_{I}^{-}$ назовем прямым ходом $ГА^{-}$.

Если при прямом ходе $ГА^{-}$ выделить новый слой в A^{-} невозможно ($ГА^{-}$ указывает на правый граничный символ Δ), работа основного алгоритма заканчивается. Установлено соответствие между формулой A^{-} и некоторой частью из В, которая полностью зафиксирована в В'. Производится переход к блоку подстановки.

Пусть теперь слой в A^{-} состоит из переменной буквы. Проверяется соответствие выбранного i -о слоя типу переменной буквы. Если проверка не удовлетворяется, обращаемся к блоку "шаг назад". Если же проверка дает положительный результат, выбранный аналог переменной буквы (i -й слой в В) подставляется вместо данной переменной буквы во всей формуле A^{-}. При необходимости расставляются или опускаются скобки. Формула A_{I}^{-} соответственно удлиняется. Далее фиксируется i -й слой в В и производится сдвиг $ГА^{-}$, $ГА_{I}^{-}$ вправо описанным выше способом.

После подстановки A^{-} и A_{I}^{-} приобретают новый вид. Однако выборка аналога переменной буквы может оказаться неудачной. Кроме

того, в блоке подстановки может потребоваться вид аналога переменной буквы. Поэтому во все позиции A_I^-, соответствующие подставленному аналогу переменной буквы, вписывается код этой переменной буквы.

После выделения нового слоя в A^- требуем, чтобы главная связь слоя в A^- (σ_{A^-}) совпадала с главной связью слоя ($i-1$) в B, если σ_{A^-} отлична от бесконечности. При выполнении этого условия $i := i - 1$ и обращаемся к блоку деления A^-. В противном случае производится переход к блоку "шаг назад".

(7). Шаг назад

Обращение к данному блоку возможно из блоков росписи таблицы вариантов, перебора по ТВ, проверки символа и фиксации переменной буквы, деления A^-. Рассмотрим эти возможности раздельно.

(А). Пусть обращение к блоку (7) произошло из блока (3) после окончания перебора на слое.

Блок "шаг назад" производит возврат на перебор по ТВ на предыдущем слое. Возврат осуществляется путем аннулирования последнего участка ТВ и его заголовка.

После возврата по ТВ определяются связи σ^- и σ^+, наложенные на слой в A^-. Если $\sigma^- \geqslant \sigma^+$, производится переход к пункту (Г). В противном случае $i := i - 1$. Ближайший правый от $ГА_I^-$ разметочный символ ∇ в A_I^-, аннулируется.

ТЕОРЕМА 4.4. Правый от $ГА_I^-$ разметочный символ ∇ существует и в A^- ему соответствует символ (связь), отличный от закрывающей скобки.

Доказательство. Имеем $\sigma^- < \sigma^+$. Случай $\sigma^- = (, \sigma^+ =)$ исключается, так как фиксация скобок и деление внутренности скобок на слои производится без выхода из блока деления A^-. Перебор для поиска аналога всей внутренности скобок невозможен. По определению формулы невозможен случай $\sigma^- = \Delta$, $\sigma^+ =)$. Покажем, что ∇ установлен в позицию, соответствующую σ^+. По определению слоя в A^- в позиции, соответствующей σ^+, в A_I^- записан ∇ или Δ. Очевидно, что в последнем случае $\sigma^- = \Delta$ и $\sigma^- \geqslant \sigma^+$, то есть условия теоремы не выполняются. Теорема доказана.

После аннулирования ∇ вновь определяются σ^- и σ^+, наложенные на новый слой в A^-. Если $\sigma^- = \sigma^+ = \Delta$, переход

к пункту (Б). Если $\sigma^- = ($, $\sigma^+ =)$, разметка скобок аннулируется путем сдвига $ГА^-$ и $ГА^-_I$ на позицию влево и аннулирование правого ∇ .

Определяются σ^- и σ^+. Если $\sigma^- \geq \sigma^+$, производится переход к пункту (Г), в противном случае к пункту (Б).

(Б). Если слой в A^- является внутренностью пары скобок, фиксация скобок аннулируется путем сдвига $ГА^-$ и $ГА^-_I$ на позицию влево и аннулированием правого ∇ . Далее слой в В размечается символом (i – I). Производится переход к блоку перебора по ТВ.

(В). Пусть обращение к блоку (7) произошло из блока проверки символа и фиксации переменной буквы по несовпадению постоянной буквы в A^- со слоем в В, либо при несоответствии типа переменной буквы выбранному аналогу. Определяются σ^- и σ^+ – связи, наложенные на слой в A^-. Переход к пункту (Б), если $\sigma^+ > \sigma^-$ либо $\sigma^- = \sigma^+ = \Delta$. В противном случае исполняется пункт (Г).

(Г). Производится сдвиг $ГА^-$ и $ГА^-_I$ влево. При этом $ГА^-$ и $ГА^-_I$ сдвигаются влево через все подряд записанные ∇ и следующую за ними подформулу вплоть до ∇ или Δ .

Если новый слой в A^- является выбранным ранее аналогом переменной буквы и левее в A^- нет подформул, помеченных в A^-_I индексом той же переменной буквы, производится отказ от выбранного аналога заменой его на переменную букву всюду в A^-, аннулированием фиксации в A^-_I и уплотнением A^- и A^-_I.

Далее $i := i - I$. Определяются σ^- и σ^+ . Если $\sigma^- \geq \sigma^+$, сдвиг $ГА^-$ влево повторяется. Цикл оканчивается при $\sigma^- < \sigma^+$.

Каждой подформуле, через которую сдвигается $ГА^-$, соответствует слой в В, поэтому производится необходимое число возвратов по ТВ с соответствующей разметкой в В'. Переход к пункту (Б).

(Д). Пусть обращение к блоку "шаг назад" произошло при росписи ТВ.

Очевидно, если при этом $i = 2$, следует обратиться к росписи нового О-слоя. Если $i = 3$, то, как легко видеть, роспись ТВ происходит после сдвига $ГА^-$ вправо и деления A^-. При росписи ТВ не выполнились МНУ. Аннулируем результат последнего деления A^-, то есть разметочный символ ∇ и, если необходимо, расфиксируем

пару скобок. После этого обращаемся к сдвигу $ГА^-$ влево – пункт (Г).

(Е). Обращение к блоку "шаг назад" произошло при делении A^-. Слою в A^-, который является выражением в скобках, поставлен в соответствие слой в В, не заключенный в скобки. В этом случае продолжается перебор по ТВ (блок 3).

(Ж). Отметим особый случай обращения из блока проверки символа и фиксации переменной буквы. Если после сдвига $ГА^-$ вправо не выполняется условие соответствия связей в полученном слое A^- и i-м слоем в В, производится исполнение пункта (Г) для аннулирования результатов фиксации и продолжения перебора.

(8). Подстановка

В результате работы предыдущих блоков в В зафиксирован аналог формулы A^-. В самой формуле A^- переменные буквы заменились своими аналогами. Блок (8) производит подстановку в формулу В заменой зафиксированной части формулой A^+. При этом переменные буквы из A^+ заменяются выбранными аналогами. Результату подстановки присваивается номер С.

(9), (10). Выходы

После реализации блока (8) осуществляется выход "подстановка совершена".

Если все О-слои исчерпаны и аналог формулы А не найден ни в одном из них, осуществляется выход "подстановка невозможна".

Описание алгоритма закончено. Остановимся на вопросе использования оператора, реализованного на основании данного алгоритма.

Оператор ПОДСТ (как и все операторы подстановок) на выходе вырабатывает специальный признак θ (ТЭТА) в ячейке 02047. θ = 0000 0000 0000 0000, если подстановка совершена, и θ = = 0000 0000 0000 0001 в противном случае; формула С расписывается только в случае совершенной подстановки. Формулы В, A^-, A^+ не изменяются в результате работы оператора. Оператор требует, чтобы формула В была почти нормальной. В результате подстановки также получается почти нормальная формула С.

Основное ограничение на использование данного оператора накладывает его быстродействие. Если на средней длины формулах (до 100 символов) совершается до 10 подстановок в секунду при достаточно сложных правилах, то при возрастании размерности до

1000 и более символов время может исчисляться минутами и более. Особенно медленно обрабатываются формулы с большим количеством коммутативных связей. Причина этого очевидна. На отдельных этапах работы оператора может возникнуть перебор, близкий к факториальному. Быстродействие даже на формулах средней длины оказывается недостаточным для решения ряда практических задач. При решении задач часто возникает неконтролируемый рост формул, от которого трудно избавиться. Такое возрастание может привести к невозможности решения задачи из-за естественных временных ограничений.

На основании изложенного авторы рекомендуют воздержаться от использования оператора ПОДСТ без крайней необходимости. Хотя при помощи оператора ПОДСТ был решен не только ряд тестов, но и некоторые практические задачи, авторы не могут гарантировать, что он полностью отлажен. Дело в том, соответствующая программа занимает около двух листов МОЗУ и имеет весьма высокую внутреннюю связность, как это видно и из алгоритма. Повидимому, нельзя вообще гарантировать, что какая либо логическая программа такой сложности может быть отлажена до конца.

Операторы подстановки

Номер оператора	Идентификатор	Назначение	Обращение	Результат работы
0205	ПОД	Подстановка с плотным кодированием	ПОД, A^-, A^+, B, C ;	В формулу B подстановка по правилу $A^- = A^+$; результат в C. Алгоритм описан в § I гл.4.
0206	ППР	Полная подстановка по равенству	ППР, B, A, C;	В формулу B полная подстановка по равенству A; результат в C. Аналог оператора П.
0207	ПР	Подстановка по равенству	ПР, B, A, C;	C – результат подстановки в B по равенству A.
02I0	ПП	Полная подстановка	ПП, B, A^-, A^+, C;	C – результат полной подстановки в B по правилу $A^- = A^+$.
02II	ПВ	Выборочная подстановка	ПВ, B, N_α, N_ω, C;	C – результат подстановки в B по первому из правил с номерами $N_\alpha \div N_\omega$, по которому возможна подстановка.
02I2	ПСР	Подстановка по системе равенств	ПСР, B, N_α, N_ω, C;	C – результат подстановки в формулу B по правилам с номерами $N_\alpha \div N_\omega$.
02I3	ППСР	Полная подстановка по системе равенств	ППСР, B, N_α, N_ω, C;	C – результат полной подстановки в B по правилам $N_\alpha \div N_\omega$.
0220	ПРП	Простейшая подстановка	ПРП, B, A^-, A^+, C;	В формулу B везде (за один присмотр) вместо подформулы A^- вписывается A^+. Результат в C.

Таблица I5

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0223	ПМ	Подстановка в массив	ПМ, B, N_α, N_ω, C;	В каждую формулу массива, прописанного под номером B, совершается полная подстановка по системе правил $N_\alpha \div N_\omega$. Результат – массив формул C.
0224	ПО	Операторная подстановка	ПО, N, A, N_1, N_2;	Под номером N_2 прописывается результат полной подстановки в формулу N_1 по массиву правил, расписанному с ячейки A формулы N. Признаком конца массива служит тетрада 7777.
0227	ППП	Полная простейшая подстановка	ППП, B, A^-, A^+, C;	В формулу B всюду (циклический просмотр) вместо подформулы A^- вписывается подформула A^+. Результат – формула C.
0236	П	Подстановка с поячеечным кодированием	П, B, A, C;	В формулу B полная подстановка по массиву правил A с результатом в C.
0242	ПММ	Подстановка в массив по массиву	ПММ, B, A, C;	Массив C – результат полной подстановки в массив B по массиву правил A.
0523	ПСС	Подстановка системы в систему	ПСС, M_α, K, K, N_α, n;	Полная подстановка в $N_\alpha \div N_{\alpha+n-1}$ по системе $M_\alpha \div M_{\alpha+K-1}$ (правила $M_\alpha \div M_{\alpha+K-1}$ записаны через знак равенства "=" с кодом 0011.

Таблица 15

Операторы подстановки

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0177	ПОДСТ	Обобщенная коммутативная подстановка	ПОДСТ, A^-, A^+, B, C;	Под номером C прописывается результат обобщенной коммутативной подстановки в формулу B по правилу $A^- = A^+$.

Таблица I5

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0230	РОМ	Роспись матриц	РОМ, ТИП, N, A_m, A_n, A_x, x, R, w;	Из массива соотношений N, записанного в виде $A\bar{x} = B$, либо $A\bar{x} - B$ расписывается матрица коэффициентов: ТИП = 0001 – матрица без свободных членов; ТИП = 0002 – матрица со свободными членами, перенесенными вправо; ТИП = 0003 – матрица со свободными членами, перенесенными влево; ТИП = 0004 – матрица для решения системы уравнений. w – номер, под которым прописывается вектор неизвестных, m (число строк) засылается в A_m, n (число столбцов) засылается в A_n, $0 \Rightarrow A_x$, если система однородна, иначе $1 \Rightarrow A_x$.
0232	ПАЛ	Программы линейной алгебры	ПАЛ, ТИП, m, n, N, K;	Оператор может решать системы линейных алгебраических уравнений, вычислять определители, обращать матрицы, выделять независимые подсистемы, обращать в нуль определители заданного порядка.
0505	В ПОДС	Выделение подсистемы	ВПОДС, ТИП, M, n, x, u, N, π;	Из уравнений $M \div M+n-1$ выделяется независимая подсистема и прописывается номерами $M \div M+[N]-1$

Таблица I5

Операторы линейной алгебры

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0506	РАНГ	Вычисление ранга	РАНГ, ТИП, M, n, x, u, ρ;	В зависимости от значения тетрады ТИП вычисляется ранг матрицы коэффициентов (основной или расширенной) при заданных в обращении буквах.
0507	СИСТ	Решение системы	СИСТ, ТИП, M, n, x, N, К;	ТИП = 0000 – система уравнений $M \div M+n-1$ разрешается относительно x, решения прописываются номерами $N \div N+[\kappa]-1$. ТИП = 0001 – система разрешается относительно dx.
0513	НО	Обнуление определителей	НО, ТИП, M, n, ρ, N, κ, x;	Под номерами $N \div N+[\kappa]-1$ расписывается система соотношений, обращающих в нуль все определители порядка ρ. ТИП = 0000 – в матрице коэффициентов при x; ТИП = 0001 – в матрице коэффициентов при x и свободных членов.
0240	АРОМ	АнтиРОМ	АРОМ, M, T, m, n, W, l_{min}, j_{min}, j_{max}, P, R;	В зависимости от Т совершает преобразование исходных формул, обратные оператору РОМ.

Таблица 15

Операторы преобразования формул

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0201	АРИФМ	Арифметик с плотным кодированием	АРИФМ, N ;	Осуществляет все операции над числами в пределах заданной в теле оператора библиотеки подпрограмм в формуле N .
0202	ВП	Вписывание	ВП, A^-, A^+, B, C;	В формулу B вместо произвольной последовательности символов, прописанной под номером A^-, вписывается последовательность символов, прописанных под номером A^+. Результату присваивается номер C.
0203	КУ	Коммутативное упорядочение	КУ, N ;	В формуле N устраняются все инверсии.
0216	ПРИП	Приведение подобных	ПРИП, N ;	В формуле N раскрываются скобки, производятся все операции над числами, приводятся подобные в пределах списка правил, заданных в теле оператора.
0217	ДИФ	Дифференцирование	ДИФ, M, N ;	В массиве формул, прописанных под номером M, вычисляются все частные производные при помощи системы правил, заданных в теле оператора. Результату присваивается номер N .

Таблица I5

Операторы преобразования формул

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0221	РОСПВ	Роспись пустых вхождений	РОСПВ, N ;	В формулу N на место пустых вхождений вписывается символ 1000.
0225	ОКРН	Округление до нуля	ОКРН, N , М, A_ε;	В формуле N заменяются нулем все вещественные числа, не превосходящие по модулю число ε из ячейки с относительным адресом A_ε формулы с номером М. Если М = 0000, A_ε – адрес в нулевом кубе МОЗУ.
0226	ВПП	Полное вписывание	ВПП, В, A^-, A^+, С;	В формулу В всюду (циклическим просмотром) вместо последовательности символов A^- вписывается последовательность A^+. Результат прописывается под номером С.

Таблица 15

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0237	КУПОР	Коммутативное упорядочение	КУПОР;	Устраняет инверсии в формуле, расписанной по символу в ячейку, относительно связей + и х. Информация о расположении формулы выбирается из тела оператора 0236.
024I	А	Арифметик с поячеечной записью	А, N ;	В формуле N производятся операции +, −, х, /, ↑ над числами.
0300	ИНТ	Интегрирование	ИНТ, ТИП, М, A_1, A_2, A_3, N_α , N ;	В заданной формуле вычисляются все первообразные. Возможна подстановка пределов интегрирования.
030I	АППР	Аппроксимация	АППР, A_1, A_2, A_3, N_1; N_2, N_α , К;	Заданное выражение аппроксимируется полиномом Чебышева первого рода с заданным числом членов.

Таблица I5

Операторы групового преобразования формул и вспомогательные операторы

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0200		Операции над формулами	0200 M ℓ 0000 ИР КОП A_1 ..... ИР КОП $A_{2\ell}$	По информационным строкам ИР КОП A_i, $i = 1, \ldots, 2\ell$, компонуется формула M. Использование оператора возможно только в режиме ручного программирования.
0231	ОФ	Объединение формул	ОФ, ТИП, N_α, N_ω, R;	Формулы с номерами $N_\alpha \div N_\omega$ объединяются в массив и прописываются под номером R. Если ТИП = 0000, то формулы $N_\alpha \div N_\omega$ аннулируются.
0233	ПЧ	Печать формул	ПЧ, ТИП, N;	В соответствии с типом может быть осуществлена печать одной формулы, массива формул, заданного числа формул с последовательными номерами, всех формул, прописанных на РП, а также заданного списка формул.
0511	РАЗДВ	Раздвоение	РАЗДВ, M, n, N;	Формулы $M \div M+n-1$ раздваиваются и прописываются также номерами $N \div N+n-1$
0512	АННФ	Аннулирование последовательности формул	АННФ, N_α, N_ω;	Формулы $N_\alpha \div N_\omega$ аннулируются.
0515	ПОП	Печать оператора	ПОП, N;	Оператор с номером N распечатывается в командах.
0516	ПЯЧ	Печать ячеек	ПЯЧ, N_α, N_ω;	Ячейки $N_\alpha \div N_\omega$ распечатываются как константы.

Таблица I5

Операторы группового преобразования формул и вспомогательные операторы

Номер оператора	Идентификатор	Название	Обращение	Результат работы
0520	ПНУМ	Перенумерация	ПНУМ, M_α ; M_ω, N_α ;	Формулам $M_\alpha \div M_\omega$ присваиваются номера $N_\alpha \div N_\omega$. Номера не должны пересекаться.
0521	ДЕМ	Деление массива	ДЕМ, М, N , К;	Формулы из массива формул М прописываются также поотдельности номерами $N \div N+[k]-1$. Содержимое ячейки К формируется оператором.
0525		Занесение поправок	См.описание	Используется только в режиме ручного программирования. Позволяет организовать достаточно простое поячеечное занесение поправок в операторы и введенные ранее формулы.
0526	ПЕЧАТЬ	Печать последовательности формул	ПЕЧАТЬ, М, n ;	Распечатывает формулы С номерами $M \div M+n-1$

Таблица I5

Операторы приложений Авто-Аналитика

Номер	Идентификатор	Название	Обращение	Результат работы
0577	КАРТАН	Реализация алгоритма Картана	КАРТАН, РЕЖИМ, N, n, k, s, N_1;	Данный оператор выполняет компилирующие функции по отношению к пакету операторов, реализующих метод внешних форм Картана.
0500	ЗАМ	Замыкание	ЗАМ;	Под номерами $N+s \div N+R-1$ расписываются уравнения, замыкающие исходную систему. Аргументы выбираются из ячеек 04074 (N); 04075 (n), 04076 (R), 04077 (s).
0501	ВНДИФ	Внешнее дифференцирование	ВНДИФ, М, N, m;	Вычисляется внешний дифференциал формул $M \div M+m-1$. Результат прописывается под номерами $N \div N+m-1$
0502	РАСЩ	Расщепление	РАСЩ, N_α, N_ω, N;	В квадратичные формы $N_\alpha \div N_\alpha+s-1$ подставляются равенства $M_\alpha \div M_\alpha+R-1$. Коэффициенты при мономах в полученных выражениях прописываются номерами $N \div N+k-1$.
0503	ЦЕПЬ	Построение цепи интегральных элементов	ЦЕПЬ, N_α, M_α, N, Q;	$N_\alpha \div N_\alpha+s-1$ – квадратичные формы, $M_\alpha \div M_\alpha+R-1$ – значения дифференциалов, N – начало массива рабочих номеров формул, с ячейки Q расписываются число Картана и характеры цепи.

Таблица I5

Номер	Идентификатор	Название	Обращение	Результат работы
0504	УПЛОТ	Уплотнение	УПЛОТ, M, n, N;	В формулах $M \div M+n-1$ приводятся подобные, тождественные нули отбрасываются, номера уплотняются. Число оставшихся формул засылается в ячейку N.
0514	ПЗАВ	Проверка на зависимость	ПЗАВ, ТИП, M, n, x_1, x_2, x_3;	В зависимости от значения ТИПа, проверяется вхождение букв x_1, x_1, или x_2, x_1, x_2, или x_3 в формулы $M \div M+n-1$.
0517	КВЫХ	Оператор выхода	КВЫХ, ТИП;	ТИП равен номеру выхода из алгоритма Картана. Выводится на печать вся информация, необходимая математику.
0522	ВНЕС	Выбор независимых соотношений	ВНЕС, M, n, x, N, K_1, K_2;	Из формул $M \div M+n-1$ выделяются независящие от x и прописываются номерами $N \div N + [k_2]-1$. Оставшиеся прописываются номерами $M \div M+[k_1]-1$.
0524	ЗПЕР	Замена переменных	ЗПЕР, $R1$;	В системе $N \div N+R-1$ (N в ячейке 04074, R в ячейке 04076) p^{ω} заменяются новыми неизвестными функциями u^{R+1}, ..., u^{R1}. Равенства, по которым производится замена, выдаются на печать.

Таблица I5

Номер	Идентификатор	Название	Обращение	Результат работы
0510	ДКС	Дифференцирование конечных соотношений	ДКС, М, n, К;	Дифференцируются конечные соотношения $M+M+n-1$ ..., уплотняются и прописываются номерами $M \div M+[k]-1$.
0540	ВМН	Вынесение общего множителя	ВМН, M_α, M_ω, N, R;	В формулах $M_\alpha \div M_\omega$ выносится за скобку множитель, прописанный под номером N. В формуле N допускаются переменные буквы. R, $R-1$ – – рабочие формулы.
0541	ВКОЭФ	Выведение коэффициентов	ВКОЭФ, M_α, M_ω, N, К, R, L;	Из формул $M_\alpha \div M_\omega$ выводятся коэффициенты при N (заданная формула) и прописываются поотдельности номерами К, К + I, ... Число полученных формул засылается в ячейку R.
0542	ПДКФ	Подстановка дифференциалов в квадратичные формы	ПДКФ, N_α, n, M_α, m;	Подставляются дифференциалы в квадратичные формы и раскрываются скобки относительно внешнего произведения.
0543	ВММ	Вынесение общего множителя в массиве формул	ВММ, М, N;	В массиве формул М выносится за скобки формула, прописанная под номером N.
0544	ИНД	Упорядочение индексов	ИНД, M_α, M_ω;	В формулах $M_\alpha \div M_\omega$ упорядочиваются индексы в мономах.

Таблица I5

Операторы приложений Авто-Аналитика

Номер	Идентификатор	Название	Обращение	Результат работы
0600	МОДЕЛЬ	Моделирование	МОДЕЛЬ, ТИП, α, М, N ;	Строится математическая модель радиоэлектронной схемы (т.е. система уравнений).
0602	СЕЛЕКТ	Селектор	СЕЛЕКТ, ТИП, М, M_1, M_2, N, x, Π ;	Оператор расщепляет исходный массив формул М в массивы M_1 и M_2. Если ТИП = 0000, то в M_2 собираются линейные алгебраические относительно x уравнения, прочие в M_1. Если ТИП = 0001, в M_2 собираются дифференциальные уравнения, прочие в M_1.
0302	ОПТИМ	Оптимизатор	ОПТИМ, Т, А, F, n, x, ρ, К, Δ ;	Минимизирует заданную функцию методом покоординатного спуска с дробным шагом.
0303	ДИФСФ	Дифференцирование сложных функций	ДИФСФ, N, М, R ;	Дифференцируется сложная функция, прописанная под номером N, результат прописывается под номером R. Массив аргументов функции прописан под номером М.

Таблица I5

Оператор ПОД (0205). Подстановка с плотным кодированием

Обращение к данному оператору имеет вид

ПОД, A^- , A^+, B, C;

Оператор осуществляет обобщенную коммутативную подстановку в формулу B по правилу $A^- = A^+$ с некоторыми ограничениями общности операции. Суть этих ограничений легко понять из описания алгоритма работы оператора, приведенного ниже.

Основными этапами работы алгоритма являются устранение инверсий в исходной формуле B, просмотр формул A^- и B с выбором подформулы в B, являющейся аналогом A^-. Выбранный аналог прописывается под рабочим номером, строится и прописывается аналог правой части правила A^+ и совершается простейшая строчная подстановка по правилу $A^- = A^+$.

Рассмотрим особенности работы алгоритма. Как уже указывалось, аналогом A^- может быть только подформула в B. Поиск этой подформулы осуществляется одновременным просмотром A^- и B. При этом в качестве аналогов переменных букв из Φ_0 выбираются минимальные подформулы, удовлетворяющие условию совпадения наложенных связей. Алгоритм использует рабочие номера формул с 0700 по 0731, накладывая тем самым определенные ограничения на использование оператора ПОД. Под номерами 0700 ÷ 0727 прописываются аналоги переменных букв, которые по окончанию работы оператора аннулируются. Под номерами 0730 и 0731 прописываются формулы A^- и A^+ с подставленными значениями переменных букв. Эти формулы оператор не аннулирует, так как они иногда требуются при дальнейших расчетах. По сравнению с оператором ПОДСТ данный оператор осуществляет квадратичный перебор вместо факториального. Это обстоятельство и послужило главной причиной разработки оператора ПОД. Наряду с оператором П оператор ПОД составляет основу группы операторов подстановки.

При работе с оператором ПОД пользователь должен учитывать, что далеко не все подстановки, доступные оператору ПОДСТ, реализует описываемый оператор. Часто требуется предварительное преобразование формул. Примерами такой предварительной обработки формул являются некоторые описанные ниже операторы. Недоста-

точным иногда бывает и быстродействие оператора ПОД вследствие больших затрат времени на чтение формул A^- и В при помощи соответствующих стандартных программ, прописку и аннулирование аналогов переменных букв, устранение инверсий в формуле В. Недостаточное быстродействие сказывается обычно при решении задач, требующих многократного преобразования формул по большим массивам правил. К таким задачам относятся задачи линейной алгебры, требующие многократного приведения подобных, реализуемого подстановкой по массиву правил.

Оператор ШПР (0206). Полная подстановка по равенству

Обращение к оператору имеет вид

ШПР, В, А, С;

В формулу В совершается полная подстановка по равенству А. Результат прописывается под номером С. Поясним понятие равенства. Часто оказывается удобным прописывать правило не двумя формулами, соответствующими левой и правой части (A^- и A^+), а одной формулой А. Для такой записи введена специальная связь "-" (тире) с кодом 0007, которая именуется равенством в правилах или просто равенством. Следует отличать равенство в правилах от обычного равенства "=" с кодом 0011. Часть формулы А от левого граничного символа до равенства в правилах считается левой частью правила, оставшаяся часть формулы А считается правой частью правила. Во избежание путаницы, не рекомендуется использовать связь "-" вне правил. Таким образом, правило А имеет вид $A^- - A^+$. Операция полной подстановки понимается как последовательное применение операции однократной подстановки, пока она возможна.

Отметим, что данный оператор, как и все операторы подстановки вообще, вырабатывает логический признак Θ (ТЭТА) в ячейки 02047 равный нулю, если хотя бы одна подстановка была совершена, и равный единице в противном случае. При описании последующих операторов специально об этом факте мы упоминать не будем.

С точки зрения пользователя работа оператора ШПР аналогична работе оператора П (см. ниже).

Оператор ПР (0207). Подстановка по равенству

Обращение к оператору имеет вид

ПР, В, А, С;

В формулу В совершается однократная подстановка по равенству А, результат прописывается под номером С. В качестве вспомогательного используется оператор ПОД. Отметим, что формула С организуется лишь в случае совершенной подстановки (как и в операторе ПОД).

Оператор ПП (0210). Полная подстановка

Обращение к оператору имеет вид

ПП, В, A^-, A^+, С;

В формулу В осуществляется полная подстановка по правилу $A^- = A^+$ (левая и правая часть прописаны отдельно), результат в С. В качестве вспомогательного используется оператор ПОД.

Оператор ПВ (0211). Выборочная подстановка

Обращение к оператору имеет вид

ПВ, В, N_α, N_ω, С;

Здесь В - номер обрабатываемой формулы, $N_\alpha, N_\alpha+1, \ldots, N_\omega$ - номера правил, записанных через равенство, С - результат подстановки.

Оператор совершает подстановку в формулу В по правилу N_α, если подстановка невозможна - по правилу $N_\alpha+1$ и так далее. Как только одно из правил применилось, оператор работу заканчивает, $\theta := 0$. Если весь список правил просмотрен, оператор работу заканчивает, $\theta := 1$.

В качестве вспомогательного используется оператор ПР.

Оператор ПСР (0212). Подстановка по системе равенств

Обращение к оператору имеет вид

ПСР, В, N_α, N_ω, С;

Здесь В - номер обрабатываемой формулы, $N_\alpha, N_\alpha+1, \ldots, N_\omega$ - номера правил, С - номер результата. Оператор осуществляет однократную подстановку в В по правилам $N_\alpha \div N_\omega$. В ка-

честве вспомогательного используется оператор ШР.

Если ни одной подстановки не было, формула С совпадает с формулой В.

Оператор ШСР (0213). Полная подстановка по системе равенств

Обращение к оператору имеет вид

ШСР, В, N_α, N_ω, С;

Здесь С – результат полной подстановки в формулу В по правилам $N_\alpha \div N_\omega$. В качестве вспомогательного используется оператор ПСР.

Оператор ПРП (0220). Простейшая подстановка

Обращение к оператору имеет вид

ПРП, В, A^-, A^+, С;

Оператор реализует подстановку вместо подформулы из В, совпадающей с A^-, формулы A^+. Результат прописывается под номером С. Для поиска вхождений A^- в формулу В осуществляется однократный просмотр всей формулы В. Результат подстановки приводится к почти нормальному виду. Если ни одна подстановка не была совершена, формула С совпадает с почти нормальным видом формулы В. При неправильной расстановке скобок в формуле В печатается соответствующий диагностический текст и работа системы прерывается.

Оператор ПМ (0223). Подстановка в массив

Обращение к оператору имеет вид

ПМ, В, N_α, N_ω, С;

В каждую формулу массива, прописанного под номером В, совершается полная подстановка по правилам $N_\alpha \div N_\omega$. Результат прописывается под номером С. Формулы из массива В, в которые не было подстановок, переписываются в массив С без изменений. Если не было ни одной подстановки, массив С совпадает с массивом В.

В качестве вспомогательного используется оператор ПОД.

Предполагается, что номер В не совпадает с номером С.

Оператор ПО (0224). Операторная подстановка

Обращение к оператору имеет вид

ПО, N, А, N_1, N_2;

В массив формул N_1 совершается полная подстановка по массиву правил, записанных в виде равенств в теле формулы N с ячейки А (А – адрес относительно начала формулы N). Массив правил начинается с ячейки А и заканчивается формулой, после которой стоит тетрада 7777. Тетрада 7777 может находиться в любой позиции ячейки, следующей за ячейкой с последним отличным от граничного символом формулы массива. Результат подстановки прописывается под номером N_2.

Оператор ПО обычно используется при построении основных операторов системы (отсюда и название) и позволяет размещать правила в теле программы, чем достигается экономия номеров формул. В качестве вспомогательного используется оператор ПММ.

Оператор ПШ (0227). Полная простейшая подстановка

Обращение к оператору имеет вид

ПШ, В, A^-, A^+, С;

В формулу В всюду (при помощи циклического просмотра) вместо формулы A^- вписывается формула A^+. Результат прописывается под номером С. В качестве вспомогательного используется оператор ПРП. Формулы В, A^-, A^+ сохраняются.

Оператор П (0236). Подстановка с поячеечным кодированием

Обращение к оператору имеет вид

П, В, А, С;

В формулу В совершается полная подстановка по массиву правил А. Результат прописывается под номером С. Если подстановок не было, формула С совпадает с исходной формулой В.

По своим функциям оператор П совпадает с оператором ПОД.

Разработка оператора П вызвана необходимостью иметь в системе оператор, имеющий значительно более высокое быстродействие

по сравнению с оператором ПОД. Основным средством повышения быстродействия является переход от плотной записи обрабатываемых формул, принятой в Авто-Аналитике (по четыре символа в ячейку), к поячеечной (по одному символу в ячейку). Оператор П переводит формулы В, A^- и A^+ из плотной записи в поячеечную, совершает полную подстановку, результирующая формула С вновь прописывается в плотном виде. Отказ от плотной записи и некоторые усовершенствования алгоритма подняли быстродействие оператора П примерно в двадцать раз по сравнению с оператором ПОД. Естественной расплатой за повышение быстродействия явилось соответствующее сокращение (в четыре раза) максимально допустимой длины обрабатываемых формул. При решении больших задач это сокращение оказывается существенным.

В качестве рабочих для поячеечной росписи формул оператор П использует ячейки АОРП – Δ + АОРП – I. Здесь АОРП – начало рабочего поля, задаваемое в числе параметров системы. Величина Δ задается восьмеричной константой в ячейке 0006 оператора П. Стандартное ее значение равно 10000. При необходимости пользователь может варьировать величиной Δ. Значение АОРП необходимо задавать с учетом того, что Δ ячеек перед рабочим полем могут быть испорчены оператором П (а также оператором Арифметик). Максимальная длина обрабатываемых формул равняется приблизительно $\Delta/2$. В случае нехватки рабочих ячеек (малого Δ) оператор П работу системы прерывает запрещенной командой.

Так же, как оператор ПОД, данный оператор не является полной реализацией операции обобщенной коммутативной подстановки, что необходимо учитывать при его использовании.

Оператор ПММ (0242). Подстановка в массив по массиву

Обращение к оператору имеет вид

ПММ, В, А, С;

В массив формул В совершается полная подстановка по массиву правил А. Результатом подстановки является массив формул С. В качестве вспомогательного используется оператор П.

Оператор ПСС (0523). Подстановка системы в систему

Обращение к оператору имеет вид

ПСС, M_α, К, N_α, n ;

В формулы $N_\alpha \div N_\alpha + n-1$ осуществляется полная подстановка по системе правил с номерами $M_\alpha \div M_\alpha + k-1$. Правила $M_\alpha \div M_\alpha + k-1$ записываются с помощью обычного знака равенства "=" с кодом 0011. В качестве вспомогательного используется оператор ПОД.

§ 2. ОПЕРАТОРЫ ЛИНЕЙНОЙ АЛГЕБРЫ

Оператор РОМ (0230). Роспись матриц

Обращение к оператору имеет вид

РОМ, ТИП, N , A_m , A_n , $A_{\bar{x}}$, x , R , w ;

Оператор РОМ является вспомогательным в пакете операторов решения задач линейной алгебры. Основные программы Авто-Аналитика предполагают строчную запись уравнений вместе с неизвестными буквами. При непосредственном решении задач линейной алгебры более удобной оказывается матричная запись, включающая в виде отдельных формул только коэффициенты заданной системы. Переход от строчной записи к матричной осуществляется при помощи оператора РОМ.

Исходная система прописана в виде массива формул $A\bar{x} = B$ или $A\bar{x} - B$ под номером N. Неизвестная буква x указывается в обращении. При этом предполагается, что буква x может иметь один (верхний) или два (верхний и нижний) индексы, то есть может быть записана в виде x^{-i} либо в виде x_{-i}^{-j} . Связи надчерк и подчерк являются соответственно признаками верхнего и нижнего индексы, индексы i и j задаются целыми числами (кодируются одной тетрадой).

Оператор РОМ формирует и прописывает под номером R матрицу коэффициентов при заданной букве x . Способ формирования указывается тетрадой ТИП. Перечислим эти способы.

ТИП = 0001. Расписывается матрица без правых частей уравнений (основная матрица).

Нулевые коэффициенты (которых естественно нет в строчной записи) заносятся оператором РОМ в результирующую матрицу.

ТИП = 0002. Расписывается расширенная матрица коэффициентов. Предполагается, что исходная матрица прописана в виде $A\bar{x} = B$.

ТИП = 0003. Расписывается расширенная матрица коэффициентов. Предполагается, что исходная матрица прописана в виде $A\bar{x} - B$.

ТИП = 0004. Расписывается матрица для решения системы линейных алгебраических уравнений.

Одновременно с росписью матрицы формируется и прописывается под номером w вектор неизвестных (каждая неизвестная буква со своим индексом в виде отдельной формулы массива). Если w = = 0000, то этот вектор не формируется. Необходимость росписи вектора w обычно связана с тем, что в исходной системе неизвестные буквы с некоторыми комбинациями могут отсутствовать. Предполагается, что в этом случае основная матрица является невырожденной. Для восстановления исходного вида системы используется вектор w.

Результирующая матрица расписывается в виде массива формул построчно.

Если указан ТИП = 001 i (i = 1,2,3,4), то в ячейки с относительным адресом A_m, A_n, A_π оператора, из которого происходило обращение к РОМу, засылаются в виде восьмеричных условных чисел величины

m – число строк в результирующей матрице;

n – число столбцов в результирующей матрице;

π – признак однородности системы, равный нулю, если система однородна и единице в противном случае.

Если указан ТИП = 000 i (= 1,2,3,4) эти величины засылаются в ячейки с адресами A_m, A_n, A_π оперативного запоминающего устройства.

Отметим, что время работы оператора РОМ иногда оказывается довольно большим (в зависимости от размерности и вида системы), так как в исходной системе N приводятся подобные и группируются коэффициенты при неизвестных буквах.

Оператор ПАЛ (0232). Программы линейной алгебры

Обращение к данному оператору имеет вид

ПАЛ, ТИП, m , n , N , К;

В зависимости от тетрады ТИП оператор ПАЛ может решать следующие задачи.

I. Вычисление определителей.

2. Решение систем линейных алгебраических уравнений.

3. Выделение независимых подсистем векторов.

4. Определение рангов матриц.

5. Приравнивание всех определителей заданного порядка к нулю.

6. Обращение матриц.

Опишем кратко работу данного оператора. Пусть задана матрица $A = (a_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. Основной цикл преобразований для решения указанной группы задач заключается в приведении исходной матрицы к треугольному виду. В качестве метода приведения выбран модифицированный метод исключения, названный методом эквивалентной матрицы. В связи с тем, что время работы ЭЦВМ растет пропорционально длине обрабатываемых формул, желательно, чтобы выражения в процессе вычислений имели как можно меньшую длину. В настоящей методике учтена также возможность (довольно характерная для практических задач) разреженности исходной матрицы (наличия большого числа нулевых компонентов). Так как коэффициенты матрицы могут иметь и символический и численный характер, требуется помнить о возможной потери точности при численных преобразованиях.

Перечисленные выше требования в какой-то мере учтены в предлагаемом методе. Метод заключается в следующем.

Для фиксации расположения компонентов матрицы на рабочем поле РП и проведения преобразований над ними расписывается специальная матрица, именуемая в дальнейшем эквивалентной матрицей ЭМ. Исходную матрицу будем называть основной матрицей ОМ. В ЭМ заносится следующая информация о каждом элементе ОМ: номер строки и номер столбца, которым принадлежит элемент; длина этого элемента; адрес ячейки, с которой начинается запись элемента на РП. Под длиной элемента понимается число позиций, занятых элементом на РП. Длина нулевого элемента считается равной нулю.

Основная матрица может находиться в любом месте РП. Количество ячеек, занятых ОМ, меняется в ходе преобразований. Эквивалентная матрица занимает ($m \times n$) ячеек и расположена в ячей-

ках, относящихся к таблице переходов. Программа диспетчер корректирует адреса в таблице переходов, поэтому адреса, занесенные в ЭМ, всегда будут действительными адресами ячеек, с которых начинаются элементы ОМ.

После росписи матриц ОМ и ЭМ выполняется следующая рекуррентная процедура.

1-й шаг. Выбирается оптимальный элемент так, чтобы он давал наименьшее удлинение матрицы при дальнейших преобразованиях. Строка, содержащая оптимальный элемент, делится на него. Оптимальный элемент выбирается фактически в ЭМ, так как в ней имеются длины всех элементов основной матрицы. Это исключает необходимость реализации длительного процесса поиска и чтения элементов ОМ и РП.

2-й шаг. Производится перестановка строки и столбца, содержащих оптимальный элемент, соответственно с первой строкой и первым столбцом. Эта перестановка также производится в ЭМ, так как в ней имеется информация о расположении любого элемента ОМ.

3-й шаг. Из каждого элемента, кроме принадлежащих первой строке и первому столбцу, вычитается произведение элементов, найденных при помощи перпендикуляров, проведенных от данного элемента к первой строке и первому столбцу. Данные об элементах, участвующих в преобразовании, выбираются из ЭМ.

4-й шаг. Матрица умножается на $(-1)^{i+j}$, где i - номер строки, j - номер столбца, которым принадлежит оптимальный элемент.

При повторном обращении к описанной процедуре первая строка и первый столбец исключаются из рассмотрения. Процесс вычислений заканчивается после приведения матрицы к треугольному виду.

Из приведенного описания видно, что наличие ЭМ значительно упрощает реализацию алгоритма на вычислительной машине, так как отпадает необходимость слежения за расположением элементов ОМ на РП. Это особенно существенно в связи с тем, что элементы ОМ имеют переменную длину. Наличие в ЭМ номеров строк и столбцов используется при решении каждой из поставленных задач.

Рассмотрим вопрос о выборе оптимального элемента. При решении этой задачи с числовыми матрицами обычно выбирается наи-

больший по абсолютной величине коэффициент (например, в методе Гаусса). Для разрешенных матриц целесообразно выбирать в качестве оптимального такой элемент, который минимизировал бы число операций при дальнейших преобразованиях. Будем выбирать оптимальный элемент в зависимости от количества нулей в соответствующих ему строке и столбце. Выбрать этот элемент, оптимальный для всей матрицы, практически невозможно, так как время решения соответствующей экстремальной задачи будет повидимому значительно превышать время решения исходной задачи по самому неэффективному алгоритму, даже для задач сравнительно малых размерностей.

Если в матрице имеются аналитические выражения, оптимальный элемент будем выбирать по длине. Время работы машины зависит в основном от длины обрабатываемых формул, поэтому необходимо минимизировать удлинение выражений в ходе преобразований ОМ. Этим достигается также наибольшая эффективность использования рабочего поля.

Пусть оптимальный элемент имеет длину ℓ . Тогда при делении строки на этот элемент и преобразовании матрицы на третьем шаге алгоритма рост матрицы можно оценить сверху выражением

$$\ell(m-1)(n-2)+L_1(m-1)+L_2(n-1),$$

где L_1 и L_2 – соответственно суммы длин элементов строки и столбца, содержащих оптимальный элемент. Способ выбора оптимального элемента на основании приведенного выражения учитывает и разреженность матрицы, так как длину нулевых элементов мы положили равной нулю.

Подозрительный на оптимальность числовой элемент, отношение которого к максимальному числовому элементу в ОМ не превосходит по абсолютной величине заранее заданного положительного числа ε , исключается из рассмотрения. Этот прием позволяет обойти потерю точности при численных преобразованиях.

Рассмотрим некоторые возможности программы, реализующей данный алгоритм. В предположении, что каждый элемент матрицы занимает одну ячейку, данная программа позволяет работать с матрицами порядка (70 x 70). С увеличением длин элементов матрицы допустимая размерность задач соответственно уменьшается.

Уменьшение допустимых размерностей обрабатываемых матриц связано также с возрастанием количества различных символов, из которых состоят элементы матрицы. В этом случае при работе программы приведение подобных длина выражений не уменьшается и на определенном этапе работы программы может произойти прерывание работы системы из-за нехватки рабочего поля.

Приведем еще раз обращение к оператору, которое имеет вид

ПАЛ, ТИП, m , n , N , К;

Здесь N - номер, под которым прописана обрабатываемая матрица, m - число строк матрицы, n - число столбцов матрицы, К - порядок определителей (отлично от нуля только для ТИП = 0620).

Если ТИП = 0630, выделяется независимая подсистема векторов, заданных строками исходной матрицы.

Если ТИП = 0640, решается система линейных алгебраических уравнений, коэффициенты которой заданы исходной матрицей.

Если ТИП = 0650, вычисляется определитель заданной матрицы.

Если ТИП = 0660, вычисляется ранг заданной матрицы в виде условного восьмеричного числа в ячейке 01224.

Если ТИП = 1250, ищется матрица, обратная к исходной.

Если ТИП = 0620, выписывается система соотношений, выполнение которых обращает в нуль все определители заданного порядка К в исходной матрице.

Результат прописывается под номером 0036 (ТИП = 0620 и ТИП = 0630) и под номером 0037 в прочих случаях в виде формулы или массива формул.

Работа системы может быть прервана оператором ПАЛ, если исходная матрица не удовлетворяет обычным требованиям невырожденности, квадратности и т.д. (в зависимости от конкретной решаемой задачи.

Оператор ВПОДС (0505). Выделение подсистемы

Обращение к оператору имеет вид

ВПОДС, ТИП, М, n , x , u , N , π ;

Из уравнений, прописанных поотдельности номерами $M \div M+n-1$, выделяется независимая подсистема уравнений и прописывается под номерами $M \div M+[N]-1$. Число уравнений в независимой

подсистеме засылается восьмеричным условным числом в ячейку N ($[N]$ означает содержимое ячейки N). В ячейку $\mathcal{T}$ засылается нуль, если система уравнений однородна, в противном случае в ячейку $\mathcal{T}$ засылается единица.

Если ТИП = 0000, выделяется подсистема коэффициентов при букве x (с верхним индексом) и свободных членов.

Если ТИП = 0001, выделяется подсистема коэффициентов при dx ($'\Delta' x$) и свободных членов.

Если ТИП = 0002, выделяется подсистема коэффициентов при du, dx ($'\Delta' U$, $'\Delta' x$) и свободных членов.

Буквы x и U задаются в обращении в виде констант. Данный оператор, как и ряд последующих является вспомогательным, комплексирующим работу операторов РОМ, ПАЛ, АРОМ.

Оператор РАНГ (0506). Вычисление ранга

Обращение к оператору имеет вид

РАНГ, ТИП, М, n , x , u , ρ ;

ТИП = 0001 – в ячейку ρ вычисляется ранг матрицы коэффициентов при dx ($'\Delta' x$) в системе уравнений $M \div M+n-1$.

ТИП = 0002 – в ячейку ρ вычисляется ранг матрицы коэффициентов при du и dx ($'\Delta' U$, $'\Delta' x$) .

ТИП = 0003 – ранг матрицы коэффициентов при x .

ТИП = 0004 – ранг матрицы коэффициентов при x и свободных членов.

Оператор СИСТ (0507). Решение системы

Обращение к оператору имеет вид

СИСТ, ТИП, М, n , x , N , К;

ТИП = 0000 – система уравнений $M \div M+n-1$ разрешается относительно x , решения прописываются номерами $N \div N+[\kappa]-1$.

ТИП = 0001 – система разрешается относительно dx ($'\Delta' x$).

В случае недоопределенной системы уравнений процесс вычислений данным оператором не прерывается. Выделяется независимая подсистема уравнений, число уравнений засылается в ячейку К. В решение будут входить неизвестные буквы.

Оператор НО (0513). Обнуление определителей

Обращение к оператору имеет вид

НО, ТИП, М, n, p, N, К, x;

Номерами $N \div N+[K]-1$ расписывается система соотношений, обращающих в нуль все определители заданного порядка p: ТИП = 0000 – в матрице коэффициентов при x; ТИП = 0001 – в матрице коэффициентов при x и свободных членов.

Оператор АРОМ (0240). АнтиРОМ

Обращение к оператору имеет вид

АРОМ, М, T, m, n, W, i_{min}, j_{min}, j_{max}, P, R;

Данный оператор выполняет функции, обратные оператору РОМ, осуществляя переход от матричной записи системы уравнений к строчной.

В обращении к оператору М – номер формулы, задающей матрицу коэффициентов, результирующая система уравнений прописывается под номером R. Под номером W может быть прописан вектор неизвестных. Если W = 0000, неизвестной считается буква P из обращения с одним индексом, если $T = 010i$, или с двумя индексами, если $T = 000i/$, $i = 1,2,3,4$.

Если матрица расписана вместе со свободными членами, то при T = 0001 (0101) уравнения расписываются со знаком равенства, свободный член переносится вправо и пишется со своим знаком. При T = 0002 (0102) свободный член переносится влево со знаком минус, равенство и правая часть не расписываются.

Если матрица расписана без свободных членов, то при T = 0003 (0103) результирующие уравнения расписываются со знаком равенства и нулевой правой частью. При T = 0004 (0104) знак равенства и правая часть не пишутся. Если W = 0000, то при неизвестной букве расписываются индексы, причем нижний индекс начинается от i_{min}, верхний индекс расписывается в границах $j_{min} \div j_{max}$.

§ 3. ОПЕРАТОРЫ ПРЕОБРАЗОВАНИЯ ФОРМУЛ

Оператор АРИФМ (0201). Арифметик с плотным кодированием

Обращение к оператору имеет вид

АРИФМ, N ;

Здесь N – номер обрабатываемой формулы. Результат прописывается под тем же номером. В формуле N реализуются все связи (знаки операций), определенные на числах.

В основном оператор АРИФМ используется в приведении подобных для приведения числовых коэффициентов, однако он может иметь и самостоятельное применение. Часто оказывается удобным просчитывать при помощи Арифметика значения формул, состоящих только из чисел и связей. Такие формулы могут получаться в процессе решения более сложной задачи либо вводиться самим программистом. Однако следует иметь в виду, что быстродействие Арифметика весьма ограничено. Дело в том, что данный оператор по принципу работы относится к программам интерпретирующего типа.

Оператор АРИФМ состоит из следующих частей: а) интерпретирующая программа; б) шкала связей; в) библиотека программ, реализующих связи.

Работа интерпретирующей программы Арифметика заключается в циклическом просмотре обрабатываемой формулы. Для каждой считанной связи σ в фиксированные ячейки заносятся числовые аргументы этой связи. Если $\sigma \in \Sigma_k$, интерпретирующая программа требует наличия в обрабатываемой формуле двух аргументов этой связи. Для $\sigma \notin \Sigma_k$ полагается достаточным наличие хотя бы одного аргумента, левого или правого. Иными словами, коммутативные связи всегда считаются двуместными, некоммутативные могут быть и одноместными. Если аргументы связи выделены, то по заданной шкале связей управление передается на соответствующую программу, которая анализирует аргументы и (если это возможно) реализует связь и заносит результат в фиксированную ячейку. Интерпретирующая программа переписывает вычисленное значение в обрабатываемую формулу, на место одного из аргументов связи. Сама связь и, при необходимости, второй аргумент затираются. Затем формула приводится к почти нормальному виду и просмотр начинается сначала. Если же выбранная связь σ не имеет числовых аргументов, либо ее нет в шкале связей, либо она не

реализуема с имеющимися числовыми аргументами, просмотр формулы продолжается дольше. Работа оператора заканчивается, если за один полный просмотр формулы ни одна связь не была реализована. Отметим, что при помощи данной программы можно работать только с числами, кодируемыми в четыре тетрады.

В операторе предусмотрено довольно простое расширение количества реализуемых связей. В настоящее время в структуру Арифметика включены связи +, x, /, ↑ (сложение, умножение, деление, возведение в степень). Знак минус в Авто-Аналитике является одноместной связью, его аргументом служит правое число. Таким образом, запись формулы Δ 5 − 6 Δ недопустима. Следует писать Δ 5 + − 6 Δ . В первую очередь некоторая неестественность записи вызвана недоопределенностью операции вычитания в отношении коммутативности. Порядок действий при наличии в обрабатываемой Арифметиком формуле ряда одинаковых неассоциативных связей следует задавать скобками, в противном случае он будет естественным.

Особую роль играет связь * (с кодом 0774), которая называется функциональной связью. Слева от этой связи записывается название функции, справа – аргумент функции. Так, например, функции $\sin x$, e^x, $|x|$ следует записывать в виде SIN * x , EXP * x , MOD * x . В настоящее время Арифметик реализует следующие функции:
sqrt (квадратный корень), mod (модуль), entire (целая часть), exp, ln, sin, cos, tg, ctg, arcsin, arccos, arctg, arcctg, sh, ch, th, cth.

На входном языке Авто-Аналитика они имеют соответственно вид: SQRT, MOD, ENTR, EXP, LN, SIN, COS, TG, CTG, ASIN, ACOS, ATG, ACTG, SH, CH, TH, CTH.

Существенным недостатком оператора АРИФМ является низкое быстродействие. Объясняется оно в основном тем, что производится обработка плотно записанной формулами, с большими потерями времени за счет использования программ записи и чтения.

Оператор ВП (0202). Вписывание

Обращение к оператору имеет вид

ВП, В, A^- , A^+, С;

Вместо произвольной последовательности символов, прописанной под номером A^- , в формулу В вписывается последовательность символов, прописанная под номером A^+. Результат прописывается под номеров С. Вписывание производится только на место первой последовательности. Если вписывания не было, формула С совпадает с формулой В. Логический признак θ (ТЭТА) вырабатывается аналогично оператором подстановки.

Оператор КУ (0203). Коммутативное упорядочение

Обращение к оператору имеет вид

КУ, N ;

В формуле N устраняются инверсии относительно всех связей, указанных в шкале коммутативных связей, иными словами, производится лексикографическое упорядочение относительно коммутативных связей. Оператор КУ работает с плотной записью формул, что соответственно сказывается на быстродействии.

Оператор ПРИП (0216). Приведение подобных

Обращение к оператору имеет вид

ПРИП, N ;

В заданной формуле N раскрываются скобки и приводятся подобные.

Приведение подобных является весьма важной и распространенной операцией при проведении аналитических выкладок на ЭВМ. Строгого понятия приведение подобных в математике не вводится. Обычно под приведением подобных понимают как раскрытие скобок, с целью упрощения структуры формулы и поиска сократимых выражений, так и группировку, то есть вынесение общего множителя за скобку, с целью сокращения длины формулы и придания ей наиболее обозримого вида. Вторая трактовка приведения подобных пожалуй даже более распространена. Тем не менее мы будем придерживаться первой. Объясняется это следующими причинами. Задача

минимизации длины формулы достаточно сложна, решение ее занимает много времени, а в результате может получиться формула крайне неестественного вида. Приведение формулы к обозримому виду за счет вынесения, например, коэффициентов при некоторых буквах при работе необходимо, однако в общем случае по очевидным причинам эта задача не алгоритмизуется. Кроме того, такое представление целесообразно лишь на выходе, а при обработке формул в ЭЦВМ не обязательно.

Реализуется приведение подобных путем циклического применения оператора подстановки и Арифметика следующим образом.

Этап I. Применение оператора Арифметик.

Этап 2. Округление до нуля.

Этап 3. Раскрытие скобок и приведение подобных.

Этап 4. Группирование числовых коэффициентов.

Этап 5. Если на этапах 3, 4 обрабатываемая формула изменилась (прошло хотя бы одно преобразование), производится переход к этапу I. В противном случае оператор работу заканчивает.

Рассмотрим подробнее содержание каждого из этапов. На первом этапе оператор Арифметик производит все операции над числами, не меняя структуры формулы. Так, например, формула $\Delta\,(5+6)\times A\,\Delta$ приведется к виду $\Delta\;11\times A\,\Delta$, однако формула $\Delta\,5\times A+6\times A\,\Delta$ останется без изменений.

Далее при помощи специального оператора ОКРН (округление до нуля) производится замена чистым нулем всех вещественных чисел, не превосходящих по модулю заданную в ячейке ЭПС величину точности нуля. Значение ЭПС задается программистом. Стандартное его значение равно $_{10}-6$. Необходимость операции округления до нуля объясняется тем, что при проведении аналитических преобразований весьма существенным моментом является вынесение тождественного равенства нулю некоторых выражений. В то же время арифметические действия над числами обычно производятся с погрешностями, в результате чего вместо нулей получаются малые числа. Использование округления до нуля до некоторой степени устраняет погрешности. Важно уметь округлять и до целых чисел, однако в общем случае эта проблема повидимому неразрешима.

На третьем этапе приводятся подобные по связям +, x, /, ↑ путем обобщенной подстановки по следующей системе правил.

I. %x / %x1 → %x * %x1↑-1.

2. (%x + %x1) * %x2 → %x * %x2 + %x1 * %x2

3. %x + 0. → %x

4. %x * 0. → 0.

5. %x * 1. → %x

6. %x↑(%x1 + %x2) → %x↑%x1 * %x↑%x2

7. (%x↑%x1)↑%x2 → %x↑(%x1 * %x2)

8. 0.↑%x → 0.

9. %x↑1. → %x

10. 1.↑%x → 1.

11. %x↑0. → 1.

12. (%x * %x1)↑%x2 → %x↑%x2 * %x1↑%x2

13. %x + %x → 2. * %x

В результате полной подстановки по системе равенств в обрабатываемой формуле раскроются скобки относительно перечисленных связей и сократятся подобные попарно.

На четвертом этапе работы оператора совершается подстановка по следующим правилам

I. %x * %x↑%z → %x↑(%z + 1.)

2. %x + %x * %z → %x * (%z + 1.)

3. %z↑%x * %z1↑%x → (%z * %z1)↑%x

4. %x↑%z * %x↑%z1 → %x↑(%z + %z1)

5. %z * %x + %z1 * %x → %x * (%z + %z1)

В результате применения этой системы правил сгруппируются числовые коэффициенты. Под группировкой здесь понимается вынесение за скобку, в результате которого появляется возможность реализации некоторых операций над числами. Например формула Δ 5 * A + 6 * A Δ преобразуется к виду Δ (5+6) * A Δ, затем оператор Арифметик приведет ее к виду Δ 11 * A Δ.

Если на этапах 3 – 4 применилось хотя бы одно правило, существует возможность дальнейшего упрощения обрабатываемой фор-

мулы, поэтому цикл (этапы I-4) повторяется. Если вид формулы не изменился, оператор работу заканчивает.

Таким образом, оператор приведение подобных раскрывает скобки и приводит буквенные и числовые подобные относительно связей, для которых введен список правил.

Настоящий оператор реализует самый простой вариант приведения подобных. Ясно, что во многих случаях введенных преобразований недостаточно. Пользователь должен сам анализировать вид получающихся в его программе формул и вводить по мере надобности дополнительные преобразования. Так, например, введенный в оператор ПРИП список правил не предусматривает приведение к общему знаменателю.

Задача приведения подобных с высокой степенью общности кажется весьма сложной и, по мнению авторов, практически неразрешимой. Дело в том, что существует масса искусственных преобразований, обучить которым машину либо невозможно, либо сопряжено с большими затратами труда и большими потерями машинного времени. Это обстоятельство лишает приведение подобных какой-либо практической ценности. В определенном смысле задача приведения подобных аналогична задаче поиска первообразных.

На основании изложенного следует призвать пользователя к некоторой осторожности при проведении аналитических выкладок, так как неточности в приведении подобных, на первый взгляд несущественные, могут привести к неправильным результатам. Ярким примером могут служить задачи линейной алгебры, где нераспознание тождественного нуля может привести к неверному решению (за счет возможного деления на нуль). Пути решения этой проблемы пока не видно. Не помогает и выдача сомнительных ситуаций для визуального контроля, так как при длине формул, превышающей несколько сот символов, человек по существу бессилен. Разумеется, те же трудности возникают и при ручном решении задач, однако из-за малых размерностей обрабатываемых выражений проблема не кажется слишком актуальной.

Оператор ДИФ (0217). Дифференцирование

Обращение к оператору имеет вид

ДИФ, М, N ;

Задачей оператора является вычисление всех частных производных в массиве формул М с прописью результата под номером N.

Символическое дифференцирование было первой задачей аналитического характера, реализованной на ЭЦВМ. Объясняется это сравнительной простотой и полной алгоритмизуемостью задачи символического дифференцирования. В Авто-Аналитике дифференцирование осуществляется подстановкой по заданному списку правил. Этот список позволяет вычислять производные основных элементарных функций. Производные в исходной формуле должны быть записаны в виде

$$\mathcal{F}\,'P'\,(x1; x2; \ldots; xK).$$

Здесь $\mathcal{F}$ – произвольная подформула, $'P'$ – код операции дифференцирования, $x1, \ldots, xK$ – переменные, по которым производится дифференцирование. Данная запись соответствует $\frac{\partial^K \mathcal{F}}{\partial x_1 \partial x_2 \ldots \partial x_K}$. В качестве вспомогательного оператор ДИФ использует оператор П. В последний для повышения быстродействия дифференцирования встроен блок, обнуляющих выражение под знаком производной, если оно не зависит от одной из переменных дифференцирования. После дифференцирования в формуле N приводятся подобные.

Оператор РОСПВ (0221). Роспись пустых вхождений
Обращение к оператору имеет вид

РОСПВ, N ;

В формулу N на место пустых вхождений вписывается код 1000.

Оператор ОКРН (0225). Округление до нуля
Обращение к оператору имеет вид

ОКРН, N , М, A_ε ;

В формуле N заменяются нулем все вещественные числа, не превосходящие по модулю число ε из ячейки с относительным адресом A_ε формулы М. Если М = 0000, A_ε – адрес в 0-м кубе МОЗУ.

Оператор ВШ (0226). Полное вписывание

Обращение к оператору имеет вид

ВШ, В, A^-, A^+, С;

В формулу В всюду (при помощи циклического просмотра) вместо последовательности символов, прописанной под номером A^-, вписывается последовательность символов, прописанная под номером A^+. Результат прописывается под номером С. Вырабатывается логический признак ТЭТА аналогично операторам подстановки.

Оператор КУПОР (0237). Коммутативное упорядочение с поячеечной записью

Обращение к оператору имеет вид

КУПОР;

Аргументов данный оператор не имеет. Информация о расположении в МОЗУ формулы, расписанной по одному символу в ячейку (в последней позиции), выбирается из рабочих ячеек оператора П. Оператор КУПОР устраняет в этой формуле инверсии относительно связей + и х, если эти связи являются коммутативными.

Оператор А (0241). Арифметик с поячеечной записью

Обращение к оператору имеет вид

А, N ;

Функции и организация данного оператора аналогичны оператору АРИФМ. Оператор А введен в Авто-Аналитик с целью повышения быстродействия. Обрабатываемая формула переводится оператором А в поячеечную запись (символ в ячейку, число в ячейку). При этом переписывании производится округление чисел до нуля по ячейке ЭПС. Для повышения быстродействия в библиотеку реализуемых связей введены только операции +, -, х, /, $\uparrow$. В качестве рабочих для росписи формулы N используются ячейки, начиная с АОРП - L , где L задается условным числом в четвертой ячейке оператора А. При нехватке отведенного поля рабочих ячеек оператор А работу системы прерывает запрещенной командой.

Оператор ИНТ (0300). Интегрирование

Обращение к оператору имеет вид

ИНТ, ТИП, М, A_1, A_2, A_3, N_α, N ;

Прежде чем пояснять значения аргументов, приведем некоторые сведения об алгоритме работы оператора. Как известно, не построен алгоритм, позволяющий вычислять первообразную от любой элементарной функции. Более того, весьма сложно разрешается вопрос о возможности представления первообразной в элементарных функциях в каждом отдельном случае. Существующие программы символического интегрирования основаны на сложных эвристических алгоритмах комбинаторного характера, потребляющих чрезмерное количество машинного времени и обычно не приводящих к решению. Для решения комплексных практических задач этот путь повидимому непригоден. Оператор ИНТ вычисляет первообразные по заданной системе правил, которая может расширяться пользователем. Если этот путь не приводит к решению, по указанию программиста включается блок аппроксимации подынтегральной функции полиномом Чебышева первого рода в заданном промежутке. В качестве результата интегрирования выдается первообразная от аппроксимирующего полинома.

Принята следующая форма записи интегралов

$$((\ldots (F \text{'} S \text{'} x_1) \text{'} S \text{'} x_2) \ldots) \text{'} S \text{'} x_n ,$$

что эквивалентно обычной записи

$$\iint \ldots \int F \, dx_1 \, dx_2 \ldots dx_n .$$

Предполагается, что $x_1, x_2, \ldots, x_n$ – буквы, либо буквы с индексами. Произвольной постоянной данная программа пренебрегает.

Оператор допускает подключение собственных правил математика, использующего систему. Правила могут быть двух типов. К первому типу относятся правила, для которых в качестве аналогов переменных букв из Φ_0 могут выбираться произвольные подформулы подынтегральной функции. Ко второму типу относятся правила, аналоги переменных букв которого не могут зависеть от переменной интегрирования. Правила первого и второго типа прописы-

ваются отдельными массивами, номера которых заносятся соответственно в ИР-6 и ИР-7.

Повторим форму обращения к данному оператору

ИНТ, ТИП, М, A_I, A_2, A_3, N_α, N ;

Здесь М – номер формулы, в которой нужно вычислить все входящие в нее интегралы.

A_I – номер ячейки, в которой записан нижний предел интегрирования. Это число воспринимается также как нижняя граница области определения подынтегральной функции при аппроксимации.

A_2 – номер ячейки, в которой записан верхний предел интегрирования. Этот предел воспринимается также как верхняя граница области определения подынтегральной функции при аппроксимации.

A_3 – адрес ячейки, в которой записана условным числом степень аппроксимирующего полинома.

N_α , $N_\alpha+1$,... – номера формул, которые используются оператором в качестве рабочих.

Если N = 0000, то A_I, A_2, A_3 рассматриваются как адреса нулевого куба МОЗУ. Если $N \neq 0000$, то A_I, A_2, A_3 – адреса относительно начала формулы N .

Распишем тетраду ТИП в двоичной форме

ТИП = 000 000 00i pqs .

Если p = 0, то обращения к блоку аппроксимации не производится;

если q = I, то производится подключение правил математика первого типа;

если s = I, то производится подключение правил математика второго типа;

если i = 0, вычисляемые в формуле М интегралы считаются неопределенными, в противном случае производится подстановка пределов интегрирования.

Из обращения к оператору видно, что если формула М содержит несколько интегралов с различными пределами интегрирования или различными областями определения подынтегральных функций, то каждый интеграл следует выделять в отдельную формулу и лишь затем обращаться к оператору ИНТ.

Оператор АПР (0301). Аппроксимация

Обращение к оператору имеет вид

АПР, A_I, A_2, A_3, N_1, N_2, N_α, К ;

Здесь A_I – адрес ячейки с нижней границей области определения аппроксимируемой функции;

A_2 – адрес ячейки с верхней границей;

A_3 – адрес ячейки, в которой записана степень аппроксимирующего полинома Чебышева первого рода;

N_1 – номер, под которым прописана аппроксимируемая функция;

N_2 – номер, под которым прописана формула, задающая переменную (букву или букву с индексом).

Если К = 0000, то A_I, A_2, A_3 – адреса нулевого куба МОЗУ. Если К ≠ 0000, то A_I, A_2, A_3 – адреса относительно начала формулы К.

§ 4. ОПЕРАТОРЫ ГРУППОВОГО ПРЕОБРАЗОВАНИЯ ФОРМУЛ И ВСПОМОГАТЕЛЬНЫЕ ОПЕРАТОРЫ

Оператор 0200. Операции над формулами

Данный оператор используется только в режиме ручного программирования. При работе с входным языком аналогичные функции выполняет оператор присвоения. Обращение к оператору 0200 имеет вид

С	0200	М	
	ℓ	0000	
К	ИР	КОП	A_I
	ИР	КОП	A_2
	ИР	КОП	A_2

Здесь М – номер, под которым прописывается формула, образованная оператором, ℓ – число ячеек, занятых аргументом (ячейка с номером оператора не считается).

Оператор начинает работу записью левого граничного символа. Далее расписываются символы по виду информационных строк. После работы оператора интегрируется первый символ в следующей за аргументом ячейке.

Если КОП = 34, оператор 0200 записывает символ, заданный последними I2 разрядами ячейки с номером $A_{исп}$.

Если КОП = 32, пишутся последние I2 разрядов $A_{исп}$.

Если КОП = 3I, без граничных символов переписывается формула с номером $A_{исп}$.

КОП = 2I – то же, что и 3I, но формула после записи аннулируется.

КОП = 30 – $A_{исп}$ воспринимается как адрес настройки на формулу, которая переписывается без граничных символов.

КОП = 20 – то же, что и 30, но переписываемые символы затираются на старом месте нулевыми кодами.

Если встречается КОП, отличный от описанных, оператор работу заканчивает. Начинается интерпретация первого символа следующей ячейки, независимо от величины ℓ. Если аргумент не занимает целого числа ячеек, лишнюю половину можно оставить нулевой.

Оператор ОФ (023I). Объединение формул

Обращение к оператору имеет вид

ОФ, ТИП, N_α, N_ω, R ;

Формулы с номерами $N_\alpha, N_\alpha+1, \ldots, N_\omega$ объединяются в массив и прописываются под номером R. Если ТИП = 0000, формулы $N_\alpha, N_\alpha+1, \ldots, N_\omega$ аннулируются.

Оператор ПЧ (0233). Печать формул

Обращение к оператору имеет вид

ПЧ, ТИП, N ;

ТИП = 0001 – печать формулы N ;

ТИП = 0002 – печать массива формул N ;

ТИП = $\alpha_1\alpha_2\alpha_3$3 – печать формул $N, N+1, \ldots, N+\alpha_1\alpha_2\alpha_3-1$ $(\alpha_1\alpha_2\alpha_3 \le 377)$;

ТИП = 0004 – печать всех формул, прописанных в ОАС;

ТИП = 0005 – печать с позиции, указанной настройкой с адресом до ближайшего граничного символа.

Обращение может иметь вид
ПЧ, 0006, ℓ , N_1 , N_2 , ... , N_ℓ .
Печатаются формулы по заданному списку; ℓ - длина списка. Если старший разряд тетрады ТИП равен нулю, оператор разделяет печатаемый текст на страницы. После каждого обращения к оператору страница завершается, независимо от числа отпечатанных строк. Если старший разряд тетрады ТИП равен единице, печать непрерывная.

Оператор РАЗДВ (05II). Раздвоение
Обращение к оператору имеет вид
РАЗДВ, М, n , N ;
Формулы $M \div M+n-1$ прописываются также номерами $N \div N+n-1$. При пересечении номеров оператор работу системы прерывает.

Оператор АННФ (05I2). Аннулирование последовательности формул
Обращение к оператору имеет вид
АННФ, N_α , N_ω ;
Оператор аннулирует формулы $N_\alpha \div N_\omega$

Оператор ПОП (05I5). Печать операторов
Обращение к оператору имеет вид
ПОП, N ;
С соответствующим заголовком распечатывается в командах оператор N .

Оператор ПЯЧ (05I6). Печать ячеек
Обращение к оператору имеет вид
ПЯЧ, N_α , N_ω ;
С соответствующим заголовком распечатываются в константах ячейки $N_\alpha \div N_\omega$ нулевого куба МОЗУ.

Оператор ПНУМ (0520). Перенумерация

Обращение к оператору имеет вид

ПНУМ, M_α, M_ω, N_α ;

Формулам $M_\alpha \div M_\omega$ присваиваются номера, начиная с номера N_α. Формулы $M_\alpha \div M_\omega$ аннулируются. При пересечении номеров формул работа системы прерывается.

Оператор ДЕМ (052I). Деление массива

Обращение к оператору имеет вид

ДЕМ, М, N , К ;

Формулы массива М расписываются также поотдельности номерами $N \div N+[k]-1$. Содержимое ячейки К формируется оператором.

Оператор 0525. Занесение поправок

Используется только в режиме ручного программирования. Позволяет организовать достаточно простое поячеечное занесение поправок в операторы и ранее введенные формулы. Обращение к оператору может быть в любой позиции ячейки, следующие за номером позиции забиваются нулевым кодом. В последующих ячейках располагаются информационные коды. Завершается информация кодом

С IIII IIII IIII IIII.

Информация для каждого отдельного оператора имеет вид

С N_1 0000 0000 0000

С 0000 ℓ_1-1 0000 A_I

код в ячейку A_2

. . . . . .

код в ячейку $A_1+\ell_1-1$

. . . . . .

С 0000 ℓ_k-1 0000 A_K

код в ячейку A_K

. . . . . .

код в ячейку $A_k+\ell_k-1$

С 7777 7777 7777 7777

В формулу N_1 с ячейки A_1 заносится e_1 кодов, с ячейки A_2 - e_2 кодов и так далее. С ячейки A_K заносится e_K кодов. Завершается информация о поправках в формулу N_1 строчкой семерок. Далее может быть любое число аналогичных массивов для других формул. Заданные коды оператор засылает как команды. После окончания работы оператора начинает интерпретироваться символ в ячейке, следующей за строчкой единиц.

§ 5. ОПЕРАТОРЫ ПРИЛОЖЕНИЙ АВТО-АНАЛИТИКА

Основным практическим приложением Авто-Аналитика является реализация метода внешних форм Картана. Впервые в мировой практике опыт реализации на ЭЦВМ отдельных этапов алгоритма Картана произведен В.А. Шурыгиным и Н.Н. Яненко [7]. Предложенная в [7] трактовка метода внешних форм Картана как алгебраическо-дифференциального алгоритма положена в основу программной реализации метода в Авто-Аналитике.

Представление метода Картана в виде машинно-ориентированного алгоритма, исследование различных специфических ситуаций, возникающих при машинной реализации этого алгоритма произвел В.П. Шапеев [22].

В.П. Шапеев сформулировал ряд ценных предложений по структуре операторов, разработал систему тестов.

Обращение к оператору, реализующему метод Картана, имеет вид

КАРТАН, РЕЖИМ, N, n, R, S, N';

Тетрада РЕЖИМ, равная 0000 или 0001 задает вариант алгоритма, $N, N+1, \ldots, N+S-1$ - исходная система, n - число независимых переменных x, R - число неизвестных функций, N' - рабочий номер либо дополнительная система соотношений.

Оператор КАРТАН реализует метод внешних форм Картана путем последовательного обращения к вспомогательным операторам.

Оператор ЗАМ (0500). Замыкание

Обращение к данному оператору имеет вид

ЗАМ;

Под номерами $N+S \div N+R-1$ расписываются уравнения, замыкающие исходную систему Пфаффа. Аргументы выбираются из рабочих ячеек оператора КАРТАН: 04074 (N); 04075 (n); 04076 (R); 04077 (S).

Оператор ВНДИФ (0501). Внешнее дифференцирование

Обращение к оператору имеет вид

ВНДИФ, М, N , m ;

Вычисляется внешний дифференциал формул $M \div M+m-1$ Результат прописывается под номерами $N \div N+m-1$

Оператор РАСЩ (0502). Расщепление

Обращение к данному оператору имеет вид

РАСЩ, N_α , M_α , N ;

В квадратичные формы $N_\alpha \div N_\alpha+S-1$ подставляются равенства $M_\alpha \div M_\alpha+R-1$. Коэффициенты при мономах в полученных выражениях прописываются номерами $N \div N+K-1$ Значение К вычисляется и засылается в ячейку 04073.

Оператор ЦЕПЬ (0503). Построение цепи интегральных элементов

Обращение к оператору имеет вид

ЦЕПЬ, N_α , M_α , N , Q ;

$N_\alpha \div N_\alpha+S-1$ – квадратичные формы, $M_\alpha \div M_\alpha+R-1$ – значения дифференциалов, N – начало массива рабочих номеров формул, с ячейки Q расписываются число Картана и характеры цепи.

Оператор УПЛОТ (0504). Уплотнение

Обращение к оператору имеет вид

УПЛОТ, М, n , N ;

В формулах $M \div M+n-1$ приводятся подобные, тождественные нули отбрасываются, номера уплотняются. Число оставшихся формул засылается условным числом в ячейку N.

Оператор ПЗАВ (0514). Проверка на зависимость

Обращение к оператору имеет вид

ПЗАВ, ТИП, М, n, x_1, x_2, x_3;

Если в соотношениях $M \div M+n-1$ есть хотя бы одно не содержащее: ТИП = 0000 – буквы x_1, ТИП = 0001 – x_1 или x_2, ТИП = 0002 – x_1, x_2 или x_3, логическому признаку ТЭТА присваивается нулевое значение, в противном случае – единичное.

Оператор КВЫХ (0517). Оператор выхода

Обращение имеет вид

КВЫХ, ТИП;

ТИП равен номеру выхода из алгоритма Картана. На печать выводится вся информация необходимая математику. Работу системы оператор не прерывает.

Оператор ВНЕС (0522). Выбор независимых соотношений

Обращение к данному оператору имеет вид

ВНЕС, М, n, x, N, K_I, K_2;

Из формул $M \div M+n-1$ выбираются не зависящие от x и прописываются номерами $N \div N+[K_2]-1$. Оставшиеся формулы прописываются номерами $M \div M+[K_1]-1$. Содержимое ячеек K_I и K_2 заносится оператором.

Оператор ЗПЕР (0524). Замена переменных

Обращение к оператору имеет вид

ЗПЕР, R_1;

В системе $N \div N+R-1$ переменные p^{ω} заменяются новыми неизвестными функциями $u^{R+1}, \ldots, u^{R_1}$. Величина R_1 вычисляется оператором. Равенства, по которым произво-

дится замена, выводятся на печать.

Оператор ДКС (0510). Дифференцирование конечных соотношений

Обращение к оператору имеет вид

ДКС, М, n, К ;

Дифференцируются конечные соотношения $M \div M+n-1$, уплотняются и прописываются номерами $M \div M+[k]-1$. Содержимое ячейки К засылается оператором.

Оператор ВМН (0540). Вынесение общего множителя

Обращение к оператору имеет вид

ВМН, M_α, M_ω, N, R ;

В формулах $M_\alpha \div M_\omega$ выносится за скобку множитель, прописанный под номером N. В формуле N допускаются переменные буквы. R, $R-1$ – рабочие формулы.

Оператор ВКОЭФ (0541). Выведение коэффициентов

Обращение к оператору имеет вид

ВКОЭФ, M_α, M_ω, N, К, R, L ;

Из формул $M_\alpha \div M_\omega$ выводятся коэффициенты при заданном под номером N выражении. Коэффициенты предварительно группируются. Результирующие формулы прописываются номерами К, К + 1, ... , К + [R] – 1. Содержимое ячейки R засылается оператором. L и L + 1 – рабочие номера формул (требуется L = = 1030).

Оператор ПДКФ (0542). Подстановка дифференциалов в квадратичные формы

Обращение к данному оператору имеет вид

ПДКФ, N_α, n, M_α, m ;

Дифференциалы $N_\alpha \div N_\alpha+n-1$ подставляются в квадратичные формы $M_\alpha \div M_\alpha+m-1$. В полученных формулах $M_\alpha \div M_\alpha+m-1$ раскрываются скобки относительно внешних

произведений и упорядочиваются индексы у мономов.

Оператор ВММ (0543). Вынесение общего множителя в массиве формул

Обращение к оператору имеет вид

ВММ, М, N ;

В массиве формул М выносится за скобки множитель, прописанный под номером N .

Оператор ИНД (0544). Упорядочение индексов

Обращение к оператору имеет вид

ИНД, M_α, M_ω;

В формулах $M_\alpha \div M_\omega$ упорядочиваются по возрастанию индексы в мономах.

Оператор МОДЕЛЬ (0600). Моделирование

Обращение к оператору имеет вид

МОДЕЛЬ, ТИП, α , М, N ;

Здесь ТИП = 0000, α - адрес начала информации о схеме, М - номер первой формулы из массива значений параметров, N - номер, под которым прописывается результирующая система уравнений.

Оператор МОДЕЛЬ по заданной пользователем топологической информации о схеме (или, проще, таблице межсоединений), введенной в систему библиотеке моделей компонентов (которая легко может расширяться) и системе равенств, задающих значение параметров компонентов строит систему уравнений, задающих математическую модель схемы в символическом виде. При этом система уравнений может быть самого произвольного вида (линейные, нелинейные, обыкновенные дифференциальные уравнения, уравнения в частных производных и так далее).

Оператор СЕЛЕКТ (0602). Селектор

Обращение к оператору имеет вид

СЕЛЕКТ, ТИП, М, M_I, M_2, N, x, Π;

Оператор расщепляет массив формул М в массивы M_I и M_2. $N, N+1, M_1+1, M_2+1$ – рабочие формулы оператора, x – буква, относительно которой ведется анализ формул при расщеплении, П и П + I – две ячейки нулевого куба МОЗУ, в которые заносится число формул в массивах M_I и M_2. Если ТИП = 0000, то в M_2 собираются линейные относительно x уравнения, прочие – в M_1. Если ТИП = 0001, то в M_I собираются дифференциальные относительно x уравнения, прочие – в M_2. Предполагается, что буква x всегда имеет верхний индекс. Оператор СЕЛЕКТ легко расширяется введением новых критериев расщепления.

Оператор ОПТИМ (0302). Оптимизатор

Обращение к оператору имеет вид

ОПТИМ, T, A, F, n, x, p, К, Δ;

Оператор осуществляет минимизацию заданной целевой функции в заданной области методом покоординатного спуска с дробным шагом.

T = 0000, если счет целевой функции осуществляется подпрограммой с входом в ячейке А и выходом по ИР-15. T = 0001, если счет целевой функции осуществляется оператором с номером А. F – номер ячейки, в которой получается значение целевой функции. Если T = 0001, то F – относительный адрес в операторе А. n – число переменных, по которым ищется минимум, К – число дроблений относительного шага, Δ – адрес ячейки, в которой находится начальное значение относительного шага. С ячейки x заслылается начальное приближение, там же получается оптимальное значение переменных. С ячейки p нижние и (после них) верхние границы области изменения переменных. Если T = 0001, то x и p – адреса относительно начала оператора А.

Оператор ДИФСФ (0303). Дифференцирование сложных функций

Обращение к оператору имеет вид

ДИФСФ, N , М, R ;

Под номером N должна быть прописана сложная функция, подлежащая дифференцированию. Аргументы этой функции прописываются массивом формул под номером M , причем независимая переменная должна быть первой формулой массива. Результат прописывается под номером R .

В заключение данного параграфа и всей книги отметим, что Авто-Аналитик отнюдь не является завершенной системой. Количество и сложность задач аналитического характера требуют его всемерного расширения. Поэтому в настоящей книге довольно подробно описаны основные структуры Авто-Аналитика и сжато - библиотека операторов (описаны далеко не все операторы). Почти не рассмотрены практические приложения системы. Материалы, изложенные в книге, позволяют писать программы вручную и на входном языке. Выбор того или иного способа зависит от склонности программиста и конкретной задачи. Изменение административной системы Авто-Аналитика можно производить только вручную. На взгляд авторов, основные программы новых пакетов прикладных задач следует писать (полностью или частично) вручную, так как быстродействие и компактность программ играют весьма важную роль в аналитических выкладках на ЭЦВМ. Отметим, что описанная в настоящей работе библиотека операторов написана вручную (объем ее составляет приблизительно 50 000 кодов). Лишь после завершения работ над этим, довольно значительным, объемом программ мы сочли возможной разработку специального входного языка Авто-Аналитика и транслятора с него.

Настоящая работа является в некотором смысле предварительным сообщением о системе Авто-Аналитик. Дополнительные материалы, а также сама система и инструкции по ее эксплуатации могут быть высланы нами по согласованию с возможными пользователями, имеющими БЭСМ-6.

Литература

1. Kahrimanian, H.G. Analitical differentiation by a digital computer. M.A. Thesis, Temple U. Phila., May 1953.
2. Nolan, J.F. Analitical differentiation on a digital computer. M.A. Thesis, Math. Dept, M.I.T, Cambridge, Mass., May 1953.
3. Brown, W.S. The ALPAC system for non-numerical algebra on a digital computer. Bell Sys. Tech. J. 42, 5 (Sept. 1963), 2081-2119.
4. Brown, W.S., Hide, J.P., and Tague, B.A. The ALPAC system for non-numerical algebra on a digital computer - II. Rational functions of several variables and truncated power series with rational-function coefficient. Bell Sys. Tech. J. 43, 2 (March 1964), 785-804.
5. Hude, J.P. The ALPAC system for non-numerical algebra on a digital computer - III. Systems of linear equations and a class of side relations. Bell Sys. Tech. J., 43, 4, pt 2 (July 1964), 1547-1562.
6. Т.Н.Смирнова, Полиномиальный прораб и проведение аналитических выкладок на ЭВМ, Труды Математического института им. В.А. Стеклова, 1962.
7. В.А.Шурыгин, Н.Н.Яненко, О реализации на электронных вычислительных машинах алгебраическо-дифференциальных алгоритмов, Проблемы кибернетики, Выпуск 6, 1961.
8. D. Barton, S.R. Bourne and J.R. Horton, The structure of the Cambridge algebra system. The Computer Journal, Volume 13, Number 3, 1970.
9. D. Barton, S.R. Bourne and S.J. Burgess. A simple algebra system. The Computer Journal, Volume 11, No.3.

10. D. Barton. On Literal Developments of the Lunar Theory with the Aid of a Computer. Astronomical Journal, No. 10, 1967
11. Bond, E., Auslander, Grisoff, S., Kenney, R., Myszewski, M., Sammet, J., Tobey, R., and Zilles, S. FORMAC - An experimental Formula Manipulation Compiler. Pros. 19th ACM Nat. Conf. Aug. 1964.
12. Perlis, A. J., Iturriaga, R., and Standish, F. A. A preliminary sceteh of formula ALGOL. Carnegie Inst. of Technology, Rep., April, 1965.
13. Newell, A., Tonge, F. M. An introduction to the IPL-V. Communications ACM, 1960, V. 3, N4.
14. McCarthy, J. The LISP Programming System. Progr. Rep., 1953.
15. Sammet, J. E. An annotated descriptor based bibliography on the use of computers for non-numerical mathematics. Comput. Rev. 2.4 (July - Aug. 1966).
16. В.Л.Катков, А.Ф.Рар . Программирование на языке ЭПСИЛОН. "Наука", Новосибирск, 1972.
17. А.Д.Закревский. Алгоритмический язык ЛЯПАС и автоматизация синтеза дискретных автоматов. Томск, Изд-во Томского ун-та, 1966.
18. В.М.Глушков, В.Г.Боднарчук, Т.А.Гринченко, А.А.Дородницына, В.П.Клименко, А.А.Летичевский, С.Б.Погребинский, А.А.Стогний, Ю.С.Фишман. "Аналитик", Кибернетика, № 3, К., 1971.
19. И.Р.Аксельрод, Л.Ф.Белоус. Входной язык системы автоматического программирования СИРИУС. Харьков, Изд-во Харьк.ун-та, 1969.
20. Л.Эфрос. Введение в программирование для ЭВМ БЭСМ-6, ВЦ СО АН СССР, Новосибирск, 1971.
21. Математическое обеспечение машины БЭСМ-6. ИТМ и ВТ, ВЦ АН СССР, М., 1967.
22. В.П. Шапеев. Логическая схема алгоритма Картана. Комплексы программ математической физики (сборник научных трудов). Новосибирск, 1972.

ОГЛАВЛЕНИЕ

Евгений Александрович Арайс

Геннадий Васильевич Сибиряков

АВТО-АНАЛИТИК

Новосибирский государственный университет

Ответственный за выпуск Е.А.Арайс

Подписано в печать 29.I.1974 г. МН 09036
Бумага 60x84, I/I6. Объем I8 п.л. Тираж 600 экз.
Заказ № II8 Цена 72 коп.

Ротапринт НГУ 630090, Новосибирск

Цена 72 коп.