



**ОРДЕНА ЛЕНИНА
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
АКАДЕМИИ НАУК СССР**

Математическое обеспечение БЭСМ - 6

В.А. Фисун

РЕФАЛ

**(описание языка и инструкция для
РЕФАЛ-интерпретатора)**

Москва, 1974 г.

ОРДЕНА ЛЕНИНА ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
АКАДЕМИИ НАУК СССР

Математическое обеспечение БЭСМ-6

В.А. Фисун

Р Е Ф А Л

(описание языка и инструкция для РЕФАЛ-интерпретатора)

Москва, 1974 г.

А Н Н О Т А Ц И Я

В данном документе описывается версия РЕФАЛа, реализованная в ОС ИМ для БЭСМ-6, правила работы с транслятором-интерпретатором РЕФАЛа, стандартный набор машинных операций языка, и правила пополнения библиотеки машинных операций.

ОГЛАВЛЕНИЕ

Входной язык транслятора с языка РЕФАЛ.	5
1. Введение.	5
2. Структура РЕФАЛ-задач.	6
3. Кодирование задач.	6-7
4. Транслятор РИНТ.	9
4.1. Характеристики транслятора.	9
4.2. Блок синтаксического контроля.	9
4.3. Запуск транслятора.	11
4.4. Распечатки транслятора.	12
5. Библиотека машинных операций.	12
5.1. Встроенные машинные операции.	13
5.1.1. Печать.	13
5.1.2. Арифметические операции.	14
5.1.3. Счетчик.	15
5.1.4. Закопать.	16
5.1.5. Выкопать.	16
5.1.6. Вывод данных.	16
5.1.7. Обращение к стандартным программам.	17
5.2. Правила составления стандартных программ.	17
5.2.1. Стандартная программа-стэк.	18
5.2.2. Стандартная программа - ввод данных.	19
5.2.3. Стандартная программа - прокрутка.	20
6. Примеры запуска задач.	24
Приложение.	30
I. Формальное описание языка РЕФАЛ.	30
I.1. Знаки.	30

1.2. Символы и выражения.	30
1.3. Предложения.	31
2. Синтаксическое отождествление.	31
3. РЕФАЛ-машина.	36
4. Машинные операции.	38
5. Рекурсивные функции.	40
Литература.	44

Входной язык транслятора с языка РЕФАЛI. Введение.

В настоящем документе описывается входной язык и правила работы РЕФАЛ-интерпретатора в ОС ИМ для ЭМ БЭСМ-6.

РЕФАЛ-интерпретатор (РИНТ) - модификация транслятора [6], которая состояла из: включения транслятора в ОС ИМ, расширения набора встроенных машинных операций, добавления аппарата организации библиотеки машинных операций по правилам интерпретирующей системы ОС ИМ.

В приложении приводится формальное описание РЕФАЛА, опубликованное В.Ф.Турчиным в 1968 г. [3] и отредактированное автором настоящего сообщения совместно с В.Ф.Турчиным. Автор выражает глубокую признательность В.Ф.Турчину за постоянное внимание и помощь в работе.

Автор благодарит Л.К.Эйсмонта за существенную помощь в модификации интерпретатора и С.Д.Зеленецкого за включение в РИНТ программы прокрутки.

2. Структура РЕФАЛ-задач.

В соответствии с формальным описанием РЕФАЛа (здесь и далее используются термины языка РЕФАЛ, определенные в приложении) информация для работы РЕФАЛ-машины должна состоять из двух частей: поля памяти и поля зрения. В поле памяти РИНТа помещается набор предложений. Перебор предложений при отождествлении начинается с последнего предложения поля памяти, причем самыми последними считаются "предложения" с детерминативами встроенных машинных операций.

В поле зрения РИНТа должно находиться рабочее выражение, с которого начнется выполнение первого шага РЕФАЛ-машины. Далее в тексте, совокупность поля памяти и поля зрения называется РЕФАЛ-задачей.

3. Копирование задач.

Для РИНТа кроме РЕФАЛ-задачи определяется поле данных. Поле данных-информация, которая может вводиться в поле зрения РЕФАЛ-машины с помощью машинной операции в процессе выполнения задачи.

Возможность вынесения данных в особое поле позволяет проводить последовательную обработку больших массивов информации, которые не помещаются в поле зрения целиком. Это связано с тем, что поле данных хранится на внешних носителях БЭСМ-6 в плотной (по-байтовой) упаковке, а в поле зрения каждому символу соответствует целое слово оперативной памяти.

Символы поля данных вводятся как объектные знаки РИНТа. Поле данных может быть пустым.

3. Кодирование задач.

РЕФАЛ-задачи и данные могут быть подготовлены (отперфорированы) на любом из устройств подготовки перфокарт (УПП), кодировку которых воспринимает программа перекодировки [10]. Для написания задач используются собственные и объектные знаки РЕФАЛа.

Кодировку собственных знаков для РИНТа определяет следующая таблица:

Собственный знак РЕФАЛа	Кодировка в УПП
'	' ИЛИ '
<u>K</u>	<u>K</u>
.	..
§	◇
<u>R</u>	<u>R</u>
<u>S</u>	<u>S</u>
<u>E</u>	<u>E</u>
((
))
⊇	<u>⊇</u> ИЛИ <u>=</u>

Алфавит объектных знаков для РИНТа определен как набор символов УПП [10], кроме знаков $-, \diamond, ', ', (,)$, которые используются для изображения собственных знаков языка и знака $_$ (пробел). Пробелы используются для удобства перфорации задач.

В любом месте РЕФАЛ-задачи может быть употреблен пробел (или последовательность пробелов).

Кроме перечисленных выше объектных знаков, в задачах можно употреблять объектные знаки, которые будут называться специальными. Они служат для изображения символов, совпадающих по начертанию с собственными знаками (скобки, \diamond), для изображения запрещенного символа-подчеркивания и отсутствующего в наборе УШЗ знака авторегистра. Для изображения пробела нужно использовать специальный знак. Исключение см. 4.3.

Ниже приводится таблица кодировки специальных объектных знаков:

Объектный знак	Кодировка знака
'	'/0'
('/1'
)	'/2'
- (подчеркивание)	'/3'
\diamond	'/4'
_ (пробел)	'/5'
знак авторегистра	'/6'

При использовании составных символов РЕФАЛ-задач следует учитывать следующее: составной символ не должен совпадать с изображением специального знака.

Остальные ограничения входного языка:

- число различных свободных переменных в предложении не должно превышать 16;

- число символов в левой части не должно быть больше 100;
- в рабочем выражении, заносимом в поле зрения, глубина скобочной структуры не должна превышать 256;
- запрещено использование имен машинных операций в качестве детерминативов функций РЕФАЛ-задачи.

4. Транслятор РИИТ.

4.1. Характеристики транслятора.

РЕФАЛ-интерпретатор - РИИТ оформлен как подчиненная задача в ОС ИПМ и является транслятором СХЕМЫ [II.]. Входные данные для РИИТа - РЕФАЛ-задача и поле данных указываются в задании для СХЕМЫ.

Имя транслятора - РИИТ.

Ресурсы, необходимы для работы транслятора:

оперативная память (ОЗУ) - 32 К,

память на магнитном барабане (МБ) - 32 К,

файлы вывода на печать - *OUT1* и ВВВОД.

Распределение оперативной памяти.

Заказываемая для РИИТа оперативная память используется следующим образом: 6К заняты программами РИИТа, 3К отводится полю ИС для работы машинных операций - стандартных программ.

Остальные 22К используются для размещения РЕФАЛ задачи.

4.2. Блок синтаксического контроля.

При работе РИИТа первым выполняется блок синтаксического контроля (БСК). Режим работы БСК определяется указателем режима (см. 4.3.) - восьмеричным числом, двоичные разряды ($\epsilon_3 \epsilon_2 \epsilon_1$) которого имеют следующий смысл:

- $\varepsilon_1 = 1$ - печать поля памяти,
 $\varepsilon_2 = 1$ - печать поля зрения,
 $\varepsilon_3 = 1$ - блокировка дальнейшей работы РИНТа.

Если РЕФАЛ-задача не содержит синтаксических ошибок и $\varepsilon_3 = 0$, то после печати информации в соответствии с заданными значениями $\varepsilon_1, \varepsilon_2$ начинается работа других блоков РИНТа. При обнаружении синтаксической ошибки производится печать текста - вид ошибки и фрагмента текста, в котором обнаружена ошибка. Если для поля, в котором обнаружена ошибка, задан режим печати, то распечатка о виде ошибки вставляется в соответствующее место текста. Наличие ошибки в задаче вызывает блокировку вызова других блоков РИНТа по окончании работы БСК.

Приводим тексты распечаток об ошибках в задачах:

Текст распечатки	Пояснение
НЕТ (Нарушен баланс структурных скобок
НЕТ)	-"-
НЕ ОПИС. СВОБ. ПЕР.	В правой части предложения употреблен идентификатор свободной переменной, отсутствующий в левой части.
НЕТ _ ⊃	В предложении нет знака отделения левой части предложения от правой.
ЛИШНЕЕ _К В ЛЕВ.Ч.	Нарушен баланс функциональных скобок в поле памяти.
ЛИШНЕЕ _К В ПРАВ.Ч.	-"-
НЕТ _ . В ПРАВ.Ч.	-"-
НЕТ ПАРНОЙ _ . В П.З.	Баланс функциональных скобок нарушен в поле зрения.
ЛИШНЕЕ _К В П.З.	-"-
ВНУТРИ КАВЫЧЕК ()	Ошибка в объектной цепочке составного символа.

Текст распечатки	Пояснение
ВНУТРИ КАВЫЧЕК	Ошибка в объектной цепочке составного символа.
В КАВЫЧКЕ СИМ > 32	--"
() В КОММЕНТАРИИ	Ошибка в комментарии.
КАВЫЧКИ В КОММЕНТАРИИ	--"
ЧУЖОЙ СИМВОЛ	Использование некорректного символа.
СИНТАКСИЧЕСКИХ ОШИБОК НЕТ.	Печатается в конце работы БСК, если
РАБОТА БСК ЗАКОНЧЕНА!!!	ξ , указателя режима равно I.
ПРОШУ ИСПРАВИТЬ	В задаче найдены синтаксические
СИНТАКСИЧЕСКИЕ ОШИБКИ!	ошибки.
БСК	Выполнение программы не происходит.

4.3. Запуск транслятора РИИТ.

Запуск транслятора осуществляется стандартным запуском СХЕМЫ.

Информационные поля РИИТа в задании для СХЕМЫ располагаются в следующем порядке:

1. Поле памяти.
2. Поле зрения.
3. Поле данных.

Разделителями между полями служит последовательность двух символов $\diamond \diamond$

Поле данных может отсутствовать, в этом случае второй разделитель $\diamond \diamond$ опускается. Первым символом поля зрения является символ-указатель режима работы блока синтаксического контроля РИИТа.

Указатель режима - восьмеричное число или пробел. Пробел эквивалентен указателю \emptyset , что означает блокировку печати текста задачи и продолжение работы РИНТа после синтаксического контроля (см. 4.2).

4.4. Распечатки транслятора.

При работе транслятора могут быть получены распечатки на различных этапах работы РИНТа. Ниже приводится перечень распечаток.

Текст распечатки	Пояснения
КОНКРЕТИЗАЦИЯ ВЫПОЛНЕНА ЧИСЛО ШАГОВ <N> СВОБОДНАЯ ПАМЯТЬ ИСЧЕРПАНА ЧИСЛО ШАГОВ <N> ОТОЖДЕСТВЛЕНИЕ НЕВОЗМОЖНО ЧИСЛО ШАГОВ <N>	Окончена работа задачи. Выполнено N шагов РЕФАЛ-машины. При работе задачи на N шаге не хватало места для поля зрения. Работа РИНТа прекращается. На шаге работы РЕФАЛ-машины при переборе поля памяти не было найдено предложение, по которому можно было бы отождествить очередное объектное выражение. Работа РИНТа прекращается.

5. Библиотека машинных операций.

Библиотека РИНТа состоит из встроенных машинных операций (МО) и стандартных программ РИНТа. Обращение к стандартным программам, составленным по особым правилам (см. 5.2) производится специальной МО.

5.1. Встроенные машинные операции.

Эта группа МО всегда находится в памяти РИПТА и их детерминативы считаются описанными в РВФАИ-задаче.

Использование детерминативов машинных операций в качестве детерминативов функций в задачах запрещается.

Перечень встроенных МО и их детерминативы:

- печать: П;
- арифметические операции: ВЧСЛ;
- счётчик: СЧ;
- закопать: ЗК;
- выкопать: ВК;
- вывод данных: ВВВД;
- обращение к стандартным программам: МО-СП.

5.1.1. Машинная операция-печать.

Обращение: К'П'Е1 .

Машинная операция печати служит для вывода на печать, через файл OUT1 , содержимого поля зрения - Е1. Поле зрения после выполнения машинной операции - пусто.

МО обеспечивает простейшее редактирование печатаемого текста:

- каждая порция информации, соответствующая одной операции, печатается с новой строки АЦПУ;
- обращение к МО с пустым аргументом производит протяжку бумаги на одну строку;
- если выводимый текст не помещается на одной строке АЦПУ, то происходит перенос на начало следующей строки.

Перед каждым символом редактирования должен находиться символ авторегистра, который изображается в поле зрения специальным символом ' / 6 ' .

При использовании символов редактирования необходимо придерживаться следующих правил:

- символ редактирования и предшествующий ему символ авторегистра должны находиться в одной порции вывода М0 - печати;
- появление в выводимом тексте любого символа редактирования (кроме пробела) отменяет автоматический переход на новую строку и управление печатью производится этими символами. В конце выводимого одной машинной операцией текста всегда помещаются символы ПС, ВК.

Для управления размещением выводимого текста на строке АЦУ можно использовать редактирующие символы [12] ,

Приводим их кодировку:

Символ редактирования	Кодировка на РЕФАЛе
ПРОБЕЛ	F
ПРОБЕН	П
ПС (перевод строки)	+
ВК (возврат каретки)	, (запятая)
КП (конец передачи)	4

5.1.2. Арифметические операции.

Обращение: К'ВЧСЛ'ΣЕ1,Е2.

Машинная операция 'ВЧСЛ' служит для реализации арифметических операций.

Операцией, значением $\Sigma 0$, может быть одна из следующих:

+, -, /, x, † .

Операнды, Е1 и Е2 целые или дробные десятичные числа.

Формат операндов следующий: $[\pm] \varepsilon_1 \dots [.] \dots \varepsilon_n$, где . служит

для отделения целой части числа от дробной, а ε_i изображают десятичные цифры. Операнды, представляемые последовательностью символов поля зрения переводятся в стандартные числа с плавающей запятой ЭВМ БЭСМ-6. Операции выполняются командами машины, соответственно, сложения, вычитания, деления, умножения. Операция возведения в степень реализуется экстракодами по следующей формуле: $\underline{E1} + \underline{E2} = \text{EXP}(LN(\underline{E1} * \underline{E2}))$.

Результат переводится в последовательность цифр, которые помещаются в поле зрения. Незначащие нули и знак + подавляются.

Например: после выполнения операции $\underline{K}'\text{ВЧСЛ}'\times 42,10 \pm$ в поле зрения поместятся символы 420.

5.1.3. Машинная операция - счетчик.

Детерминатив операции: 'СЧ'.

Машинная операция служит для работы со счетчиками.

Имеются два вида обращения.

5.1.3.1. $\underline{K}'\text{СЧ}' + \underline{E1} = \underline{E2}$

Выполнение этого вида обращения вызовет запоминание $\underline{E2}$ под именем $\underline{E1}$. Поле зрения после выполнения - пусто. $\underline{E1}$ и $\underline{E2}$ не должны содержать более 6 объектных знаков каждый.

5.1.3.2. $\underline{K}'\text{СЧ}'\underline{E1}$

Машинная операция с таким аргументом заносит в поле зрения последовательность объектных знаков, которая была запомнена ранее под именем $\underline{E1}$. Машинная операция - "счетчик" - отличается от МО - "закопать" и "выкопать" (см. 5.1.4., 5.1.5.) тем, что запоминаемые знаки упаковываются в машинное слово БЭСМ-6 и занимают одну

ячейку ОЗУ. Используя введенные операции, можно записать увеличение счетчика А на единицу в виде:

$$\underline{K}'\text{СЧ}'\text{А} = \underline{K}'\text{ВЧСЛ}' + \underline{K}'\text{СЧ}'\text{А}_2, \text{I}_2$$

5.1.4. Машинная операция - закопать.Обращение: К'ЭК'САЕ2

Машинная операция "закапывает" (запоминает) часть поля зрения, определяемое Е2 под именем СА. СА - произвольный символ программы. Содержимое поля зрения после выполнения машинной операции - пусто. Следует иметь в виду, что запоминаемый текст хранится в памяти в списковой форме, также как и представляется в поле зрения.

5.1.5. Машинная операция - выкопать.Обращение: К'БК'СА

После выполнения такой машинной операции в поле зрения будет помещен текст, закопанный ранее под именем СА и содержимое СА - очищается.

Пример: Для того, чтобы пополнить текст, записанный ранее под именем 'АЛФАВИТ' буквой Б, можно использовать следующие операции: К'ЭК' 'АЛФАВИТ' К'БК' 'АЛФАВИТ' Б

5.1.6. Машинная операция - вывод данных.

Машинная операция - вывод данных (ВЫВД) служит для обеспечения связи транслятора РИИТ с системой программирования ОС ИМ [11]. Связь между трансляторами (работами) в системе программирования происходит через особый файл - 200 канал. В частности, все поля РЕФАЛ-программы передаются РИИТу как содержимое 200-канала. Транслятор РИИТ может передавать информацию следующим работам в задании также через 200 - канал. Для вывода информации в 200 - канал и служит операция ВЫВД.

Обращение: K'ВЫВД'EI.

Ограничения на аргументы операции следующие:

- нельзя выводить составные символы;
- специальный знак авторегистра '/6' служит признаком ошибок и блокирует запуск следующих после РИНТа трансляторов в задании;
- последним выводимым символом в 200 канал должен быть символ $\bar{7}$ (логическое отрицание).

5.1.7. Обращение к стандартным программам.

Детерминатив 'МО-СП'.

Данная машинная операция служит для обращения к стандартным программам.

Обращение: K'МО-СП' E₁ E₂ ? E] ,

где

$E_1 E_2$ - десятичные цифры, номер стандартной программы (СП).

? - тип обращения: , - обычное обращение
: - фиксация СП на поле ИС
; - отмена фиксации СП

EI - аргумент стандартной операции.

Правила составления стандартных программ для РИНТа приводятся ниже (см. 5.2).

5.2. Правила составления стандартных программ:

Стандартные программы должны быть оформлены по правилам подготовки СП в ОС ИПМ и записаны в архив под именами

КСАР $E_1 E_2$, где $E_1 E_2$ (десятичные цифры)

- номер СП.

Если результат работы СП-текст, который надо поместить в поле зрения, то он формируется на поле СП. Перед выходом из СП адреса начала и конца текста нужно записать в пару подряд идущих ячеек. Адрес первой из них хранится в младших разрядах ячейки 00002. Если результат работы - пустая последовательность, то адрес начала текста устанавливается в ноль.

Формируемый текст должен составляться с учетом представления символов поля зрения. Каждый символ располагается в 47-39 разрядах машинного слова в коде УШ БЭСМ-6,

Некоторые символы кодируются особо:

(как 200

) как 300

F как I4I

Ø как I4Ø

Специальные знаки и составные символы нельзя записывать в поле зрения.

Для выборки аргументов поля зрения используется подпрограмма РИИТ а - "дай символ".

Обращение к подпрограмме - команда передачи управления по адресу 5055 с возвратом по I индекс-регистру. После обращения адрес текущего символа заносится в 9 индекс-регистр. Символы аргумента выбираются последовательно, начиная с первого. Адрес последнего символа аргумента стандартной программы хранится в ячейке I350 РИИТа.

5.2.1. Стандартная программа - стек.

Номер СП - 00.

СП-стек предназначена для работы с таблицей.

Виды обращения:

1. Запись: $\underline{K}'MO-CP' \emptyset \emptyset, \uparrow \underline{EI} = \underline{E2}, \underline{}$

В первую свободную строку таблицы заносится код $\underline{E2}$ с ключом \underline{EI} . Ключ и код могут состоять только из букв и цифр и число символов в каждом понятии не должно превышать 8. Поле зрения после выполнения - пусто.

2. Поиск: $\underline{K}'MO-CP' \emptyset \emptyset, \underline{EI}, \underline{}$

По ключу \underline{EI} в таблице ищется код с таким ключом, записанный в таблицу последним. В поле зрения заносится найденный код. Если в таблице нет кода с заданным ключом, в поле зрения помещается символ, буква \emptyset .

3. Провести черту: $\underline{K}'MO-CP' \emptyset \emptyset, <, \underline{}$

В первую свободную строку таблицы заносится служебный код - "черта". Поле зрения после выполнения - пусто.

4. Убрать черту: $\underline{K}'MO-CP' \emptyset \emptyset, >, \underline{}$

Содержимое таблицы до последней "черты" считается свободным. Поле зрения после выполнения - пусто.

Стандартная программа перед началом работы должна фиксироваться на поле ИС. Длина СП - II57 кодов.

5.2.2. Стандартная программа - ввод данных.

Номер СП - $\emptyset I$.

СП предназначена для ввода символов поля данных.

Обращение: $\underline{K}'MO-CP' \emptyset I, \underline{}$

Каждое обращение помещает в поле зрения очередные 72 символа поля данных. По исчерпанию поля данных в поле зрения помещается текст, состоящий из одного символа * . Программа должна фиксироваться на поле ИС. Длина программы - 2726 кодов.

5.2.3. Стандартная программа - прокрутка.

Номер СП - 05.

Стандартная программа - прокрутка, предназначенная для отладки задач, печатает стандартную информацию о шагах РЕФАЛ-машины.

Стандартная информация состоит из:

номера шага РЕФАЛ-машины,
 конкретизируемого выражения,
 результата замены.

Указание о шагах, для которых печатается эта информация, задается в параметрах обращения к СП-прокрутки. Обращение к прокрутке может встретиться на любом шаге РЕФАЛ-машины. Перед началом работы стандартная программа фиксируется на поле ИС выражением:

К'МО-СП'05:.

Пример 1.

Если на каком-либо шаге РЕФАЛ-машины ведущим термом поля зрения будет выражение: К'МО-СП'05,.
 то печать стандартной информации осуществляется для всех последующих шагов.

Пример 2.

Если ведущим термом M -ого шага РЕФАЛ-машины станет выражение: К'МО-СП'05,(I0):(20).
 то печать стандартной информации начнется с $M+I0$ шага и закончится на $M+20$ шаге РЕФАЛ-машины.

Пример 3.

Если обращение к прокрутке имеет вид:

К'МО-СП'05,'A':'B'.

то печать стандартной информации начнется с шага, на котором будет

конкретизироваться функция с детерминативом 'А' (в дальнейшем такой детерминатив будем называть детерминативом шага) и закончится на шаге с детерминативом 'В' .

Пример 4.

Если обращение к прокрутке имеет вид:

$\underline{K' MO - CI' \emptyset 5, 'A'(3):'B'(2)}_{\underline{A}}$

то печать стандартной информации начнется, после чего, как детерминатив шага 'А' встретится 3 раза и закончится тогда, когда в процессе прокрутки детерминатив шага 'В' встретится во второй раз.

Пример 5.

Если ведущим термом M -го шага РЕФАЛ-машины станет выражение: $\underline{K' MO - CI' \emptyset 5, (I \emptyset) = 'A', 'B'}_{\underline{A}}$

то, начиная с $M + 10$ шага и до конца РЕФАЛ-задачи, стандартная информация будет печататься для шагов с детерминативами 'А' и 'В'.

Пример 6.

Если ведущим термом M -го шага РЕФАЛ-машины станет выражение: $\underline{K' MO - CI' \emptyset 5, \neq 'A', 'B': (I \emptyset)}_{\underline{A}}$

то в ближайшие 10 шагов стандартная информация будет печататься для всех шагов кроме тех, детерминатив которых равен 'А' или 'В'.

Пример 7.

Если обращение к прокрутке имеет вид:

$\underline{K' MO - CI' \emptyset 5, 'A': 'B'; (2): (5); \neq 'B'}_{\underline{A}}$

то сначала будет печататься стандартная информация о всех шагах между шагами с детерминативами 'А' и 'В' ; потом будет печататься информация о всех шагах, начиная с $M+2$ и по $M+5$ шаг (где M - шаг РЕФАЛ-машины, на котором встретился детерминатив 'В') и, наконец, начиная с $M+5$ шага и до конца РЕФАЛ-задачи будет печататься информация о всех шагах за исключением тех, детерминатив которых равен 'В' .

Формальное описание обращения к прокрутке.Синтаксис.

<обращение к прокрутке> ::= K'MO-CP'05, <список заданий> .

<список заданий> ::= <задание на прокрутку> |

<список заданий>; <задание на прокрутку>

<задание на прокрутку> ::= [<указатель начала>]

[<указатель условия>]

[<указатель конца>]

<указатель начала> ::= <детерминатив> |

<детерминатив> (<целое десятичное без знака>)|

(<целое десятичное без знака>)

<указатель условия> ::= <детерминатив> |

≠ <детерминатив> | <указатель условия>, <детерминатив>

<указатель конца> ::= <детерминатив> |

<детерминатив> (<целое десятичное без знака>)|

:(<целое десятичное без знака>)

Семантика.

В общем виде обращение к прокрутке состоит из набора независимых заданий на прокрутку. Количество этих заданий не ограничено и обработка их производится последовательно слева направо. Каждое задание на прокрутку определяет диапазон прокручиваемых шагов независимо от остальных заданий.

Начало и конец диапазона прокручиваемых шагов задаются указателями начала и конца, а в указателе условия задаются шаги РЕФАД-машины, для которых в этом диапазоне печатается стандартная информация.

Если при работе прокрутки ведущим термом поля зрения станет новое обращение к прокрутке (прокрутка в прокрутке), то старое обращение к прокрутке игнорируется и выполняются указания нового задания. Если указателем начала является детерминатив, то прокрутка начинается с шага, детерминатив которого равен заданному.

Детерминативы, употребляемые в задании на прокрутку не могут совпадать с символами $,,,:=,\neq$.

Если указатель начала представляет из себя последовательность, состоящую из детерминатива и числа N заключенного в скобки, то прокрутка начнется как только в N -ый раз встретится данный детерминатив шага.

Если указателем начала является число N , то прокрутка начнется с $N+M$ -ого шага, где M -текущий шаг РЕФАЛ-машины.

Если указатель начала отсутствует, то прокрутка начнется с M -ого шага РЕФАЛ-машины.

Семантика указателя конца аналогична указателю начала. Если указатель конца отсутствует, прокрутка продолжается до конца задачи.

Если указателю условия является список детерминативов, отделенных друг от друга запятой и начинающаяся со знака $=$, то стандартная информация печатается только для тех шагов, детерминативы которых перечислены в списке.

Если указатель условия начинается со знака \neq , то информация печатается для всех шагов между началом и концом прокрутки, за исключением тех, детерминативы которых указаны в списке.

В прокрутке предусмотрена выдача следующих сообщений:

1) В начале работы СП-прокрутка выдается сообщение:

ЗАДАНИЕ НА ПРОКРУТКУ и текст самого задания.

2) В конце диапазона K -ого задания на прокрутку из списка заданий выдается сообщение:

ТЕКУЩИЙ ШАГ: <значение шага РЕФАЛ-машины>

КОНЕЦ <K> ЭЛЕМЕНТА СПИСКА ШАГОВ

и управление передается следующему заданию списка заданий.

- 3) Если прокрутка оканчивается раньше, чем РЕФАЛ-задача, то выдается сообщение:

ТЕКУЩИЙ ШАГ : < значение шага РЕФАЛ-машины >
 КОНЕЦ ПРОКРУТКИ.

- 4) Если число детерминативов в указателе условия больше 16, то детерминативы, начиная с 17-го игнорируются и выдается сообщение:

ПРЕДУПРЕЖДЕНИЕ: ДЛИННОЕ УСЛОВИЕ ПРОДОЛЖЕНИЯ ПРОКРУТКИ

- 5) При наличии ошибки в синтаксисе в K -ом задании на прокрутку выдается сообщение:

ОШИБКА: НЕВЕРНЫЙ СИНТАКСИС ЗАДАНИЯ НА ПРОКРУТКУ

после чего управление передается очередному заданию списка заданий.

Если в числе, заключенном в скобки обнаружен нецифровой символ, дополнительно выдается сообщение:

НЕ ЦИФРА .

- 6) Если неверно задан диапазон прокрутки в задании на прокрутку (прокрутка начинается с шага, номер которого больше указанного в условии конца) выдается сообщение:

ОШИБКА: НЕВЕРНЫЙ ДИАПАЗОН ПРОКРУТКИ.

и управление передается следующему заданию списка заданий.

6. Примеры запуск задач.

- 6.1. Пример I показывает использование РИПТа для запуска написанного на РЕФАЛе компилятора с языка СИМУЛа на язык АЛМО. Задание для схемы здесь состоит: из: транслятора EDITOR,

который вводит (читает из архива) в поле памяти РИИТ раздел КСИМ; транслятора РИИТ; транслятора АЛМО; транслятора МТЕСТ - программы, полученной в результате работы транслятора АЛМО.

Программа КСИМ (компилятор СИМУЛА-АЛМО) использует для вывода в 200 канал функцию 'АЛМО', для обращения к стеку - 'СТ' и её работа начинается с детерминатива 'ТРАП' (см. 5.1.6., 5.2.1.)

Замечание. Эксплуатируемый в ОС ИИМ компилятор с языка СИМУЛА запускается специальным транслятором [13], для которого нужно задавать только программу на СИМУЛе. Этот транслятор производит формирование РЕФАЛ-задачи и вызов транслятора РИИТ.

- 6.2. В примере 2 показана задача, которая считает число предложений в разделе КСИМ.
- 6.3. Пример 3 иллюстрирует применение стандартной программы прокрутки.

СУБАВА К ЗАВАЧЕ МНМ.ОС 83СМ-6

СУБАВА ПЕЧАТЬ К ЗАВАЧЕ МС 8АФ

```

0001
0002 MNP
0003 -ARMO TTTT(START)
0004 MNR1 (TEXT)

0005 MNP1
0006 PRMMH1
0007 ZMFX-MVOP,
0008 PRBH,
0009 (1)
0010 PMY
0011 (START)
0012 (START)
0013 EDITOR,START)
0014 ЧЕ ШКОТЯ КСМ
0015 ОЕ
0016 ((FIN)
0017
0018 -L'CT:EL-K-MO-CN100,EL-
0019 -K'AMO'-EL-CK'UBH'-EL-
0020 -L'MO-CN:00:..
0021 -L'TPAP,
0022 'BGIN',INTEGER,1,J)
0023 I:1000)
0024 SIMUL,BEGIN,,INTEGER,M,ACTIVU,PREBTO(P),'REAL,P)
0025 'BGIN'
0026 TPJLEI:THEM'BGIN'
0027 M(J)P(J)=J+1,END)
0028 L:TPMEXTCURRENT)/MONE',THEM,0,0,T,M)
0029 TP,EXTIME(MEXTCURRENT))-TIME2,THEM,M,ACTIVATE,,NEW'PREBTO
0030 (TIME-2)/BLAV,2,MOLD(2,P)),0,0,T,L,END,PREBETA),INTEGER,M)
0031 ,ARRAV,MI:1)
0032 M:2)
0033 J:M)
0034 M:50)
0035 OUTPUT(L,'E',M),ACTIVATE,,NEW'PREBTO(3,0)'AV',S,MOLD(M))
0036 OUTPUT(L,'Y',,,'C',-20)),
0037 ,I:10MEXPIHNCAL:IGNPOCT,4,M,C,0,0,I',,
0038 ,,'C',(-20)),)
0039 ,FOR,I,1,STEP:1,UNTIL,4,1,00,
0040 OUTPUT(L,'Y',,,'E',M,(2,0,0)/0,(3)'EX'/(2,0/4(I))'S',(3)I',)
0041 OUTPUT(L,'Y',,,'C',(-20)),)
0042 ,END:SPATOCCH
0043 ,END.
0044 -
0045 (1)
0046 (FIN)
0047 (FIN)
0048

```


ПРИМЕР 2.

ВЫВАЧА К ЗАДАЧЕ ИММ.ОС ВЗСМ-6
f

```

0001 ВЪВАЧА ПЕЧАТЬ К ЗАДАЧЕ ИММ
0002 (START)
0003 ОК'А'..Ъ1Ъ.К'А'..К'ИМОСН'Ъ1Ъ...
0004 ОК'А'..Ъ1Ъ.Ъ1Ъ.Ъ1Ъ.Ъ1Ъ.Ъ1Ъ...
0005 ОК'ВУСН'Ъ1Ъ.Ъ1Ъ.Ъ1Ъ.Ъ1Ъ.Ъ1Ъ...
0006 ОК'А'..Ъ1Ъ.Ъ1Ъ.Ъ1Ъ.Ъ1Ъ.Ъ1Ъ...
0007 ОК'И'ЧНОП'Ъ1Ъ.ПРЕДЛОЖЕНИЯ//Ъ1Ъ//Ъ1Ъ.К'ИСУС'..
0008 ..
0009 ..
0010 ОК'ИМОСН'Ъ1Ъ...К'ЕСУ'ЕСО...К'А'..
0011 ..
0012 (I
0013 EDITOR(START)
0014 ЧТЕ ВМОТИ КСММ
0015 ОКЕ
0016 (I)(FIN)
0017 (FIN)
0018

```

```

ВЫВАЧА ОУТ1 К ЗАДАЧЕ ИММ
ЧТИ ВМОТИ КСММ
КСЕ
КОЛЧЕСТВО СТРОК В ТЕКСТЕ -- 1674
ЗАВАНО ЛОВАЕ ДАННЫХ
РЕФАЛ-ИНТЕРПРЕТАТОР
ЧИСЛО ПРЕДЛОЖЕНИЙ В 940
ТР,ЧИСЛАИЯ ЗАКОНЧЕНА ЧИСЛО ВАРОВ.РЕФАЛ МАШИНЫ 9451

```

.....
 ЗАВАЧА ИММ ДАТА 17.7.74 НАЧАЛО СЧЕТА 20.00.00 КОНЕЦ СЧЕТА 21.02.41 ВРЕМЯ СЧЕТА 00.00.23 ОС.МММ 07 01.00.74

Приложение.I. Формальное описание языка РЕФАЛ.I.1. Знаки.

$\langle \text{знак} \rangle ::= \langle \text{собственный знак} \rangle \mid \langle \text{объектный знак} \rangle$
 $\langle \text{собственный знак} \rangle ::= \langle \text{скобка} \rangle \mid \langle \text{управляющий знак} \rangle$
 $\langle \text{скобка} \rangle ::= \langle \text{символьная скобка} \rangle \mid$
 $\quad \langle \text{структурная скобка} \rangle \mid \langle \text{функциональная скобка} \rangle$
 $\langle \text{символьная скобка} \rangle ::= = '$
 $\langle \text{структурная скобка} \rangle ::= = (\mid)$
 $\langle \text{функциональная скобка} \rangle ::= = \underline{K} \mid \div \mid \geq$
 $\langle \text{управляющий знак} \rangle ::= = \$ \mid \langle \text{указатель переменной} \rangle \mid \underline{R}$
 $\langle \text{указатель переменной} \rangle ::= = \underline{S} \mid \underline{W} \mid \underline{E}$

Объектным знаком может быть любой, отличный от собственного знака. Предполагается, что алфавит объектных знаков конечен, хотя он и не фиксируется в описании языка.

I.2. Символы и выражения.

$\langle \text{символ} \rangle ::= \langle \text{объектный знак} \rangle \mid \langle \text{составной символ} \rangle$
 $\langle \text{составной символ} \rangle ::= = ' \langle \text{объектная цепочка} \rangle '$
 $\langle \text{объектная цепочка} \rangle ::= \langle \text{объектный знак} \rangle \mid \langle \text{объектная цепочка} \rangle \langle \text{объектный знак} \rangle$
 $\langle \text{выражение} \rangle ::= \langle \text{пусто} \rangle \mid \langle \text{выражение} \rangle \langle \text{терм} \rangle$
 $\langle \text{пусто} \rangle ::= =$
 $\langle \text{терм} \rangle ::= \langle \text{символ} \rangle \mid \langle \text{свободная переменная} \rangle \mid$
 $\quad (\langle \text{выражение} \rangle)$
 $\langle \text{свободная переменная} \rangle ::= \langle \text{указатель переменной} \rangle \mid$
 $\quad \langle \text{идентификатор} \rangle$
 $\langle \text{идентификатор} \rangle ::= \langle \text{объектный знак} \rangle$

1.3. Предложения.

- $\langle \text{предложение} \rangle :: = \S \langle \text{комментарий} \rangle \langle \text{указатель направления} \rangle$
 $\langle \text{левая часть} \rangle \supseteq \langle \text{правая часть} \rangle$
 $\langle \text{комментарий} \rangle :: = \langle \text{объектная цепочка} \rangle$
 $\langle \text{указатель направления} \rangle :: = R \langle \text{пусто} \rangle$
 $\langle \text{левая часть} \rangle :: = K \langle \text{типовое выражение} \rangle$
 $\langle \text{правая часть} \rangle :: = \langle \text{выражение} \rangle$
 $\langle \text{набор предложений} \rangle :: = \langle \text{предложение} \rangle |$
 $\langle \text{набор предложений} \rangle \langle \text{предложение} \rangle$

2. Синтаксическое отождествление.

- 2.1. Свободные переменные вида $\underline{S}A$, $\underline{W}A$, $\underline{E}A$, где A - идентификатор, назовем, соответственно, свободными переменными символа, термина и выражения. Идентификатор однозначно определяет свободную переменную; никакое выражение не может содержать свободных переменных с разными указателями, но с одинаковыми идентификаторами.
- 2.2. Выражение, которое не содержит функциональных скобок (но, вообще говоря, содержит свободные переменные), мы будем называть типовым выражением.
- Выражение, не содержащее свободных переменных, но возможно содержащее функциональные скобки, мы будем называть рабочим выражением.
- Выражение, в котором не содержится ни свободные переменные, ни знаки конкретизации, называется объектным выражением. В таблице перечислены определенные выше понятия выражения и указано, какие знаки они могут содержать (+), и какие не могут (-).

	$\underline{S} \underline{W} \underline{E}$	$\underline{K} \underline{.}$	' ()	Объектные знаки
Объектное выражение	-	-	+	+
Рабочее выражение	-	+	+	+
Типовое выражение	+	-	+	+
Выражение	+	+	+	+

2.3. Будем говорить, что объектное выражение E_0 может быть отождествлено как типовое выражение E_t , если свободные переменные символов, термов и выражений, входящие в E_t можно заменить такими символами, термами и выражениями, соответственно, что в результате выражение E_t совпадает с E_0 . Эти символы, термы и выражения будем называть значениями соответствующих свободных переменных. Если свободная переменная входит в E_t более одного раза, все её вхождения должны быть заменены одним и тем же значением.

2.3.1. При наличии в типовом выражении E_t более одной свободной переменной выражения может оказаться, что существует несколько вариантов придания значений свободным переменным, которые приводят к отождествлению. Эта неоднозначность может быть снята двумя способами, из которых первый мы будем называть отождествлением слева направо (или в нормальном направлении), а второй - отождествлением справа налево (или в обратном направлении). Ниже (пункт 2.3.2) мы описываем отождествление слева направо, указывая в скобках те изменения, который надо внести, чтобы получить отождествление в обратном направлении.

2.3.2. Из всех вариантов выбирается тот, в котором свободная переменная выражения, встречающаяся первой, если просматривать E_t слева направо (справа налево), игнорируя те свободные переменные, которые заключены в скобки, принимает значение, содержащее минимальное число символов и скобок. Если это не устраняет неоднозначности, то такой же отбор производится по второй слева (справа) свободной переменной выражения и т.д. По отношению к каждому выражению E_t' , которое, будучи заключено в скобки, образует один из термов E_t , т.е. к такому E_t' , что E_t можно представить в виде $E_t''(E_t')E_t'''$, неоднозначность устраняется тем же способом, что и по отношению к E_t в целом.

2.4. Пункт 2.3. определяет алгоритм синтаксического отождествления, который применим всякий раз, когда задано некоторое объектное выражение E_0 и некоторое типовое выражение E_t и результатом применения которого является, во-первых, заключение о возможности или невозможности отождествления E_0 как (частного случая) E_t и, во-вторых, если отождествление возможно, то сопоставление каждой свободной переменной из E_t некоторого символа, термина или выражения (соответственно типу свободной переменной), называемого значением этой свободной переменной.

2.5. Примеры и пояснения.

Выражение $A \vee A((B)) \wedge \wedge \wedge$ может быть отождествлено как $\underline{S1} \underline{B} \underline{S1} \underline{E2} \underline{WA}$, причем $\underline{S1}$ примет значение A , $\underline{E2} - ((B)) \wedge \wedge$, а $\underline{WA} - \wedge$.

Выражение $B \vee B \vee B$ может быть отождествлено, как $\underline{E1} \underline{E1} \underline{E1}$, но не как $\underline{E1} \underline{E1} \underline{E1}$.

Рассмотрим процесс отождествления выражения

$$ABA + (B+A)C - + + \quad (1)$$

как (частного случая) выражения

$$\underline{E}1 + \underline{E}2 \quad (2)$$

Пусть отождествление идет слева направо. Поскольку $\underline{E}1$ должно принять минимальное из всех возможных значений, мы сначала придаем ему значение "пусто", а затем набираем терм за термом, пока не станет возможным отождествление. Поскольку в выражении (2) за $\underline{E}1$ непосредственно следует значащий символ + (такой символ мы называем "опорным" для $\underline{E}1$), мы можем набирать значение $\underline{E}1$ до ближайшего знака +. Оно становится ABA . Относя к $\underline{E}2$ остаток выражения (1), то есть $(B+A)C - + +$, мы делаем вывод о возможности отождествления.

Пусть отождествление происходит справа налево. Тогда $\underline{E}2$ принимает минимальное возможное значение до ближайшего справа знака +, то есть "пусто", а $\underline{E}1$ принимает значение

$$ABA + (B+A)C - +$$

Рассмотрим отождествление выражения

$$PAB + (Z) + B(AB) + A(AB + (Z) + B(AB)) \quad (3)$$

как (частного случая) выражения

$$P\underline{E}1 + \underline{S}2(\underline{E}1)\underline{E}2 \quad (4)$$

(Когда мы не указываем направления - значит оно нормальное, то есть слева направо). Символ P в (4) мы сразу сопоставляем с символом P в (3). Затем набираем значение $\underline{E}1$ до ближайшего знака +. Получаем AB . Однако, мы видим, что в (4) за знаком + идет $\underline{S}2$, а в (3) - левая скобка, следовательно, при данном значении $\underline{E}1$ отождествление невозможно. Удлиняем $\underline{E}1$ до следующего знака +. Получаем $AB + (Z)$. Свободная переменная $\underline{S}2$ принимает значение B . Скобки, охватывающие второе вхождение $\underline{E}1$ в (4),

должны быть сопоставлены скобкам, объемлющим выражение AB . Однако это выражение не совпадает со значением $\underline{E1}$. Следовательно, необходимо снова удлинить $\underline{E1}$. На этот раз мы успешно заканчиваем отождествление: $\underline{E1}$ принимает значение $AB+(Z)+B(AB)$,

$\underline{E2}$ - значение A , а $\underline{E3}$ - "пусто".

Рассмотрим более подробно случай, когда типовое выражение E_t , являющееся образцом для отождествления, содержит скобки, внутри которых заключено более одной свободной переменной выражения. В этих случаях мы не должны присваивать значения этим переменным, прежде чем не закончим присвоение значений переменным, не находящимся в скобках, иначе говоря, мы должны перескочить через скобки (сопоставив им, конечно, пару скобок в выражении E_t), закончить отождествление на основном уровне, и лишь после этого проводить отождествление внутри скобок. Если это выражение, в свою очередь, содержит внутренние скобки, к ним применяется это же правило. Свободные переменные, находящиеся на любом уровне скобочной структуры имеют у нас приоритет перед всеми переменными, находящимися на более низких (внутренних) уровнях. Различия, к которым это приводит, продемонстрируем на следующем примере.

Пусть выражение

$$P(A+A)A+A \quad (5)$$

должно быть отождествлено как

$$P(\underline{E1}+\underline{E2})\underline{E3}+\underline{E1} \quad (6)$$

Перескакивая через скобки, мы приписываем $\underline{E3}$ минимальное возможное значение, то есть A . Тогда $\underline{E1}$ принимает значение $A+A$. Входим внутрь скобок и видим, что $\underline{E2}$ должно принять значение A , и тогда отождествление будет закончено. Если бы не было приоритета более высокого скобочного уровня, мы начали бы с того, что придали $\underline{E1}$ значение A . Тогда $\underline{E2}$ приняло

бы значение $A+A$, а E^2 - значение $A+A$, и мы закончили бы отождествление с другими значениями свободных переменных.

3. РЭФАЛ-машина

3.1. РЭФАЛ-машина имеет поле зрения и поле памяти. Поле зрения - это запоминающее устройство, которое в каждый момент времени содержит некоторое рабочее выражение. Поле памяти содержит набор предложений, который может быть любым, но не может меняться в процессе работы машины.

Работа РЭФАЛ-машины состоит в преобразовании выражения, находящегося в поле зрения в соответствии с набором предложений, находящихся в поле памяти. Это преобразование мы будем называть конкретизацией выражения.

3.2. Область действия знака конкретизации \underline{K} называется выражение, заключенное между этим знаком \underline{K} и соответствующей ему точкой \cdot .

3.3. Работа РЭФАЛ-машины складывается из ряда элементарных конкретизаций, называемых шагами. Выполнение шага начинается с исследования поля зрения в поисках выражения, подлежащего конкретизации в первую очередь. Просматривая поле зрения слева направо, машина ищет первый знак \underline{K} , в области действия которого не содержится других знаков \underline{K} . Этот знак \underline{K} называется ведущим знаком конкретизации, а терм, который им начинается - ведущим термом. Если в поле зрения нет ни одного знака \underline{K} , то это значит, что конкретизация выполнена до конца и машина останавливается.

Найдя ведущий знак конкретизации, машина просматривает слева направо набор предложений, находящийся в поле памяти, ища такое предложение, что область действия ведущего знака конкретизации может быть отождествлена как его левая часть; если указатель

направления в предложении есть R , то отождествление проводится справа налево, если "пусто" — то слева направо. Найдя первое такое предложение, машина заменяет ведущий терм на правую часть предложения, в которой все свободные переменные замещаются на значения, которые были им приданы в процессе отождествления. Этой операцией замены выполнение шага заканчивается, и машина переходит к следующему шагу.

Если в поле памяти машина не может найти подходящего предложения, конкретизация считается невыполнимой, и машина останавливается.

3.4. Примеры.

Пусть в поле памяти РЭСАД-машины находится набор предложений:

$$\S 3.1. \underline{KEA} + \underline{EBI} \supseteq \underline{KEAI} + \underline{EB}_2$$

$$\S 3.2. \underline{KEA} + 0 \supseteq \underline{EA}$$

И пусть в поле зрения введено выражение:

$$\underline{KOIII} + \underline{OII}_2$$

Начиная первый шаг, машина объявит единственный знак K в поле зрения ведущим и начнет искать подходящее предложение. Таковым является § 3.1. Свободная переменная \underline{EA} примет значение $OIII$, переменная \underline{EB} — OI . В результате замены в поле зрения окажется выражение:

$$K \ OIII + \underline{OI}_2$$

На втором шаге опять будет использован § 3.1. Поле зрения примет вид: $\underline{KOIIIII} + 0_2$

Теперь § 3.1 не применим, машина станет исследовать § 3.2, который оказывается применимым. В результате в поле зрения будет стоять:

$$OIIIII$$

Здесь нет знаков K , и работа закончена.

Допустим теперь, что кроме §§ 3.1 и 3.2 в поле памяти занесены еще два предложения:

§ 3.3. KX \supseteq OIII

§ 3.4. KY \supseteq OII

а в поле зрения стоит выражение:

KX₁ + KY₂₂

Здесь в первом шаге ведущим будет знак K, стоящий перед X. §§ 3.1, 3.2 окажутся неприменимыми, и будет использован § 3.3. .

Поле зрения примет вид:

KOIII + KY₂₂

После второго шага поле зрения будет:

KOIII + OII₁

Затем машина повторит те же шаги, что и в предыдущем случае, и придет к тому же результату.

4. Машинные операции.

Если предполагать, что мы можем непрерывно следить за состоянием поля зрения РЕФАЛ-машины и, в частности, можем извлечь из него конечный результат конкретизации, то описание РЕФАЛ-машины можно считать законченным. Однако, на практике бывает чрезвычайно

полезно выводить из машины и промежуточные результаты.

При интерпретации работы РЕФАЛ-машины на ЭВМ промежуточные и конечные результаты удобно получать на внешних носителях информации ЭВМ. Для этого нужно использовать операции вывода на внешние устройства. Ясно, что эта операция не может быть описана на языке РЕФАЛ. Точно так же, часто бывает полезно использовать какую-либо операцию, описанную не на языке РЕФАЛ, а непосредственно на языке вычислительной машины, или же производимую любым другим способом. Чтобы включить эти операции в систему программирования

на РЕФАЛе, мы вводим понятие машинной операции.

4.1. Каждой машинной операции соответствует символ машинной операции. Предполагается, что РЕФАЛ-машина содержит список символов всех машинных операций, которые она умеет выполнять. Машина выполняет машинную операцию тогда, когда ведущий терм в поле зрения имеет вид:

$$\underline{KME}_.$$

где M - символ машинной операции, а E - некоторое выражение, являющееся объектом, над которым производится операция. Таким образом, с учетом наличия машинных операций, мы должны следующим образом уточнить алгоритм выполнения шага:

Найдя ведущий знак K , машина исследует, не является ли непосредственно следующий за ним символ символом машинной операции, и если это так, выполняет соответствующую машинную операцию, после чего шаг считается законченным; в противном случае машина поступает как было описано выше.

4.2. Примеры.

Введем машинную операцию печати, которую будем изображать символом 'П' и опишем следующим образом. Если ведущий терм есть $\underline{K'P'E}_.$, то машина выдает (печатает) выражение E , а ведущий терм заменяет на E

Используя операцию 'П', можно на РЕФАЛе описать операцию печати со стиранием 'ПС'. Для этого достаточно ввести два предложения;

§ 3.5. $\underline{K'PS'E}_1 \supseteq \underline{K'STERET'K'P'E}_1.$

§ 3.6. $\underline{K'STERET'E}_1 \supseteq$

Можно ввести также операцию 'СТОП', которая независимо от аргумента приводит к остановке РЕФАЛ-машины. Используя операцию 'СТОП' можно на РЕФАЛе описать операцию печати с остановом

' ПЕЧСТОП ' с помощью следующего предложения:

§3.7. К' ПЕЧСТОП' ≥ К' П' Е. И. К' СТОП'.

Число машинных операций принципиально не ограничивается. Функции, описанные первоначально на РЕФАЛе могут, если это понадобится для повышения эффективности, быть заменены позже на эквивалентные машинные операции.

5. Рекурсивные функции

Формальное определение РЕФАЛ-машины требует при каждой конкретизации просмотра подряд всех предложений до тех пор, пока не будет найдено подходящее. Поэтому и человек, пишущий на РЕФАЛе какое-либо предложение,

должен помнить об этом и

учитывать, что может сработать какое-либо вышестоящее предложение (т.е. предложение, расположенное ближе к началу набора предложений). Это, конечно, неудобно и может вызвать путаницу. Необходимо как-то разграничить сферы действия предложений, чтобы одни предложения не мешали другим, и чтобы не было необходимости просматривать все предложения. Способ разграничения сферы действия предложений напрашивается сам собой. Если левая часть предложения начинается с какого-либо значащего символа, то предложение может подойти только в том случае, если конкретизируемое выражение (так мы будем для краткости называть область действия ведущего знака конкретизации) начинается с того же символа. Этот символ мы будем называть детерминативом предложения и будем требовать, чтобы все предложения, за исключением, быть может, небольшого числа чрезвычайно общих предложений содержали детерминативы. Легко видеть, что порядок расположения в памяти предложений с различными детерминативами несущественен. Набор предложений распадается, по существу,

на ряд меньших наборов (групп), состоящих из предложений с одним и тем же детерминативом. Порядок предложений внутри группы, разумеется, весьма существенен.

Используя эту систему, мы предложения 3.1 и 3.2, определяющие сложение, перепишем в виде:

$$\S 4.1. \underline{K+EA}, \underline{EBI} \supseteq \underline{K+EA}, \underline{EB}_1$$

$$\S 4.2. \underline{K+EA}, 0 \supseteq \underline{EA}$$

Здесь запятая отделяет одно число от другого. Теперь, чтобы выполнить сложение, надо в поле зрения занести выражение вида

$$\underline{K+0III}, 0II_1$$

Усеченное вычитание мы определим следующими предложениями:

$$\S 5.1. \underline{K+EA}, \underline{EBI} \supseteq \underline{K+EA}, \underline{EB}_1$$

$$\S 5.2. \underline{K+EA}, 0 \supseteq \underline{EA}$$

$$\S 5.3. \underline{K0}, \underline{EB} \supseteq 0$$

Символы + и - являются детерминативами в группах предложений 4 и 5. Эти группы фактически независимы друг от друга, как и от групп с другими детерминативами. В процессе выполнения шага РЕФАЛ-машины мы первым делом выделяем детерминатив, т.е. первый символ, следующий за ведущим знаком \underline{K} и начинаем просматривать предложения с данным детерминативом, игнорируя предложения с другими детерминативами.

Мы видим, что каждая группа предложений определяет некоторую рекурсивную функцию, определенную на множестве выражений и принимающую значения из этого множества. При этом детерминатив служит символом функции, а остальная часть выражения - аргументом. Разница лишь в том, что вместо обычной записи $\Psi(\varepsilon)$, где ε - некоторое выражение, мы должны в поле зрения записать $\underline{K\Psi\varepsilon}_1$.

Выражение E может, в свою очередь, содержать знак конкретизации, то есть требовать вычисления. Легко видеть, что благодаря правилу нахождения ведущего знака K мы приступаем к вычислению функции лишь после того, как вычислены значения всех аргументов.

Отсутствие формального разделения между символом функции и ее аргументом является совершенно естественным при столь общем аргументе, как выражение (т.е. любая цепочка символов, сбалансированная по скобкам). В выражении, оставшемся после отщепления детерминатива можно, в свою очередь, выделить "символы функции", характеризующие различные частные случаи аргумента. Пусть, например, у нас есть группа предложений с детерминативом φ . Среди них может быть подгруппа предложений, у которых в левой части вслед за φ идет какой-либо символ, отличный от указателя переменной, например, α . Эти предложения могут работать только тогда, когда конкретизируемое выражение начинается с символа $\varphi\alpha$, таким образом, они определяют некую "подфункцию" функции φ . Подфункция $\varphi\alpha$ может также делиться на подфункции и т.д.

Теперь в рамках формального описания языка РЕФАЛ мы установим следующее соглашение, которое служит для удобства пользователя, и которое, собственно говоря и делает РЕФАЛ языком рекурсивных функций. Назовем предложения, содержащие детерминатив, (что есть левая часть которых начинается со значащего символа) предложениями типа F ; все прочие назовем предложениями типа G . И условимся, что все предложения типа G стоят ниже (позже) любого предложения типа F . Эти последние будем записывать группами и пусть в группу входят все предложения с данным детерминативом. Вследствие этого соглашения РЕФАЛ-машина, найдя ведущий знак K , за которым

следует значащий символ, обращается сначала к группе предложений с соответствующим детерминативом, и лишь не найдя подходящего предложения, обращается к предложениям типа G .

Л И Т Е Р А Т У Р А

1. В.Ф.Турчин, **Метаязык для формального описания алгоритмических языков**, сб. "Цифровая вычислительная техника в программировании", Сов. Радио, 1966.
2. В.Ф.Турчин, **Метаалгоритмический язык**, "Кибернетика", № 4, 1968.
3. В.Ф.Турчин, **Алгоритмический язык рекурсивных функций (РЕФАЛ)**. Описание языка и приемы программирования. ИИМ АН СССР, 1968 г.
4. В.Ф.Турчин, С.Н.Флоренцев, В.А.Фисун, **Использование языка РЕФАЛ для автоматизации программирования**. Доклад на межвузовской конференции по автоматизации обработки экономической информации. Москва, май 1967 г.
5. В.Ф.Турчин, **Транслятор с АЛГОЛа, написанный на языке РЕФАЛ**, Доклад, направленный на I-ую Всесоюзную конференцию по прогр. в Киеве в ноябре 1968 г.
6. С.Н.Флоренцев, В.Ю.Олинин, В.Ф.Турчин, **РЕФАЛ-интерпретатор**. Доклад, направленный на I-ую Всесоюз.конф. по прогр. в Киеве в ноябре 1968 г.
7. С.Н.Флоренцев, В.Ю.Олинин, В.Ф.Турчин, **Эффективный интерпретатор для языка "РЕФАЛ"** Препринт ИИМ АН СССР, № 29, 1969 г.
8. С.Н.Флоренцев, В.Ю.Олинин, С.А.Романенко, В.Ф.Турчин, **Описание системы программирования "РЕФАЛ" (Инструкция для пользователей)**, Препринт ИИМ АН СССР, № 30, 1969 г.
9. В.Ф.Турчин, **Программирование на языке РЕФАЛ**, ИИМ АН СССР, Препринты ИИМ АН СССР №№ 45, 46, 47, 48, 49, 1971 г.
10. Д.Карташов, **I. Стандартная программа ВВОДТ. II. Стандартная программа ВЫВОДТ**. ИИМ АН СССР, 1973 г.
11. В.Л.Ушкова. **Программа "СХЕМА" в ОС ИИМ**, ИИМ АН СССР, 1973 г.

12. Инструкция по использованию операционной системы ОС ИИМ, ИИМ АН СССР, 1970 г.
13. С.П.Бычков, В.А.Фастун, С.Н.Флоренцев, Л.К.Эйсмонт, СИМУЛА (Описание языка и инструкция по работе с компилятором) ИИМ АН СССР, 1974 г.

№ T-09856 от " 2 " VIII 1974 г. Заказ № 2707 Тираж 200 экз

Ордена Ленина институт прикладной математики
Москва, Мясницкая пл., 4