



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

**В.Ф. Турчин**

**ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ РЕФАЛ**

**V Использование метафункций в языке рефал**

**Препринт № 49 за 1971 г**

**Москва**

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

**В.Ф.ТУРЧИН**

**ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ РЕФАЛ**

**5.ИСПОЛЬЗОВАНИЕ МЕТАФУНКЦИЙ В ЯЗЫКЕ РЕФАЛ**

**Москва, 1971г.**

## I. Компилирующие метафункции

Мы называем метафункциями (или метапроцедурами) такие функции, конкретизация которых включает управление процессом конкретизации других функций. Это очень общее определение; оно станет яснее, когда мы определим по отдельности два класса, на которые распадаются метафункции: компилирующие и интерпретирующие метафункции.

Компилирующие метафункции – это функции, которые изменяют содержимое поля памяти рефал-машины и поэтому могут определить новые или переопределить старые функции (функции 'ЗК' и 'ВК', формальное определение которых включает преобразование поля памяти, мы не относим к метафункциям). Мы называем такие функции компилирующими, потому что содержимое поля памяти представляет собой программу выполнения алгоритма, и конкретизация этих функций есть, в сущности, формирование программы перед ее исполнением в соответствии с некоторой входной информацией, закодированной в виде аргумента функции. Такой именно смысл и вкладывается в программировании в понятие компиляции.

Компилирующие метафункции могут использоваться в системе программирования на рефале для таких целей как:

деление рефал-программы на части с тем, чтобы не держать в памяти описание одного сложного алгоритма в то время, когда выполняется другой;

внесение изменений в программу в процессе отладки;

использование и переработка библиотечных программ, например, ресурсных схем в сочетании с описанием синтаксиса, как это было

описано в выпуске 4, п.6.

Кроме того переопределение функций может использоваться и оперативно, как средство сделать описание алгоритма более компактным и изящным.

Формальное описание рефала содержит одну простую метафункцию с детерминативом '  $\geq$  ', которая уничтожает прежнее содержимое поля памяти и заменяет его новым, которое записано на метакоде в аргументе функции. Этой функции, в сочетании с функцией '  $\xi$  ' образующей в поле зрения копию (в метакоде) текущего содержимого поля памяти, достаточно, чтобы описать любое преобразование поля памяти (используя, конечно, возможности рефал-машины для преобразования текстов на метакоде). Записать в поле памяти любой текст дополнительно к уже существующему можно с помощью функции:

$$\S 77 \underline{K} \text{ 'запись' } \underline{E} A \geq \underline{K} \text{ ' } \geq \text{ ' } \underline{E} A \underline{K} \underline{P}$$

Эта функция рассчитана на использование в тех случаях, когда известно, что в старом поле памяти нет предложений с такими же детерминативами, какие содержатся во вновь записываемом тексте, или когда по замыслу программиста новые предложения должны быть приписаны к старым группам предложений спереди; (ясно, что если необходимо, можно определить функцию, приписывающую предложения сзади).

Если надо переопределить некоторое число функций с помощью новых предложений, а остальные функции оставить неизменными, можно воспользоваться функцией 'ОПРФ' - "определить функции", описание которой мы приводим ниже. В этом описании мы пользуемся квазипредикативной переменной с детерминативом 'ОПИСФ' - "описания функций". Функция 'ОПИСФ' выделяет все предложения с детерминативами, которые входят в указанный список, и ставит их на первые места (не изменяя, конечно,

их относительного порядка). Таким образом, непредикативность этой переменной выражается в том, что в процессе отождествления совершается перестановка предложений, но так как эта перестановка не влияет на работу машины, ее можно игнорировать и считать нашу переменную "почти предикативной".

§ 78 Выделение и отщепление описаний функций с детерминативами, указанными в списке ЕД

$$\underline{K}'\text{ОПИСФ}'(\underline{ЕД})\underline{ЕА} \cong \underline{K} \alpha \underline{ЕА}(\quad)(\underline{ЕД}) \text{ .}$$

$$\begin{aligned} \S 78A.1 \underline{K} \alpha \underline{Е}'\text{ДОДЕТ}'1 \underline{СД} \underline{Е}'\text{АРТИПЧ}'2 \underline{ЕЗ} (\underline{Е4})(\underline{Е5})(\underline{Е6}'\underline{СД} \underline{Е7}) \cong \\ \underline{K} \alpha \underline{ЕЗ} (\underline{Е4} \underline{Е1} \underline{СД} \underline{Е2})(\underline{Е5})(\underline{Е6} \underline{СД} \underline{Е7}) \text{ .} \end{aligned}$$

$$\begin{aligned} \S 78A.2 \underline{K} \alpha \underline{Е}'\text{ДОДЕТ}'1 \underline{СХ} \underline{Е}'\text{АРТИПЧ}'2 \underline{ЕЗ} (\underline{Е4})(\underline{Е5})(\underline{ЕД}) \cong \\ \underline{K} \alpha \underline{ЕЗ} (\underline{Е4})(\underline{Е5} \underline{Е1} \underline{СХ} \underline{Е2})(\underline{ЕД}) \text{ .} \end{aligned}$$

$$\S 78A.3 \underline{K} \alpha (\underline{Е4})(\underline{Е5})(\underline{ЕД}) \cong (\underline{Е4})\underline{Е5}$$

§ 79 Часть предложения до детерминатива

$$.1 \underline{K}'\text{ДОДЕТ}'\underline{VП} \underline{Е1} \underline{VK} \underline{Е2} \cong (\underline{VП} \underline{Е1} \underline{VK}) \underline{Е2}$$

$$\S .2 \underline{K}'\text{ДОДЕТ}' \cong \neg$$

§ 80 Аргумент и правая часть предложения

$$.1 \underline{K}'\text{АРТИПЧ}' \underline{Е1} \underline{VП} \underline{Е2} \cong (\underline{Е1}) \underline{VП} \underline{Е2}$$

$$\S .2 \underline{K}'\text{АРТИПЧ}' \underline{Е1} \cong (\underline{Е1})$$

§ 81 Определить функции, описания которых содержатся в аргументе ( в метакоде )

$$\underline{K}'\text{ОПРФ}' \underline{Е1} \cong \underline{K} \beta \underline{Е1} (\quad)(\quad) \text{ .}$$

§81А Функция  $\beta$  составляет список детерминативов (последняя сумка)

$$.1 \underline{K} \beta \underline{E} \text{'ДОДЕТ'} \underline{I} \underline{CД} \underline{E} \text{'АРТИПЧ'} \underline{2} \underline{E3} (\underline{E4}) (\underline{E5} \underline{CД} \underline{E6}) \cong \\ \underline{K} \beta \underline{E3} (\underline{E4} \underline{E1} \underline{CД} \underline{E2}) (\underline{E5} \underline{CД} \underline{E6}) \underline{.}$$

$$\S \quad .2 \underline{K} \beta \underline{E} \text{'ДОДЕТ'} \underline{I} \underline{CД} \underline{E} \text{'АРТИПЧ'} \underline{2} \underline{E3} (\underline{E4}) (\underline{E5}) \cong \\ \underline{K} \beta \underline{E3} (\underline{E4} \underline{E1} \underline{CД} \underline{E2}) (\underline{E5} \underline{CД}) \underline{.}$$

$$\S \quad .3 \underline{K} \beta (\underline{E4}) (\underline{E5}) \cong \underline{K} \gamma (\underline{E4}) (\underline{E5}) \underline{K} \text{'П'} \underline{.} \underline{.}$$

$$\S 81Б \underline{K} \gamma (\underline{EД}) (\underline{E5}) \underline{E} \text{'ОПИСФ'} (\underline{EД}) \underline{6} \underline{E7} \cong \underline{K} \text{'=} \underline{E5} \underline{E7} \underline{.}$$

Полезно также ввести функцию 'УБРФ' - "убрать функции"

§82 Убрать из поля памяти описания функций с детерминативами из списка  $\underline{EД}$

$$\underline{K} \text{'УБРФ'} \underline{EД} \cong \underline{K} \delta (\underline{EД}) \underline{K} \text{'П'} \underline{.} \underline{.}$$

$$\S 82А \underline{K} \delta (\underline{EД}) \underline{E} \text{'ОПИСФ'} (\underline{EД}) \underline{I} \underline{E2} \cong \underline{K} \text{'=} \underline{E2} \underline{.}$$

Простейший способ реализации компилирующих метапроцедур - это реализация одной процедуры полной перегрузки поля памяти, то-есть процедуры '='. Тогда остальные метапроцедуры можно реализовать, описав их на рефале, как показано выше. Такая реализация вполне удовлетворительна, если эти процедуры выполняются сравнительно редко, то-есть лишь при переходе от одного крупного этапа вычислений к другому. Если же переопределение одной или нескольких функций происходит систематически, как следствие использования определенного приема

программирования на рефале, то для обеспечения эффективности вычислительного процесса необходимо так построить систему программирования, чтобы метафункция 'ОПФ' могла быть реализована как машинная операция, которая выполнялась бы путем перезагрузки только тех частей поля памяти, которые содержат предложения, относящиеся к переопределяемым функциям.

Задача 6.1. Описать преобразование 'МЮБР', действующее как  $M_A^{-1}$  на метакод от символьных и объектных скобок и всех объектных знаков, иначе говоря, обладающее тем свойством, что

$$\underline{K} ' \text{мюбр} ' \underline{K} ' \text{п} ' \dots$$

дает объектное выражение, полученное из поля памяти частичным метакодовым преобразованием, не затрагивающим составных символов, круглых скобок и объектных знаков.

## 2. Интерпретирующие метафункции

Прежде чем определять понятие интерпретирующей метафункции, мы приведем пример такой функции.

Часто бывает необходимо выполнить последовательно над некоторым выражением  $\langle \mathcal{E} \rangle$  ряд процедур, например,  $\alpha, \beta, \gamma, \delta \in$ . Для этого надо ввести в поле зрения выражение

$$\underline{K} \in \underline{K} \delta \underline{K} \gamma \underline{K} \beta \underline{K} \alpha \langle \mathcal{E} \rangle \dots$$

Писать много знаков конкретизации и считать точки неудобно, поэтому введем процедуру 'ВП' - "выполнить последовательно" - с синтаксисом

$$\# \underline{K} ' \text{вп} ' ( \underline{E} ' \text{сдф} ' \text{л} ) ' \text{над} ' \underline{E} ' \text{арг} ' 2 \ni \underline{E} ' \text{рез} ' 3$$

где 'СДФ' - список детерминативов функций, 'АРГ' - аргумент, 'РЕЗ' - результат. Разделитель 'НАД' вводится для удобства обозрения.

$$\S 88.1 \quad \underline{K} \text{ 'ВП' } (\underline{E} X \underline{\subseteq} F) \text{ 'НАД' } \underline{E} A \cong$$

$$\underline{K} \text{ 'ВП' } (\underline{E} X) \text{ 'НАД' } \underline{K} \underline{\subseteq} F \underline{E} A \dots$$

$$\S 88.2 \quad \underline{K} \text{ 'ВП' } ( ) \text{ 'НАД' } \underline{E} A \cong \underline{E} A$$

Теперь последовательное выполнение процедур (горизонтальная связь) осуществляется конкретизацией

$$\underline{K} \text{ 'ВП' } (\epsilon \delta \gamma \beta \alpha) \text{ 'НАД' } \langle \mathcal{E}_0 \rangle \dots$$

Функция 'ВП' не меняет содержимого поля памяти, однако в её аргумент входят детерминативы других функций, и функция 'ВП' определяет, когда и какая функция будет конкретизироваться. Содержанием работы, выполняемой функцией 'ВП', является управление работой других функций, осуществляемое в процессе самой работы, без изменения поля памяти. Такие функции мы и называем интерпретирующими метафункциями. Формальным признаком интерпретирующей метафункции является наличие в её описании хотя бы одного предложения, в правой части которого есть хотя бы один знак  $\underline{K}$ , за которым следует либо свободная переменная, либо другой знак  $\underline{K}$ .

Приведем еще один пример интерпретирующей метафункции.

§ 84 Сквозной просмотр с применением функции  $\underline{\subseteq} F$

$$\text{.I } \underline{K} \text{ 'СКВ' } \underline{\subseteq} F \underline{E} 1 (\underline{E} 2) \underline{E} 3 \cong$$

$$\underline{K} \underline{\subseteq} F \underline{E} 1 \dots (\underline{K} \text{ 'СКВ' } \underline{\subseteq} F \underline{E} 2 \dots) \underline{K} \text{ 'СКВ' } \underline{\subseteq} F \underline{E} 3 \dots$$

$$\S .2 \underline{K} \text{ 'СКВ' } \underline{\subseteq} F \underline{E} 1 \cong \underline{K} \underline{\subseteq} F \underline{E} 1 \dots$$



Если  $\varphi$  - некоторая функция, производящая просмотр и преобразование без накопления информации, и описанная таким образом, что просмотр производится только на верхнем уровне скобочной структуры (например, § 8 или § 14), то приписывание к детерминативу  $\varphi$  слева детерминатива 'СКВ' превращает функцию  $\varphi$  в сквозную функцию, то-есть конкретизация

$$\underline{\kappa} \text{ 'СКВ' } \varphi \langle \mathcal{E} \rangle \text{ .}$$

осуществляет применение функции  $\varphi$  к выражению  $\langle \mathcal{E} \rangle$  на всех уровнях скобочной структуры.

Очень важной для практики является метафункция активизации символов конкретизации. Она служит для изменения порядка выполнения конкретизацией в поле зрения. Допустим, например, что правая часть некоторого предложения имеет вид:

$$\langle \mathcal{E}_1 \rangle \langle \mathcal{E}_2 \rangle (\langle \mathcal{E}_3 \rangle) \langle \mathcal{E}_4 \rangle \langle \mathcal{E}_5 \rangle$$

где  $\langle \mathcal{E}_i \rangle$  - какие-то выражения, и что по смыслу алгоритма необходимо, чтобы после того как правая часть окажется в поле зрения, в первую очередь выполнялись конкретизации, входящие в выражения  $\langle \mathcal{E}_2 \rangle$  и  $\langle \mathcal{E}_4 \rangle$ , а уже затем - те, которые входят в  $\langle \mathcal{E}_1 \rangle$ ,  $\langle \mathcal{E}_3 \rangle$ ,  $\langle \mathcal{E}_5 \rangle$ . Чтобы добиться этого, заменим в выражениях  $\langle \mathcal{E}_1 \rangle$ ,  $\langle \mathcal{E}_3 \rangle$ ,  $\langle \mathcal{E}_5 \rangle$  каждый знак конкретизации на символ 'к' за которым следует левая скобка, а конкретизационную точку - на правую скобку. Таким образом, все подвыражения, имеющие вид

$$\underline{\kappa} \langle \mathcal{E} \rangle \text{ .}$$

превратятся в

$$\text{'к' } (\langle \mathcal{E} \rangle)$$

Модифицированные таким образом выражения  $\langle \mathcal{E}_i \rangle$  обозначим через  $\langle \mathcal{E}'_i \rangle$ , выражения  $\langle \mathcal{E}_2 \rangle$  и  $\langle \mathcal{E}_4 \rangle$ , входящие в первую очередь конкретизации, оставим неизменными, и подвергнем всю правую часть действия процедуры АКТ

$$\underline{\kappa} \text{'АКТ'} \langle \mathcal{E}'_1 \rangle \langle \mathcal{E}_2 \rangle (\langle \mathcal{E}'_3 \rangle) \langle \mathcal{E}_4 \rangle \langle \mathcal{E}'_5 \rangle \underline{\cdot}$$

которую определим таким образом, чтобы она совершала обратное превращение всех подвыражений

$$\text{'K'} (\langle \mathcal{E} \rangle)$$

в

$$\underline{\kappa} \langle \mathcal{E} \rangle \underline{\cdot}$$

(то-есть "активизировала" бы символы 'K'):

$$\S 85.1 \quad \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 1 \text{'K'} (\underline{\varepsilon} 2) \underline{\varepsilon} 3 \underline{\cdot}$$

$$\underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 1 \underline{\cdot} \underline{\kappa} \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 2 \underline{\cdot} \underline{\cdot} \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 3 \underline{\cdot}$$

$$\S 85.2 \quad \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 1 \underline{\cdot} \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 1 \underline{\cdot}$$

$$\S 85A.1 \quad \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 1 (\underline{\varepsilon} 2) \underline{\varepsilon} 3 \underline{\cdot} \underline{\varepsilon} 1 (\underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 2 \underline{\cdot}) \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 3 \underline{\cdot}$$

$$\S 85A.2 \quad \underline{\kappa} \text{'АКТ'} \underline{\varepsilon} 1 \underline{\cdot} \underline{\varepsilon} 1$$

Легко видеть, что теперь конкретизации будут выполняться в нужном порядке.

Использование процедуры 'АКТ' позволяет организовать выполнение конкретизаций в две очереди. Программист может, если это покажется ему удобным, ввести и более сложные метапроцедуры, позволяющие более детально определить порядок конкретизации.

Задача 6.2. Описать функцию 'АКТН' активизирующую символы 'К' в соответствии с их номерами. Номер символа 'К' изображается термом, следующим за 'К'. Номера от 1 до 9 пишутся одним знаком, многозначные номера заключаются в скобки:

$$'K'(17)('A' + B 'K'3 ('SPAPCK' \underline{L} \underline{R}))$$

В первую очередь активизируются символы 'К' с меньшими номерами. В пределах одной очереди действует обычное правило определения ведущего знака К.

### 3. Интерпретируемый операторный язык

Метаязыковая природа рефала проявляется в том, что оставаясь формально в рамках рефала и пользуясь только рефал-машиной как машиной-исполнительницей, можно фактически описывать алгоритмы на любом языке. Для этого надо лишь написать на рефале транслятор-интерпретатор для используемого языка, иначе говоря описать семантику данного языка в терминах рефала. Пусть разработан некоторый язык  $\mathcal{L}$ . Сопоставим ему функцию с детерминативом ' $\mathcal{L}$ ' имеющую смысл транслятора-интерпретатора для этого языка. Функцию ' $\mathcal{L}$ ' надо описать на рефале таким образом, чтобы выполнение рефал-машиной конкретизации:

$$\underline{K} ' \mathcal{L} ' < A_{\mathcal{L}} >$$

где  $< A_{\mathcal{L}} >$  - описание алгоритма  $A$  (вместе с начальными данными) на языке  $\mathcal{L}$ , было бы в то же время выполнением алгоритма  $A$ .

В настоящем разделе мы опишем операторный язык общей направленности, не содержащий никаких специальных операторов, а фиксирующий лишь самый факт "операторности" вместе с такими неотделимыми от него понятиями как присваивание значения, переход по метке и проверка условия. В некоторых случаях такой язык может явиться полезным допол-

нением к сугубо функциональному программированию на "чистом" рефале. Мы будем называть его сокращенно ИОЯ - интерпретируемый операторный язык.

Объектами (именами), которым присваиваются значения, в ИОЯ являются идентификаторы, определяемые как обычно (алгол-60). Присваивание осуществляется с помощью закапывания, а извлечение значения - с помощью выкапывания. Таким образом, например,

$$\underline{K} \ ' \ \underline{B} \ K \ ' \ \text{зоур}$$

есть значение идентификатора `зоур`      Выполнение программы на ИОЯ сводится, в конечном счете, к изменению значений идентификаторов.

Приступим к описанию синтаксиса и семантики ИОЯ. Синтаксис будет описываться с помощью бэкусовского аппарата, расширенного путем добавления конструкции

$$\langle \text{штрихованный прямоугольник} \dots \rangle$$

(см. выпуск 2).

Семантику мы будем описывать предложениями на рефале, по мере введения соответствующих синтаксических понятий. Таким образом, к концу раздела мы получим готовый интерпретатор для ИОЯ. Пояснения, которыми будут сопровождаться формальные синтаксические и семантические определения, имеют целью не столько сделать ясным конечный результат (его нетрудно понять и без комментариев), сколько продемонстрировать методы работы на рефале, и в частности, методы, используемые при разработке нового языка; фактически, мы будем описывать одновременно процесс создания языка и процесс написания для него транслятора.

Программу на ИОЯ мы определим как совокупность операторов, разделенных точкой с запятой и, возможно, помеченных одной или несколькими метками, отделяемыми от оператора двоеточием (алгольная система). Для формального описания синтаксиса (и, как мы увидим ниже, семантики) удобно ввести понятие участка — последовательности непомеченных операторов.

$$\begin{aligned} \langle \text{программа} \rangle &::= \langle \text{участок} \rangle \langle \langle \text{метка} \rangle : \langle \text{участок} \rangle \rangle \\ \langle \text{участок} \rangle &::= \langle \langle \text{оператор} \rangle ; \dots \rangle \\ \langle \text{метка} \rangle &::= \langle \text{идентификатор} \rangle \end{aligned}$$

Из того, что участок может быть пустым, следует возможность наличия нескольких меток перед оператором. Программа, если она содержит хотя бы один оператор, должна оканчиваться на точку с запятой; это — следствие того, что мы предпочитаем однородный синтаксис, упрощающий описание семантики. Ясно, что нетрудно было бы модифицировать синтаксис, разрешив опускать в конце программы точку с запятой. Однако, нужно ли это?

Легко предвидеть, что в процессе исполнения программы на ИОЯ окажется необходимым искать указанную метку для передачи на нее управления. Чтобы не просматривать весь текст программы, было бы полезно заключить предварительно каждый участок в скобки. Тогда при поиске метки будет фактически просматриваться только список меток данной программы. Из подобных же соображений имеет смысл изменить средство разделения операторов, а именно, заключить каждый оператор в скобки, убрав знаки ";". Итак, мы приходим к мысли провести некую препроцессинг текста, прежде чем приступить к выполнению программы. Поэтому первым предложением нашего интерпретатора будет

§ 86  $\underline{\kappa}' \text{ИОЯ}' \underline{\varepsilon}' \ni \underline{\kappa}' \text{выппрог}' \underline{\kappa}' \text{ппр}' \underline{\varepsilon}' 1 \dots$

где 'ППР' - функция препроцессии, а 'ВЫППРОГ' - выполнения программы.

Программируя на рефале, всегда надо иметь перед глазами конкретные примеры обрабатываемых языковых объектов - с тем уровнем детализации, который необходим на данной стадии. Обозначая через  $Q_i$  различные операторы (относительно которых мы пока даже не знаем, как они будут выглядеть), напомним пример программы:

$Q_1 ; Q_2 ; \text{LAP} \{ Q_3 ; \text{CASE} 1 : \text{случ} 2 : Q_4 ; Q_5 ; Q_6 ;$

Как должна работать процедура 'ППР'? Отщеплять участки и заключать их в скобки. Но еще необходимо обработать каждый участок. Чтобы не совершать просмотр дважды, будем обрабатывать участок одновременно с отщеплением. Следовательно, введем квазипредикативную переменную с детерминативом 'УЧ' - участок. Чтобы сэкономить на форматном преобразовании, будем писать рекурсивную переменную в виде  $\underline{\varepsilon}' (\text{'УЧ}' ( )) \langle \alpha \rangle$ . Тогда функцию 'УЧ' можно описать как функцию, требующую обращения

$\underline{\kappa}' \text{'УЧ}' ( ) \langle \mathcal{C} \rangle \dots$

Вот это описание:

§ 87.1  $\underline{\kappa}' \text{'УЧ}' ( \underline{\varepsilon} \mathcal{Y} ) \underline{\varepsilon}' \text{ИД}' \text{М} \underline{\varepsilon}' \ni ( \underline{\varepsilon} \mathcal{Y} ) \underline{\varepsilon}' \text{М} \underline{\varepsilon}'$

§ 87.2  $\underline{\kappa}' \text{'УЧ}' ( \underline{\varepsilon} \mathcal{Y} ) \underline{\varepsilon}' 0 ; \underline{\varepsilon}' \ni \underline{\kappa}' \text{'УЧ}' ( \underline{\varepsilon} \mathcal{Y} ( \underline{\varepsilon}' 0 ) ) \underline{\varepsilon}' 1 \dots$

§ 87.3  $\underline{\kappa}' \text{'УЧ}' ( \underline{\varepsilon} \mathcal{Y} ) \ni ( \underline{\varepsilon} \mathcal{Y} )$

(Описание функции 'ИД' как и метасинтаксической переменной <ИДЕНТИФИКАТОР>, мы предполагаем уже имеющимся; см. Главу 5, п. 1.)

Опираясь на процедуру отщепления участка, опишем процедуру пре-процессии. Не будем стремиться получить сразу наиболее компактное описание. Будем сначала писать программу "в лоб", и уже потом, анализируя полученную программу, преобразовывать её, если можно, к более экономному виду. Такой метод можно рекомендовать в качестве общего метода работы на рефале. Программы на рефале, подобно формулам математики, отличаются от программ на обычных языках программирования своей относительной компактностью и высокой способностью допускать преобразования. Эти свойства следует всячески использовать. Не надо стремиться решить задачу в уме – это верный способ проглядеть какой-нибудь частный случай или допустить иную ошибку. Не надо лениться выписывать в явном виде промежуточные результаты: практика показывает, что несмотря на время, потраченное на письмо, это скорейший путь к получению конечного результата. Рефал-предложения можно выводить, как выводят формулы в математике.

Определение программы содержит в правой части два элемента, из которых второй – циклический. Так как обработка циклического элемента потребует авторекурсивной (вызывающей саму себя) функции, разделяем функции обработки первого и второго элемента, и, обозначая нужную функцию через  $\alpha$  записываем:

$$\S A \subseteq \alpha \in ('u'()) \cup E2 \cong (E\gamma) \subseteq \beta \in E2 \perp$$

$$\S B.1 \subseteq \beta \cong$$

$$\S B.2 \subseteq \beta \in 'иД' M: E ('u'()) \cup E2 \cong E M: (E\gamma) \subseteq \beta \in E2 \perp$$

Здесь левые части в точности соответствуют синтаксическим определениям, а правые выражают необходимое преобразование.





перестановочны). Остается § А.3. Заменяя в нем  $\alpha$  на  $\beta$ , видим, что он не противоречит § Б.1 и § Б.2. На какое место поставить этот параграф? Так как участок может быть, в частности, пустым, это предложение более обще, чем оба остальных, следовательно должно быть помещено в конце.

Итак, подставляя вместо  $\beta$  детерминатив 'ППР', получаем окончательно:

$$\S 88.1 \quad \underline{\kappa} \text{ 'ппр' } \cong$$

$$\S 88.2 \quad \underline{\kappa} \text{ 'ппр' } \in \text{'ид.' м: } \underline{\varepsilon} \text{ ('уч' ( )) } \underline{\varepsilon} \underline{\varepsilon} \cong \underline{\varepsilon} \text{ м: } (\underline{\varepsilon} \underline{\varepsilon}) \underline{\kappa} \text{ 'ппр' } \underline{\varepsilon} \underline{\varepsilon} \underline{\varepsilon}$$

$$\S 88.3 \quad \underline{\kappa} \text{ 'ппр' } \underline{\varepsilon} \text{ ('уч' ( )) } \underline{\varepsilon} \underline{\varepsilon} \cong (\underline{\varepsilon} \underline{\varepsilon}) \underline{\kappa} \text{ 'ппр' } \underline{\varepsilon} \underline{\varepsilon} \underline{\varepsilon}$$

Программа, которую мы приводили выше в качестве примера, примет после препроцессии вид:

$$((Q_1)(Q_2)) \text{ LABEL: } ((Q_3)) \text{ CASE 1: } ( ) \text{ СЛУЧ 2: } ((Q_4)(Q_5)(Q_6))$$

Пустые участки между следующими подряд метками не выбрасываются.

После препроцессии программа приобретает новый синтаксис, и нам следует зафиксировать его формальное определение, чтобы оно служило руководством при составлении левых частей предложений.

$$\langle \text{прог I} \rangle ::= \langle \langle \text{метка} \rangle : (\langle \text{участ I} \rangle) \dots \rangle$$

$$\left| \left( (\langle \text{оператор} \rangle) \langle \text{участ I} \rangle \right) \langle \langle \text{метка} \rangle : (\langle \text{участ I} \rangle) \dots \rangle \right.$$

$$\langle \text{участ I} \rangle ::= \langle (\langle \text{оператор} \rangle) \dots \rangle$$

Выполнение программы 'ВЫПРОГ' разбивается на выполнение отдельных участков - 'ВЫПУЧ' которое, в свою очередь, требует выполнения отдельных операторов - 'ВЫПОП' Как связать эти три функции?

Функция 'ВЫПРОГ' нуждается, очевидно, в формате, фиксирующем начало выполняемого участка. Определим этот формат так: \*)

$$\# \underline{\leq} \alpha (\langle \text{начало} \rangle) (\langle \text{участ} 1 \rangle) \langle \text{конец} \rangle \geq$$

где

$$\langle \text{конец} \rangle \quad \langle \langle \text{метка} \rangle (\langle \text{участ} 1 \rangle) \dots \rangle$$

а определение понятия  $\langle \text{начало} \rangle$  не имеет большого значения, и его можно не выписывать;  $\langle \text{участ} 1 \rangle$  - выполняемый участок.

Следующий шаг:

$$\S \text{ В } \underline{\leq} \alpha (\underline{\in} \text{Н}) (\underline{\in} \text{У}) \underline{\in} \text{К} \geq \underline{\leq} \beta (\underline{\in} \text{Н}) \underline{\leq} \text{'выпуч'} (\underline{\in} \text{У}) \underline{\in} \text{К} \underline{\leq}$$

Функция  $\beta$  должна решить, выполнять ли следующий участок, или переходить по метке. Для этого результат конкретизации функции 'ВЫПУЧ' должен содержать необходимую информацию. Поэтому определим синтаксис 'ВЫПУЧ' так:

$$\# \underline{\leq} \text{'выпуч'} (\underline{\in} \text{'участ' } 1 \text{ } \underline{\in} \text{У}) \geq (\underline{\in} \text{У}) \text{'идна'} (\langle \text{метка} \rangle) \\ | (\underline{\in} \text{У})$$

Отсюда

$$\S \text{ Г.1 } \underline{\leq} \beta (\underline{\in} \text{Н}) (\underline{\in} \text{У}) \text{'идна'} (\underline{\in} \text{М}) \underline{\in} \text{К} \geq \underline{\leq} \gamma (\underline{\in} \text{М}) \underline{\in} \text{Н} (\underline{\in} \text{У}) \underline{\in} \text{К} \underline{\leq}$$

$$\S \text{ Г.2 } \underline{\leq} \beta (\underline{\in} \text{Н}) (\underline{\in} \text{У}) \underline{\in} \text{'ид'} \text{ } \underline{\in} \text{М} \text{ } \underline{\in} \text{1} \geq \underline{\leq} \alpha (\underline{\in} \text{Н} (\underline{\in} \text{У}) \underline{\in} \text{М} \text{ } \underline{\in} \text{1}) \underline{\in} \text{1} \underline{\leq}$$

$$\S \text{ Г.3 } \underline{\leq} \beta (\underline{\in} \text{Н}) (\underline{\in} \text{У}) \geq$$

\*)

Предвидя, что для описания 'ВЫПРОГ' понадобится вертикальная последовательность функций, мы обозначаем эти последние через  $\alpha, \beta, \dots$

Формулируя предложение Г.3, мы приняли решение, как поступать, когда выполнение программы закончено: уничтожить все конкретизируемое выражение (ибо все связи между функциями здесь вертикальны). Таким образом, 'ВЫПРОГ' будет определена как глагольная функция:

$$\S D \underline{K} \gamma (EM) \underline{E1} \underline{EM} : \underline{E2} \cong \underline{K} \alpha (E1 \underline{EM} : ) \underline{E2} .$$

Функция 'ВЫПРОГ' вызывает  $\alpha$  причем надо учесть два случая, соответствующие двум вариантам правой части в определении {прогI}

$$\S E.1 \underline{K} 'выппрог' \underline{E1} \underline{ид} 'M : \underline{E1} \cong \underline{K} \alpha (EM : ) \underline{E1} .$$

$$\S E.2 \underline{K} 'выппрог' \underline{E1} \cong \underline{K} \alpha ( ) \underline{E1} .$$

(Возможность пустой программы мы исключаем, чтобы не загромождать описание).

Исключая с помощью прогонки функцию  $\alpha$  и заменяя  $\beta$  на 'ВПИ' а  $\gamma$  - на 'ПЕРЕХОД' получаем:

$$\S 89.1 \underline{K} 'выппрог' \underline{E1} \underline{ид} 'M : \underline{W} \underline{У} \underline{ЕК} \cong$$

$$\underline{K} 'ВПИ' (EM : ) \underline{K} 'выпуч' \underline{W} \underline{У} . \underline{ЕК} .$$

$$\S 89.2 \underline{K} 'выппрог' \underline{W} \underline{У} \underline{ЕК} \cong \underline{K} 'ВПИ' ( ) \underline{K} 'выпуч' \underline{W} \underline{У} . \underline{ЕК} .$$

$$\S 90.1 \underline{K} 'ВПИ' (EN) \underline{W} \underline{У} 'идна' (EM) \underline{ЕК} \cong$$

$$\underline{K} 'ПЕРЕХОД' (EM) EN \underline{W} \underline{У} \underline{ЕК} .$$

$$\S 90.2 \underline{K} 'ВПИ' (EN) \underline{W} \underline{У} \underline{E1} \underline{ид} 'M : \underline{W} \underline{2} \underline{ЕК} \cong$$

$$\underline{K} 'ВПИ' (EN \underline{W} \underline{У} EM : ) \underline{K} 'выпуч' \underline{W} \underline{2} . \underline{ЕК} .$$

$$\S 90.3 \underline{K} 'ВПИ' (EN) \underline{W} \underline{У} \cong$$

§ 91  $\underline{K}$  'ПЕРЕХОД' (EM)  $\underline{E}$  EM :  $\underline{W}$   $\underline{U}$   $\underline{E}$  K  $\underline{\equiv}$

$\underline{K}$  'ВП' (E'EM :)  $\underline{K}$  'ВЫПУЧ'  $\underline{W}$   $\underline{U}$   $\underline{E}$  K  $\underline{.}$

функцию 'ВЫПУЧ' удобно определить таким образом, чтобы она вызывала дублирование ( размножение ) оператора, который передается функции 'ВЫПОП' Тогда функцию 'ВЫПОП', описывающую семантику операторов ИОЯ, мы определим так, чтобы значением выполнения оператора, то-есть результатом конкретизации

$\underline{K}$  'ВЫПОП' <оператор>  $\underline{.}$

было либо <пусто>, когда перехода не требуется, либо 'ИДНА' ( <метка> ), что указывает на необходимость перехода. Кроме того, разумеется, эта конкретизация может вызывать изменение значений некоторых идентификаторов. Итак, от функции 'ВЫПОП' мы ожидаем синтаксиса

$\# \underline{K}$  'ВЫПОП' <оператор>  $\underline{\equiv}$  'ИДНА' ( <метка> ) | <пусто>  
 функции 'ВЫПУЧ' в процессе её конкретизации придадим формат

$\# \underline{K}$  'ВЫПУЧ' <выполненные опер.> ( <невыполненные опер.> )  $\underline{\equiv}$

§ 92 Выполнение участка

.I Случай, когда предыдущий оператор дал указание о переходе по метке:

$\underline{K}$  'ВЫПУЧ' 'ИДНА'  $\underline{W}$  M  $\underline{E}$  B (EN)  $\underline{\equiv}$  (E' B EN) 'ИДНА'  $\underline{W}$  M

§ 92.2 Выполнение следующего оператора:

$\underline{K}$  'ВЫПУЧ'  $\underline{E}$  B ( (EO) EN )  $\underline{\equiv}$

$\underline{K}$  'ВЫПУЧ'  $\underline{K}$  'ВЫПОП'  $\underline{E}$  O  $\underline{.}$   $\underline{E}$  B (EO) (EN)  $\underline{.}$

## § 92.3 Конец участка

$$\underline{\leq} \text{'выпуч'} \underline{\in} \mathbb{B} ( ) \underline{\geq} ( \underline{\in} \mathbb{B} )$$

Синтаксис операторов ИОЯ таков:

<оператор>            <пусто> | <оператор перехода>  
                           | <оператор присваивания> | <усл. оператор>

<оператор перехода>            'ИДНА' <метка>

<оператор присваивания>        <левч>        <правч>

<левч>                    < <эллевч> ... >

<эллевч> ::= <идентификатор> | \* | ( < <эллевч> ... > )

<правч>                    < <элправч> ... >

<элправч> ::= <функция> | <идентификатор> | ,  
                           | ( . <рефал-выражение> ) | \* | ( < <элправч> ... > )

<функция>                <составной символ>

<усл. оператор>            'ЕСЛИ' <условие> 'ТО' ( <оператор> )  
                               'ЕСЛИ' <условие> 'ТО' ( <оператор> ) 'ИНАЧЕ' ( <оператор> )

<условие>                <правч>

Семантика достаточно ясно описывается следующими рефал-предложениями.

## § 93.1 Пустой оператор

$$\underline{\leq} \text{'выпоп'} \underline{\geq}$$

§ 93.2 Оператор перехода

$$\underline{K} \text{ 'выпол' 'идна' } \in M \cong \text{'идна'} (\in M)$$

§ 93.3 Условный оператор. Полный вариант. Условие  $\in \mathcal{U}$  сохраняется при обращении к функции 'ПРОВУСЛ' чтобы после проверки условия (по его значению) можно было бы восстановить значения всех идентификаторов, входящих в  $\in \mathcal{U}$

$$\underline{K} \text{ 'выпол' 'если' } \in \mathcal{U} \text{ 'то' } \underline{w1} \text{ 'иначе' } \underline{w2} \cong \\ \underline{K} \text{ 'провусл' } (\in \mathcal{U}) (\underline{K} \text{ 'знач' } \in \mathcal{U} \text{ } \underline{w1} \text{ } \underline{w2} \text{ } \underline{.}$$

§ 93.4 Условный оператор. Сокращенный вариант

$$\underline{K} \text{ 'выпол' 'если' } \in \mathcal{U} \text{ 'то' } \underline{w1} \cong \\ \underline{K} \text{ 'провусл' } (\in \mathcal{U}) (\underline{K} \text{ 'знач' } \in \mathcal{U} \text{ } \underline{w1} \text{ } ( ) \text{ } \underline{.}$$

§ 93.5 Оператор присваивания

$$\underline{K} \text{ 'выпол' } \in L = \in P \cong \underline{K} \text{ 'присв' } (\in L) \underline{K} \text{ 'знач' } \in P \text{ } \underline{.}$$

§ 94 Вычисление значения правой части (или условия)

.I Область действия составного символа (переменная  $\underline{S}$  'СОСТ' Ф) в качестве детерминатива функции простирается до ближайшей звездочки

$$\underline{K} \text{ 'знач' } \underline{S} \text{ 'СОСТ' } \Phi \in A * \in Z \cong \\ \underline{K} \underline{S} \Phi \underline{K} \text{ 'знач' } \in A \text{ } \underline{.} \text{ } \underline{K} \text{ 'знач' } \in Z \text{ } \underline{.}$$

§ 94.2 а если звездочки нет, то - до конца подвыражения:

$$\underline{K} \text{ 'знач' } \underline{S} \text{ 'СОСТ' } \Phi \in A \cong \underline{K} \underline{S} \Phi \underline{K} \text{ 'знач' } \in A$$

§ 94.3

$$\underline{K} \text{ 'знач' } \in \text{'ид' и } \in Z \cong \underline{K} \text{ 'вк' } \in \text{'ид' и } \underline{.} \underline{K} \text{ 'знач' } \in Z \text{ } \underline{.}$$

§ 94.4 Запятая служит только для разделения идентификаторов

$$\underline{K}^{\text{'знач'}}$$
,  $\underline{E}1 \supseteq \underline{K}^{\text{'знач'}}$   $\underline{E}1 \_$

§ 94.5 Аппарат для изображения готовых значений ("констант", в отличие от "переменных", изображаемых идентификаторами)

$$\underline{K}^{\text{'знач'}}$$
 ( $\_ \underline{E}C$ )  $\underline{E}2 \supseteq \underline{E}C$   $\underline{K}^{\text{'знач'}}$   $\underline{E}2 \_$

§ 94.6 Остальные скобки остаются на своих местах

$$\underline{K}^{\text{'знач'}}$$
 ( $\underline{E}1$ )  $\underline{E}2 \supseteq (\underline{K}^{\text{'знач'}}$   $\underline{E}1 \_)$   $\underline{K}^{\text{'знач'}}$   $\underline{E}2 \_$

§ 94.7  $\underline{K}^{\text{'знач'}}$   $\supseteq$

§ 95 Присваивание идентификаторам левой части значений, которые берутся из соответствующих подвыражений значения правой части

$$\_I \underline{K}^{\text{'присв'}}$$
 ( $(\underline{E}1)\underline{E}2$ ) ( $\underline{E}A$ )  $\underline{E}B \supseteq$

$$\underline{K}^{\text{'присв'}}$$
 ( $\underline{E}1$ )  $\underline{E}A \_ \underline{K}^{\text{'присв'}}$  ( $\underline{E}2$ )  $\underline{E}B \_$

§ 95.2  $\underline{K}^{\text{'присв'}}$  ( $\underline{E}1$  ( $\underline{E}2$ ))  $\underline{E}A$  ( $\underline{E}B$ )  $\supseteq$

$$\underline{K}^{\text{'присв'}}$$
 ( $\underline{E}2$ )  $\underline{E}B \_ \underline{K}^{\text{'присв'}}$  ( $\underline{E}1$ )  $\underline{E}A \_$

§ 95.3 На каждом уровне скобочной структуры в левой части должно быть не более одного идентификатора или звездочки, иначе было бы неясно, как делить между ними значение правой части

$$\underline{K}^{\text{'присв'}}$$
 ( $\underline{E}'\text{ид}'$  и)  $\underline{E}P \supseteq \underline{K}^{\text{'зк'}}$   $\underline{E}и = \underline{E}P \_$

§ 95.4 Звездочка означает уничтожение соответствующего подвыражения

$$\underline{K}^{\text{'присв'}}$$
 ( $\_*$ )  $\underline{E}A \supseteq$

§ 96 Проверка условия

$$\cdot I \underline{K}' \text{'ПРОВУСЛ'} (EY) ((EZ)) (E') (E2) \cong$$

$$\underline{K}' \text{'ПРИСВ'} (\underline{K}' \text{'СКЕЛЕТ'} EY \cdot) EZ \cdot \underline{K}' \text{'ВЫПОП'} E1 \cdot$$

§ 96.2  $\underline{K}' \text{'ПРОВУСЛ'} (EY) (\neg EZ) (E') (E2) \cong$

$$\underline{K}' \text{'ПРИСВ'} (\underline{K}' \text{'СКЕЛЕТ'} EY \cdot) EZ \cdot \underline{K}' \text{'ВЫПОП'} E2 \cdot$$

§ 97 Уничтожаются детерминативы функций, входящие в условия, а константы заменяются на звездочку. Таким образом, остается "идентификаторный скелет" условия. Запятые и звездочки не должны входить в условие.

$$\cdot I \underline{K}' \text{'СКЕЛЕТ'} \subseteq \text{'СОСТ'} \Phi E1 \cong \underline{K}' \text{'СКЕЛЕТ'} E1 \cdot$$

§ 97.2  $\underline{K}' \text{'СКЕЛЕТ'} (\cdot E1) E2 \cong * \underline{K}' \text{'СКЕЛЕТ'} E2 \cdot$

§ 97.3  $\underline{K}' \text{'СКЕЛЕТ'} E' \text{'ИД'} \text{'И'} E2 \cong E \text{'И'} \underline{K}' \text{'СКЕЛЕТ'} E2 \cdot$

§ 97.4  $\underline{K}' \text{'СКЕЛЕТ'} (E1) E2 \cong (\underline{K}' \text{'СКЕЛЕТ'} E1 \cdot) \underline{K}' \text{'СКЕЛЕТ'} E2 \cdot$

§ 97.5  $\underline{K}' \text{'СКЕЛЕТ'} \cong$

На этом формальное описание ИОЯ заканчивается.

Сделаем несколько замечаний по семантике ИОЯ.

В типичном случае, идентификатор, входящий в правую часть оператора присваивания, входит и в его левую часть. Это означает, что значение идентификатора каким-то образом преобразуется. Если идентификатор не входит в правую часть, но входит в левую, то он получает новое значение, но старое не забывается, ибо происходит закапывание без предварительного выкапывания. Такие операторы увеличивают стэк значений идентификатора. Если идентификатор входит в правую часть,



но не входит в левую, то происходит выкапывание без закапывания, вследствие чего последнее значение идентификатора уничтожается и восстанавливается предпоследнее в стеке значений. Таким образом, пользователю предоставляются все те возможности, которые в алголе-60 обеспечиваются блочной структурой памяти. Оператор

$x1$       (.ДЯДЯ-СТЕПА)

означает, выражаясь языком алгола-60, что происходит вход в блок, где описан идентификатор  $x1$ , и присвоение ему значения "ДЯДЯ-СТЕПА". Оператор

$* = x1$

означает выход из этого блока. Если мы напишем

$x1 = y1$

то это, очевидно, отнюдь не равнозначно алгольному оператору, присваивания

$x1 = y1$

Чтобы смоделировать этот последний, мы должны воспользоваться функцией дублирования

§ 98  $\leq 'A' \in \in \in \in \in$

Нужный оператор ИОЯ можно записать, например, так:

$* (x1)(y1) = x1 'A' (y1)$

Сложный вид оператора вскрывает неэлементарность алгольного присваивания. Мы уже обсуждали подобные вопросы в связи с понятием размножения переменных (Глава 4 п. 5).

Разумеется, ничто не мешало бы нам дополнительно ввести в ИОЯ оператор алгольного присваивания, изображаемого знаком  $=$ , который автоматически переводился бы в нужную конструкцию с "нашим" присваиванием. Однако для символьных задач алгольное присваивание не обладает большой ценностью — из-за того, что требует дублирования значений

переменных. Именно "наше" присваивание обеспечивает простую и эффективную запись в наиболее типичных случаях. Например, чтобы слить два списка - СПИС1 и СПИС2 в один список - СПИС1, нам достаточно записать

СПИС1      СПИС1, СПИС2

Здесь происходит именно слияние, то-есть дублирование отсутствует, и идентификатор СПИС2 теряет свое прежнее значение. Если бы мы пользовались алгольным присваиванием, пришлось бы дополнительно описывать процесс освобождения памяти из -под значения идентификатора СПИС2, не говоря уж о потере времени на бессмысленное дублирование.

Оператор, отщепляющий от текста с именем ПРОГ идентификатор, и присваивающий его имени ПЕРВИДЕНТ, записывается так:

(ПЕРВИДЕНТ) ПРОГ      'ид' ПРОГ

Заметим, что если ПРОГ не начинается с идентификатора, рефал-машина испытает аварийную остановку вследствие невозможности выполнить присваивание (функция 'ПРИСВ' § 95).

В условных операторах используются стандартные рефал-предикаты. Например, введя предикат равенства

§ 99.1  $\underline{K}$  'РАВ' (E1)(E1)  $\equiv$  ((E1)(E1))

§ 99.2  $\underline{K}$  'РАВ' (E1)(E2)  $\equiv$   $\neg$  (E1)(E2)

мы можем написать условный оператор:

'ЕСЛИ' 'РАВ' (x1)(y1) ТО' ('ИДНА' M1)

При определении предикатов и обращении к ним надо соблюдать известную осторожность, чтобы получался правильный "идентификаторный скелет". Этого всегда можно достичь с помощью введения скобок.

Логические связки введем следующим образом:

$$\S 100.1 \quad \underline{\kappa} \text{ 'и не' } (\underline{\varepsilon 1}) \supseteq \neg \underline{\varepsilon 1}$$

$$\S 100.2 \quad \underline{\kappa} \text{ 'и не' } \neg \underline{\varepsilon 1} \supseteq (\underline{\varepsilon 1})$$

$$\S 101.1 \quad \underline{\kappa} \text{ 'или' } ((\underline{\varepsilon 1}))((\underline{\varepsilon 2})) \supseteq ((\underline{\varepsilon 1})(\underline{\varepsilon 2}))$$

$$\S 101.2 \quad \underline{\kappa} \text{ 'или' } ((\underline{\varepsilon 1}))(\neg \underline{\varepsilon 2}) \supseteq ((\underline{\varepsilon 1})(\underline{\varepsilon 2}))$$

$$\S 101.3 \quad \underline{\kappa} \text{ 'или' } (\neg \underline{\varepsilon 1})(\underline{\varepsilon 2}) \supseteq ((\underline{\varepsilon 1})(\underline{\varepsilon 2}))$$

$$\S 101.4 \quad \underline{\kappa} \text{ 'или' } (\neg \underline{\varepsilon 1})(\neg \underline{\varepsilon 2}) \supseteq \neg (\underline{\varepsilon 1})(\underline{\varepsilon 2})$$

$$\S 102.1 \quad \underline{\kappa} \text{ 'и' } ((\underline{\varepsilon 1}))((\underline{\varepsilon 2})) \supseteq ((\underline{\varepsilon 1})(\underline{\varepsilon 2}))$$

$$\S 102.2 \quad \underline{\kappa} \text{ 'и' } ((\underline{\varepsilon 1}))(\neg \underline{\varepsilon 2}) \supseteq \neg (\underline{\varepsilon 1})(\underline{\varepsilon 2})$$

$$\S 102.3 \quad \underline{\kappa} \text{ 'и' } (\neg \underline{\varepsilon 1})(\underline{\varepsilon 2}) \supseteq \neg (\underline{\varepsilon 1})(\underline{\varepsilon 2})$$

$$\S 102.4 \quad \underline{\kappa} \text{ 'и' } (\neg \underline{\varepsilon 1})(\neg \underline{\varepsilon 2}) \supseteq \neg (\underline{\varepsilon 1})(\underline{\varepsilon 2})$$

Теперь можно использовать такие условия, как

$$\text{'или' ('РАВ'(x1)(x2))('и не' 'РАВ'(z1)((.Дядя-степа)))}$$

и любые конструкции, построенные из предикатов и связок аналогичным образом. Нетрудно расширить ИОЯ, разрешив использовать логические выражения в обычной форме и включив в препроцессию перевод их в конструкции указанного типа. Предоставляем это сделать читателю в качестве упражнения.

Использовать ИОЯ для несложных действий над небольшими объектами — значит сильно снизить эффективность программы из-за потерь времени на интерпретацию (хотя они могут быть существенно уменьшены путем прогонки). Фундаментальные преобразования объектов удобнее и практичнее описывать с помощью рекурсивных функций. Интерпретируемый опера-

торный язык может быть иногда удобным для использования на высшем уровне иерархии понятий: для перемещения больших массивов информации и регулирования процесса применения к ним сложных преобразований. В таких случаях относительная затрата времени на интерпретацию оказывается небольшой.

### Задачи

- 6.3. Написать на ИОЯ программу, выполняющую ту же работу, что функция  $\varphi$  §§ I7 (стр.3/9) . Два аргумента закопаны под именами АРГ1 и АРГ2. Результирующий список закопать под именем СПИС. Сравнить с §§ I7.
- 6.4. Решить задачу 6.2, используя функцию 'ИОЯ' для описания функций 'АКТН' и 'УЧЕТ' . Описания функций 'ОПМИН' и 'АКТМ' сохранить.

## РЕШЕНИЯ ЗАДАЧ

I.1.

$$\S \text{ I.1 } \mathbb{K} \times \mathbb{E}a, 0 \cong 0$$

$$\S \text{ I.2 } \mathbb{K} \times \mathbb{E}a, \mathbb{E}b \cong \mathbb{K} + \mathbb{K} \times \mathbb{E}a, \mathbb{E}b, \mathbb{E}a$$

I.2.

$$\S \mathbb{K} + 0 \mathbb{E}a, 0 \mathbb{E}b \cong 0 \mathbb{E}a \mathbb{E}b$$

I.3.

$$\S \text{ I.1 } \mathbb{K} \dot{+} 0, \mathbb{E}a \cong \mathbb{E}a$$

$$\S \text{ I.2 } \mathbb{K} \dot{+} \mathbb{E}a, 0 \cong \mathbb{E}a$$

$$\S \text{ I.3 } \mathbb{K} \dot{+} \mathbb{E}a \mathbb{E}b \cong \mathbb{K} \dot{+} \mathbb{E}a, \mathbb{E}b$$

2.1.

$$A) \underline{K} \varphi \underline{E}1 + \underline{E}2 (\underline{W}3 A (* *) \underline{E}1 B C) \equiv$$

$$B) \underline{K} \alpha \underline{E}A (\underline{E}1) (\underline{W}2) (\underline{Z}) (\underline{S}3) \equiv$$

$$B) \underline{K} \alpha \underline{E}A (\underline{W}K \underline{E}1) (\underline{W}L \underline{W}2) (X Z) (\underline{S}M \underline{S}3) \equiv$$

$$Г) \underline{K} 'BP' (\underline{E}1 + \underline{E}2 + \underline{E}3) \underline{E}1 + \underline{E}2 (\underline{E}3) \equiv$$

Переменная № 9 - открытая

$$Д) \underline{K} \varphi \underline{E}A (\underline{E}1) (\underline{W}'KC'2) (A) (\underline{S}('CY'14)3) \equiv$$

$$E) \underline{K} \varphi \underline{E}A (\underline{E}1) (\underline{W}('KC' \underline{E}1)2) (A) (\underline{S}(\beta \underline{E}A)3) \equiv$$

$$E) \underline{K} 'Phi' \underline{E}1 (\underline{E}7 \underline{E}'CиЛ'6 \underline{G} \underline{G}) \underline{E}2 \equiv$$

Переменные № 2 и № 8 - открытые

$$B) \underline{K} \alpha \underline{E}'A'1 \underline{W}2 + \underline{E}3 + \underline{E}(\beta \underline{E}3)A \underline{E}B \equiv$$

Переменная № 5 - открытая

2.2.

- А) Разные указатели переменной при одинаковых идентификаторах.
- Б) Вхождение переменной  $\underline{S} A$  в спецификатор рекурсивной переменной  $\underline{W}1$  не является повторным.
- В) Ошибок нет.
- Г) Ошибок нет.
- Д) Повторное вхождение переменной  $\underline{E}A$  имеет непустой спецификатор.

Е) То же с переменной ЕI

Ж) Ошибок нет.

### 2.3.

А) В правую часть входит переменная EA, которой нет в левой части.

Б) Переменная в правой части имеет непустой спецификатор.

В) Синтаксических ошибок нет.

### 2.4.

Нет: рекурсивная переменная §2 может, вообще говоря, принять любое значение.

### 2.5.

Второй параграф гарантирует, что подходящее предложение всегда будет найдено. Первый параграф может оказаться подходящим не более одного раза.

### 2.6.

Первый параграф, содержащий рекурсию, воспроизводит структуру

$$\underline{K} \alpha (< \underline{E}_1 >) < \underline{E}_2 >$$

причем выражение  $< \underline{E}_1 >$  каждый раз становится короче на один терм. Следовательно, он может оказаться подходящим лишь конечное число раз.

### 2.7.

Пусть задана схема нормального алгоритма. Не меняя порядка формул в схеме, преобразуем каждую простую формулу вида

$$< A > \rightarrow < B >$$

в предложение

$$\S \underline{K} \varphi \underline{E}_1 < A > \underline{E}_2 \Rightarrow \underline{K} \varphi \underline{E}_1 < B > \underline{E}_2 .$$

в каждую заключительную формулу

$$\langle A \rangle \rightarrow \langle B \rangle$$

- в предложении

$$\S \text{ K } \varphi \text{ E1 } \langle A \rangle \text{ E2 } \supset \text{ E1 } \langle B \rangle \text{ E2}$$

В конце допишем еще одно предложение:

$$\S \text{ K } \varphi \text{ E1 } \supset \text{ E1}$$

Применение заданного нормального алгоритма к слову  $\langle c \rangle$  эквивалентно конкретизации

$$\text{K } \varphi \langle c \rangle \text{ .}$$

3.1.

$$\underline{\vee} \text{ П } \underline{4} \text{ Д. } \underline{1} \underline{\vee} \text{ K } \underline{Q} \text{ ПЕРВСИМ } \underline{Q} \underline{\vee} \text{ S1 } \underline{\vee} \text{ E2 } \underline{\vee} \supset \underline{\vee} \text{ S1}$$

$$\underline{\vee} \text{ П } \underline{4} \text{ Д. } \underline{2} \underline{\vee} \text{ K } \underline{Q} \text{ ПЕРВСИМ } \underline{Q} \underline{\vee} \text{ E1 } \underline{R} \underline{\vee} \text{ E2 } \underline{\vee} \supset$$

$$\underline{\vee} \text{ K } \underline{Q} \text{ ПЕРВСИМ } \underline{Q} \underline{\vee} \text{ E1 } \underline{\vee}$$

$$\underline{\vee} \text{ П } \underline{4} \text{ Д. } \underline{3} \underline{\vee} \text{ K } \underline{Q} \text{ ПЕРВСИМ } \underline{Q} \underline{\vee} \supset$$

3.2.

$$\underline{\text{K}} \text{ 'ПЕРВСИМ' } (A B C) \text{ .}$$

3.3.

$$\underline{\vee} \underline{\vee} \underline{\vee} \text{ E1}$$

3.4.

$$2n+1$$

3.5.

$$\S \text{ K } \varphi \text{ E1 } \underline{Q} \text{ СИЛ } \underline{Q} \underline{\vee} A + B \underline{R} + \underline{\vee} \underline{\vee} \underline{\vee} \underline{Q} \text{ K } \underline{Y} \underline{\vee} \underline{Q} \underline{4} \underline{\vee} \underline{R} \underline{E2} \supset \text{ E1 } \text{ E2}$$





4.1.

$$\S 1.1 \underline{K} \psi (\underline{E1} \underline{WA} \underline{EA} \underline{E2}) \underline{EA} \underline{EB} \cong \underline{E1} \underline{WA} / \underline{K} \psi \underline{WA} \underline{E2} .$$

$$\S 1.2 \underline{K} \psi \underline{E1} \quad \underline{E1}$$

4.2.

$$\underline{K} \psi ((\ ) \underline{1} (x+y) \underline{B}) ((\ ) \underline{A} \underline{B} \underline{B} \underline{C} \underline{C} \underline{D} \underline{D}) .$$

4.3.

В первом формате:

$$\S 1.1 \underline{K} \psi (\underline{E1} \underline{WA} \underline{E2}) \underline{WA} \underline{EB} \cong \underline{K} \psi (\underline{E1} \underline{WA} \underline{E2} \underline{K} \alpha \underline{WA} \underline{.}) \underline{EB} .$$

$$\S 1.2 \underline{K} \psi (\underline{E1}) \underline{WA} \underline{EB} \cong \underline{K} \psi (\underline{E1} \underline{K} \beta \underline{WA} \underline{.}) \underline{EB} .$$

$$\S 1.3 \underline{K} \psi (\underline{E1}) \cong \underline{E1}$$

Сравнивая процесс отождествления при двух форматах:

$$\begin{array}{cccccccc} 1 & 2 & 6 & 7 & 8 & 3 & 4 & 5 \\ \underline{K} \psi (\underline{E1} \underline{WA} \underline{E2}) \underline{WA} \underline{EB} \cong \end{array}$$

$$\begin{array}{cccccccc} 1 & 2 & 5 & 4 & 6 & 7 & 8 & 3 \\ \underline{K} \psi \underline{WA} \underline{EB} (\underline{E1} \underline{WA} \underline{E2}) \cong \end{array}$$

находим, что сколько-нибудь существенных отличий нет.

4.4.

$$\S \underline{K} \psi (\underline{E1} \underline{SA} \underline{E2}) \underline{E3} * \underline{E4} \underline{SA} \underline{E5} * \underline{E6} \cong * \underline{E4} \underline{SA} \underline{E5} *$$

$$\S \underline{K} \psi \underline{E3} * \underline{E4} \underline{SA} \underline{E5} * \underline{E6} (\underline{E1} \underline{SA} \underline{E2}) \cong * \underline{E1} \underline{SA} \underline{E5} *$$

Эффективный алгоритм:

$$\S 1 \underline{K} \psi (\underline{EC}) \underline{E3} * \underline{E4} \cong \underline{K} \alpha ((\ ) \underline{EC}) (\underline{E3}) * \underline{E1} .$$

$$\S 2.1 \quad \underline{K} \alpha ((E1) \subseteq A E2) (E3) * E4 \subseteq A E5 \supseteq$$

$$\underline{K} \beta ((E1) \subseteq A E2) (E3) * (E4 \subseteq A) E5 \supseteq$$

$$\S 2.2 \quad \underline{K} \alpha ((E1) \subseteq A E2) E3 \supseteq \underline{K} \alpha ((E1 \subseteq A) E2) E3 \supseteq$$

$$\S 3.1 \quad \underline{K} \beta ((E1) \subseteq A E2) (E3) * (E4 \subseteq A) E5 * E6 \quad * E4 \subseteq A E5 *$$

$$\S 3.2 \quad \underline{K} \beta ((E1) \subseteq A E2) (E3) * (E4 \subseteq A) E5 \supseteq$$

$$\underline{K} \alpha ((E1 \subseteq A E2) (E3) * E4 \subseteq A E5 \supseteq$$

Краткое и эффективное описание, использующее рекурсивные переменные - см. задачу 5.5.

#### 4.5.

$$\S 1 \quad \underline{K} 'иск \neq \underline{W} 1 E2 \supseteq \underline{K} \alpha \underline{W} 1 ( ) E2 \supseteq$$

$$\S 2.1 \quad \underline{K} \alpha (E1 A) (EX) \neq E2 \supseteq A EX \neq$$

$$\S 2.2 \quad \underline{K} \alpha \underline{W} 1 (EX) \underline{W} 2 E3 \supseteq \underline{K} \alpha \underline{W} 1 (EX \underline{W} 2) E3 \supseteq$$

$$\S 2.3 \quad \underline{K} \alpha \underline{W} 1 \underline{W} X \supseteq 'убы'$$

#### 4.6.

$$\S 1 \quad \underline{K} \psi E1 \supseteq \underline{K} \psi E1 ( ) \supseteq$$

$$\S 2.1 \quad \underline{K} \psi E1 \underline{W} A \underline{W} A (E2) \supseteq \underline{W} A \underline{W} A$$

$$\S 2.2 \quad \underline{K} \psi E1 \underline{W} B (E2) \supseteq \underline{K} \psi E1 (\underline{W} B E2) \supseteq$$

$$\S 2.3 \quad \underline{K} \psi (E2) \supseteq$$

#### 4.7.

$$\S 24A.2 \quad \underline{K} \alpha \subseteq A \subseteq B E1 \supseteq \neg \subseteq A \subseteq B \underline{K} '3K' \text{АЛФАВИТ} = E1 \supseteq$$

4.8.

$$\S \underline{K} 'u' ('AA') ('AA') \approx 'AA'$$

$$\S \underline{K} 'u' ('AA') ('HET') \approx 'HET'$$

$$\S \underline{K} 'u' ('AA') (\underline{E1}) \quad \underline{K} \underline{E1} \perp$$

$$\S \underline{K} 'u' ('HET') (\underline{E1}) \approx 'HET'$$

$$\S \underline{K} 'u' (\underline{E1}) \underline{W2} \approx \underline{K} (\underline{K} \underline{E1} \perp) \underline{W2}$$

4.9.

$$\S 1.1 \underline{K} \varnothing \underline{W1} \underline{W2} \underline{W3} \underline{E4} \approx (\underline{W1} \underline{W2} \underline{W3} \underline{E4})$$

$$\S 1.2 \underline{K} \varnothing \underline{E1} \approx \neg \underline{E1}$$

4.10.

$$\S 1 \underline{K} 'cumm' \underline{EX} \approx \underline{K} \alpha () \underline{EX} ()$$

$$\S 2.1 \underline{K} \alpha (\underline{E1}) \underline{SA} \underline{EX} \underline{SA} (\underline{E2}) \quad \underline{K} \alpha (\underline{E1} \underline{SA}) \underline{EX} (\underline{SA} \underline{E2}) \perp$$

$$\S 2.2 \underline{K} \alpha (\underline{E1}) (\underline{E2}) \approx (\underline{E1} \underline{E2})$$

$$\S 2.3 \underline{K} \alpha (\underline{E1}) \underline{SX} (\underline{E2}) \approx \underline{E1} \underline{SX} \underline{E2}$$

$$\S 2.4 \underline{K} \alpha (\underline{E1}) \underline{EX} (\underline{E2}) \approx \neg \underline{E1} \underline{EX} \underline{E2}$$

4.11.

$$\S 1.1 \underline{K} 'BCTAB' (\underline{E1} \underline{SA} \underline{B}) \underline{SA} \approx \underline{K} 'BCT1' (\underline{E1}) \underline{K} 'nop' \underline{SA} \underline{B} \underline{SA} \perp$$

$$\S 1.2 \underline{K} 'BCTAB' () \underline{SA} \approx \underline{SA}$$

$$\S \quad \underline{K} 'BCT1' (\underline{E1}) (\underline{SA} \underline{B} \underline{SA}) \approx \underline{E1} \underline{SA} \underline{B} \underline{SA}$$

$$\S 1.2 \underline{K} 'BCTAB' (\underline{E1}) \underline{SA} \approx \underline{SA} \underline{SA} \underline{B}$$

4.12.

Q открытыми переменными:

$$\S 1.1 \underline{K}' \text{ликппр}' \underline{E1} \sqcup \underline{E2} \ni \underline{K}' \text{ликппр}' \underline{E1} \sqcup \underline{K}' \text{ликппр}' \underline{E2} \sqcup$$

$$\S 1.2 \underline{K}' \text{ликппр}' \underline{E1} (\underline{E2}) \underline{E3} \ni \underline{E1} (\underline{K}' \text{ликппр}' \underline{E2} \sqcup) \underline{K}' \text{ликппр}' \underline{E3} \sqcup$$

$$\S 1.3 \underline{K}' \text{ликппр}' \underline{E1} \ni \underline{E1}$$

Без открытых переменных:

$$\S 2.1 \underline{K}' \text{ликппр}' \sqcup \underline{E1} \ni \underline{K}' \text{ликппр}' \sqcup \underline{E1} \sqcup$$

$$\S 2.2 \underline{K}' \text{ликппр}' (\underline{E1}) \underline{E2} \ni (\underline{K}' \text{ликппр}' \underline{E1} \sqcup) \underline{K}' \text{ликппр}' \underline{E2} \sqcup$$

$$\S 2.3 \underline{K}' \text{ликппр}' \sqcup \underline{E2} \ni \sqcup \underline{K}' \text{ликппр}' \underline{E2} \sqcup$$

$$\S 2.4 \underline{K}' \text{ликппр}' \sqcup$$

4.13.

Функцию  $\varphi$  определенную предложениями  $\S X$ , можно описать следующим образом:

$$\S \underline{K} \varphi \underline{E1} \ni \underline{K} \beta \underline{K} \alpha_n \quad \underline{K} \alpha_2 \underline{K} \alpha_1 (\underline{E1}) \therefore \quad \therefore$$

$$\S \mathcal{Y}1.1 \underline{K} \alpha_1 (\langle \mathcal{E}_1 \rangle) \ni * \langle \mathcal{E}'_1 \rangle$$

$$\S \mathcal{Y}1.2 \underline{K} \alpha_1 \underline{E1} \ni \underline{E1}$$

$$\S \mathcal{Y}2.1 \underline{K} \alpha_2 (\langle \mathcal{E}_2 \rangle) \ni * \langle \mathcal{E}'_2 \rangle$$

$$\S \mathcal{Y}2.2 \underline{K} \alpha_2 \underline{E1} \ni \underline{E1}$$

$$\S \text{ Ун1 } \underline{\subseteq} \alpha_n (\langle \mathcal{E}_n \rangle) \supseteq * \langle \mathcal{E}'_n \rangle$$

$$\S \underline{\subseteq} \beta * \underline{\mathbb{E}1} \supseteq \underline{\mathbb{E}1}$$

4.14.

Между § 45.2 и § 45.3 вставить предложение

$$\S \underline{\subseteq} \text{'СПАР1'} ((\underline{\mathbb{E}M})) \underline{\mathbb{R}} \underline{\mathbb{E}1} \supseteq$$

$\underline{\subseteq}$ 'ВЫВ' ОШИБКА: НЕПАРНАЯ ПРАВАЯ СКОБКА  $(\underline{\mathbb{E}M}) \underline{\mathbb{R}} \underline{\mathbb{E}1}$ .

После § 45.4 добавить предложение

$$\S \underline{\subseteq} \text{'СПАР1'} (\underline{\mathbb{E}M}) \supseteq$$

$\underline{\subseteq}$ 'ВЫВ' ОШИБКА: НЕПАРНЫЕ ЛЕВЫЕ СКОБКИ  $\underline{\mathbb{E}M}$ .

5.1.

$$\S \underline{\subseteq} \text{'НЕБЦ'} \underline{\subseteq} \text{'БУКВА'} \underline{\mathbb{B}} \supseteq \neg \underline{\subseteq} \underline{\mathbb{B}}$$

$$\S \underline{\subseteq} \text{'НЕБЦ'} \underline{\subseteq} \text{'ЦИФРА'} \underline{\mathbb{C}} \supseteq \neg \underline{\subseteq} \underline{\mathbb{C}}$$

$$\S \underline{\subseteq} \text{'НЕБЦ'} \underline{\mathbb{W}A} \supseteq (\underline{\mathbb{W}A})$$

$$\S \underline{\subseteq} \text{'ДОБОР'} (\underline{\subseteq}A) \underline{\mathbb{E}1} \underline{\mathbb{W}} \text{'НЕБЦ'} \underline{\mathbb{B}} \underline{\mathbb{E}2} \supseteq (\underline{\subseteq}A \underline{\mathbb{E}1}) \underline{\mathbb{W}} \underline{\mathbb{B}} \underline{\mathbb{E}2}$$

$$\S \underline{\subseteq} \text{'ДОБОР'} (\underline{\subseteq}A) \underline{\mathbb{E}1} \supseteq (\underline{\subseteq}A \underline{\mathbb{E}1})$$

5.2.

$$\S 1.1 \underline{\subseteq} \text{'ЧИСЛО'} \underline{\mathbb{E}} \text{'ЗН'} \underline{\mathbb{1}} \underline{\mathbb{E}} \text{'ЧИСБЗ'} \underline{\mathbb{2}} \underline{\mathbb{E}X} \supseteq (\underline{\mathbb{E}} \underline{\mathbb{1}} \underline{\mathbb{E}} \underline{\mathbb{2}}) \underline{\mathbb{E}X}$$

$$\S 1.2 \underline{\subseteq} \text{'ЧИСЛО'} \underline{\mathbb{E}X} \supseteq \neg \underline{\mathbb{E}X}$$

$$\S 2.1 \underline{\subseteq} \text{'ЗН'} \underline{\mathbb{+}} \underline{\mathbb{E}1} \supseteq (\underline{\mathbb{+}}) \underline{\mathbb{E}1}$$

$$\S 2.2 \underline{\subseteq} \text{'ЗН'} \underline{\mathbb{-}} \underline{\mathbb{E}1} \supseteq (\underline{\mathbb{-}}) \underline{\mathbb{E}1}$$

$$\S 2.3 \underline{\subseteq} \text{'ЗН'} \underline{\mathbb{()}} \underline{\mathbb{E}1} \supseteq (\underline{\mathbb{()}}) \underline{\mathbb{E}1}$$

$$\S 3.1 \quad \underline{K} ' \text{чисбз}' \underline{E} ' \Delta E C Y ' 1 \quad \text{to} \quad \underline{E} ' 3 H ' 2 \quad \underline{E} ' \text{целбз}' 3 \underline{E} 4 \geq (\underline{E} 1 \text{ to } \underline{E} 2 \underline{E} 3) \underline{E} 4$$

$$\S 3.2 \quad \underline{K} ' \text{чисбз}' \underline{E} ' \Delta E C Y ' 1 \underline{E} 2 \geq (\underline{E} 1) \underline{E} 2$$

$$\S 3.3 \quad \underline{K} ' \text{чисбз}' \quad \text{to} \quad \underline{E} ' 3 H ' 1 \underline{E} ' \text{целбз}' 2 \underline{E} 3 \geq (\text{to } \underline{E} 1 \underline{E} 2) \underline{E} 3$$

$$\S 3.4 \quad \underline{K} ' \text{чисбз}' \underline{E} 1 \geq \neg \underline{E} 1$$

$$\S 4.1 \quad \underline{K} ' \Delta E C Y ' \underline{E} ' \text{целбз}' 1 \quad \underline{E} ' \text{целбз}' 2 \underline{E} 3 \geq (\underline{E} 1, \underline{E} 2) \underline{E} 3$$

$$\S 4.2 \quad \underline{K} ' \Delta E C Y ' \underline{E} ' \text{целбз}' 1 \underline{E} 2 \geq (\underline{E} 1) \underline{E} 2$$

$$\S 4.3 \quad \underline{K} ' \Delta E C Y ' \underline{E} ' \text{целбз}' 1 \underline{E} 2 \geq (\underline{E} 1) \underline{E} 2$$

$$\S 4.4 \quad \underline{K} ' \Delta E C Y ' \underline{E} 1 \geq \neg \underline{E} 1$$

$$\S 5.1 \quad \underline{K} ' \text{целбз}' \underline{S} ' \text{цифра}' 1 \underline{E} 2 \geq \underline{K} ' \Delta O B Y ' (\underline{S} 1) \underline{E} 2 \perp$$

$$\S 5.2 \quad \underline{K} ' \text{целбз}' \underline{E} 2 \geq \neg \underline{E} 2$$

$$\S 6.1 \quad \underline{K} ' \Delta O B Y ' (\underline{E} 1) \underline{S} ' \text{цифра}' 2 \underline{E} 3 \geq \underline{K} ' \Delta O B Y ' (\underline{E} 1 \underline{S} 2) \underline{E} 3 \perp$$

$$\S 6.2 \quad \underline{K} ' \Delta O B Y ' (\underline{E} 1) \underline{E} 2 \geq (\underline{E} 1) \underline{E} 2$$

### 5.3.

$$\S \underline{K} ' \text{пол}' \underline{E} ( ' R ' (+ - ) ) 1 \underline{S} A \underline{E} 2 \geq \underline{S} A \underline{K} ' \text{пол}' \underline{E} 1 \perp \underline{K} ' \text{пол}' \underline{E} 2 \perp$$

$$\S \underline{K} ' \text{пол}' \underline{E} ( ' R ' (x / ) ) 1 \underline{S} M \underline{E} 2 \geq \underline{S} M \underline{K} ' \text{пол}' \underline{E} 1 \perp \underline{K} ' \text{пол}' \underline{E} 2 \perp$$

$$\S \underline{K} ' \text{пол}' \underline{E} ( ' R ' ( \uparrow ) ) 1 \underline{S} P \underline{E} 2 \geq \underline{S} P \underline{K} ' \text{пол}' \underline{E} 1 \perp \underline{K} ' \text{пол}' \underline{E} 2 \perp$$

$$\S \underline{K} ' \text{пол}' (\underline{E} 1) \geq \underline{K} ' \text{пол}' \underline{E} 1 \perp$$

$$\S \underline{K} ' \text{пол}' \underline{S} X \geq \underline{S} X$$

5.4.

$$\S 1.1 \quad \underline{K} 'ЭЛМН' \underline{W} A \underline{E} 1 \supseteq (\underline{W} A) \underline{E} 1$$

$$\S 1.2 \quad \underline{K} 'ЭЛМН' \supseteq \neg$$

$$\S 2.1 \quad \underline{K} 'ПРЕОБР' (\underline{E} 1 \underline{W} A \underline{E} 2) \underline{W} A \supseteq (\underline{E} 1 \underline{W} A \underline{W} A \underline{E} 2)$$

$$\S 2.2 \quad \underline{K} 'ПРЕОБР' (\underline{E} 1) \underline{W} A \supseteq (\underline{E} 1 \underline{W} A)$$

Вводя сокращения 'ЭЛМН'  $\varepsilon$ , 'ПРЕОБР'  $\tau$ , производим устранение рекурсивных переменных.

$$\S 3 \quad \underline{K} \psi (\underline{E} R) \underline{E} 1 \supseteq \underline{K} \alpha (\underline{E} R) \underline{K} \varepsilon \underline{E} 1 \cdot \cdot$$

$$\S 4.1 \quad \underline{K} \alpha (\underline{E} R) (\underline{E} A) \underline{E} 1 \supseteq \underline{K} \psi \underline{K} \tau (\underline{E} R) \underline{E} A \cdot \underline{E} 1 \cdot$$

$$\S 4.2 \quad \underline{K} \alpha (\underline{E} R) \neg \underline{E} 1 \supseteq \underline{K} \beta (\underline{E} R) \underline{E} 1 \cdot$$

$$\S 5 \quad \underline{K} \beta (\underline{E} R) \supseteq \underline{E} R$$

Прогонка  $\S 3$  дает два предложения:

$$\S 3.1 \quad \underline{K} \psi (\underline{E} R) \underline{W} A \underline{E} 1 \supseteq \underline{K} \alpha (\underline{E} R) (\underline{W} A) \underline{E} 1 \cdot \supseteq \\ \underline{K} \psi \underline{K} \tau (\underline{E} R) \underline{W} A \cdot \underline{E} 1 \cdot$$

$$\S 3.2 \quad \underline{K} \psi (\underline{E} R) \supseteq \underline{K} \alpha (\underline{E} R) \neg \cdot \supseteq \underline{K} \beta (\underline{E} R) \cdot \supseteq \underline{E} R$$

Прогоняем  $\S 3.1$ :

$$\S 3.1.1 \quad \underline{K} \psi (\underline{E} X \underline{W} A \underline{E} Y) \underline{W} A \underline{E} 1 \supseteq \underline{K} \psi (\underline{E} X \underline{W} A \underline{W} A \underline{E} Y) \underline{E} 1 \cdot$$

$$\S 3.1.2 \quad \underline{K} \psi (\underline{E} X) \underline{W} A \underline{E} 1 \supseteq \underline{K} \psi (\underline{E} X \underline{W} A) \underline{E} 1 \cdot$$



Окончательно получаем:

$$\S 6 \quad \underline{\kappa} \psi \underline{E1} \supseteq \underline{\kappa} \psi ( ) \underline{E1} .$$

$$\S 7.1 \quad \underline{\kappa} \psi (\underline{EX} \underline{WA} \underline{EY}) \underline{WA} \underline{E1} \supseteq \underline{\kappa} \psi (\underline{EX} \underline{WA} \underline{WA} \underline{EY}) \underline{E1} .$$

$$\S 7.2 \quad \underline{\kappa} \psi (\underline{EX}) \underline{WA} \underline{E1} \supseteq \underline{\kappa} \psi (\underline{EX} \underline{WA}) \underline{E1} .$$

$$\S 7.3 \quad \underline{\kappa} \psi (\underline{EX}) \supseteq \underline{EX}$$

### 5.5.

$$\S \quad \underline{\kappa} \psi (\underline{E1} \underline{\subseteq} \underline{A} \underline{E2}) \underline{E} ('A_0' *) \underline{3} * \underline{E} ('A_0' \underline{\subseteq} \underline{A}) \underline{4} \underline{\subseteq} \underline{A}$$

$$\underline{E} ('A_0' *) \underline{5} \underline{E} \underline{6} \supseteq * \underline{E} \underline{4} \underline{\subseteq} \underline{A} \underline{E} \underline{5} *$$

### 5.6.

§ 1 Начальное присваивание пустого значения открытой переменной  $\underline{E1}$ :

$$\underline{\kappa} \psi \underline{EX} \supseteq \underline{\kappa} \alpha ( ) \underline{EX} .$$

§ 2 Попытка отщепления  $\underline{EA}$ :

$$\underline{\kappa} \alpha (\underline{E1}) \underline{E2} \supseteq \underline{\kappa} \beta (\underline{E1}) \underline{\kappa} 'UNT' \underline{E2} .$$

§ 3.1 Успешное отщепление (см. предл. 75.1)

$$\underline{\kappa} \beta (\underline{E1}) (\underline{EA}) \underline{E2} \supseteq$$

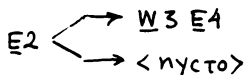
§ 3.2 Удлинение  $\underline{E1}$  при безуспешном отщеплении

$$\underline{\kappa} \beta (\underline{E1}) \neg \underline{W2} \underline{E3} \supseteq \underline{\kappa} \alpha (\underline{E1} \underline{W} \underline{2}) \underline{E3} .$$

§ 3.3 Удлинение невозможно (см. предл. 75.2)

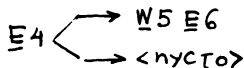
$$\underline{\kappa} \beta (\underline{E1}) \neg \supseteq \underline{E1}$$

Этим устранение рекурсивных переменных закончено. Теперь произведем оптимизацию программы путем прогонки (с учетом §§ 76.1-2). Так как из функций  $\varphi$ ,  $\alpha$ ,  $\beta$  рекурсивно вызывается только функция  $\alpha$  прогонять надо предложение, описывающее функцию  $\alpha$



$$\begin{aligned} \S 2.1 \quad \underline{K} \alpha (E1) W3 E4 &\supseteq \underline{K} \beta (E1) \underline{K}' \text{УНТ}' W3 E4 \perp \supseteq \\ &\underline{K} \beta (E1) \neg E4 \perp \end{aligned}$$

$$\S 2.2 \quad \underline{K} \alpha (E1) \supseteq \underline{K} \beta (E1) \underline{K}' \text{УНТ}' \perp \supseteq \underline{K} \beta (E1) \neg \perp \supseteq E1$$



$$\begin{aligned} \S 2.1.1 \quad \underline{K} \alpha (E1) W3 W5 E6 &\supseteq \underline{K} \beta (E1) \neg W5 E6 \perp \supseteq \\ &\underline{K} \alpha (E1 W5) E6 \perp \end{aligned}$$

$$\S 2.1.2 \quad \underline{K} \alpha (E1) W3 \supseteq \underline{K} \beta (E1) \neg \perp \supseteq E1$$

Итак, в результате получаем:

$$\S 2.1 \quad \underline{K} \alpha (EX) W1 W2 EY \supseteq \underline{K} \alpha (EX W2) EY \perp.$$

$$\S 2.2 \quad \underline{K} \alpha (EX) W1 \supseteq EX$$

$$2.3 \quad \underline{K} \alpha (EX) \supseteq EX$$

### 6.1.

$$\S 1 \quad \underline{K}' \text{МЮБР}' E1 \supseteq \underline{K}' \text{СКВ}' \text{'ВОЗ}' \underline{K}' \text{СПАРСК}' \underline{K}' \text{ВСС}' E1$$

§ 2 Восстановление составных символов

$$.1 \quad \underline{K}' \text{ВСС}' E1 \underline{Q}' EC \underline{Q}' E2 \supseteq$$

$\mathbb{E}1 \ \underline{K}' \supseteq \underline{V} \ \underline{P} \ \underline{V} \ \underline{K} \ \underline{Q} \ \underline{F} \ \underline{H} \ \underline{Q} \ \underline{V} \ \supseteq \ \underline{Q} \ \underline{E} \ \underline{C} \ \underline{Q} \ \underline{K}' \ \underline{P}' \ \underline{\cdot} \ \underline{\cdot}$   
 $\underline{K}' \ \underline{F} \ \underline{H}' \ \underline{\cdot} \ \underline{K}' \ \underline{U} \ \underline{B} \ \underline{R} \ \underline{P} \ \underline{P}' \ \underline{K}' \ \underline{P}' \ \underline{\cdot} \ \underline{\cdot} \ \underline{K}' \ \underline{V} \ \underline{C} \ \underline{C}' \ \underline{E} \ \underline{2} \ \underline{\cdot}$

§ 2.2  $\underline{K}' \ \underline{V} \ \underline{C} \ \underline{C}' \ \underline{E} \ \underline{1} \ \supseteq \ \underline{E} \ \underline{1}$

§ 3 Убрать первое предложение

$\underline{K}' \ \underline{U} \ \underline{B} \ \underline{R} \ \underline{P} \ \underline{P}' \ \underline{V} \ \underline{P} \ \underline{E} \ \underline{1} \ \underline{V} \ \underline{P} \ \underline{E} \ \underline{2} \ \supseteq \ \underline{K}' \ \supseteq \ \underline{V} \ \underline{P} \ \underline{E} \ \underline{2} \ \underline{\cdot}$

§ 4 Восстановление объектных знаков

.I  $\underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{V} \ \underline{Q} \ \underline{E} \ \underline{1} \ \supseteq \ \underline{Q} \ \underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{E} \ \underline{1} \ \underline{\cdot}$

§ 4.2  $\underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{V} \ \underline{L} \ \underline{E} \ \underline{1} \ \supseteq \ \underline{L} \ \underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{E} \ \underline{1} \ \underline{\cdot}$

§ 4.3  $\underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{V} \ \underline{R} \ \underline{E} \ \underline{1} \ \supseteq \ \underline{R} \ \underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{E} \ \underline{1} \ \underline{\cdot}$

§ 4.4  $\underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{V} \ \underline{V} \ \underline{E} \ \underline{1} \ \supseteq \ \underline{V} \ \underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{E} \ \underline{1} \ \underline{\cdot}$

§ 4.5  $\underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{\subseteq} \ \underline{A} \ \underline{E} \ \underline{1} \ \supseteq \ \underline{\subseteq} \ \underline{A} \ \underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \underline{E} \ \underline{1} \ \underline{\cdot}$

§ 4.6  $\underline{K}' \ \underline{V} \ \underline{O} \ \underline{3}' \ \supseteq$

## 6.2.

§ I  $\underline{K}' \ \underline{A} \ \underline{K} \ \underline{T} \ \underline{H}' \ \underline{E} \ \underline{1} \ \supseteq \ \underline{K}' \ \underline{A} \ \underline{K} \ \underline{T} \ \underline{H}' \ \underline{K}' \ \underline{3} \ \underline{K}' \ \underline{M} \ \underline{I} \ \underline{N} \ \underline{N} \ \underline{O} \ \underline{M} \ \underline{=} \ \underline{*} \ \underline{\cdot}$   
 $\underline{K}' \ \underline{O} \ \underline{P} \ \underline{R} \ \underline{M} \ \underline{I} \ \underline{N}' \ \underline{E} \ \underline{1} \ \underline{\cdot} \ (\underline{K}' \ \underline{V} \ \underline{K}' \ \underline{M} \ \underline{I} \ \underline{N} \ \underline{O} \ \underline{M} \ \underline{\cdot}) \ \underline{\cdot}$

§ 2 Определение минимального номера (закапывается под именем МИННОМ)

.I  $\underline{K}' \ \underline{O} \ \underline{P} \ \underline{R} \ \underline{M} \ \underline{I} \ \underline{N}' \ \underline{E} \ \underline{1} \ \underline{K}' \ \underline{W} \ \underline{H} \ \underline{E} \ \underline{2} \ \supseteq$

$\underline{K} \ \underline{\alpha} \ \underline{E} \ \underline{1} \ \underline{\cdot} \ \underline{K}' \ \underline{W} \ \underline{H} \ \underline{K}' \ \underline{U} \ \underline{C} \ \underline{H} \ \underline{E} \ \underline{T}' \ \underline{W} \ \underline{H} \ \underline{\cdot} \ \underline{K}' \ \underline{O} \ \underline{P} \ \underline{R} \ \underline{M} \ \underline{I} \ \underline{N}' \ \underline{E} \ \underline{2} \ \underline{\cdot}$

§ 2.2  $\underline{K}' \ \underline{O} \ \underline{P} \ \underline{R} \ \underline{M} \ \underline{I} \ \underline{N}' \ \underline{E} \ \underline{1} \ \supseteq \ \underline{K} \ \underline{\alpha} \ \underline{E} \ \underline{1} \ \underline{\cdot}$

§ 3.I  $\underline{K} \ \underline{\alpha} \ \underline{E} \ \underline{1} \ (\underline{E} \ \underline{2}) \ \underline{E} \ \underline{3} \ \supseteq \ \underline{E} \ \underline{1} \ (\underline{K}' \ \underline{O} \ \underline{P} \ \underline{R} \ \underline{M} \ \underline{I} \ \underline{N}' \ \underline{E} \ \underline{2} \ \underline{\cdot}) \ \underline{K} \ \underline{\alpha} \ \underline{E} \ \underline{3} \ \underline{\cdot}$

§ 3.2  $\underline{K} \times \underline{E1} \supseteq \underline{E1}$

§ 4.1  $\underline{K}' \text{УЧЕТ}' (\underline{EH}) \supseteq \underline{K}' \text{ЗК}' \text{МИННОМ} =$

$\underline{K}' \text{МИН}' \underline{EH}, \underline{K}' \text{ВК}' \text{МИННОМ} \dots$

§ 4.2  $\underline{K}' \text{УЧЕТ}' \underline{EH} \supseteq \underline{K}' \text{УЧЕТ}' (\underline{EH}) \dots$

§ 5.1  $\underline{K}' \text{МИН}' \text{УСЛ}' (- \underline{EP}) \underline{E1}, \underline{E2} \supseteq \underline{E1}$

§ 5.2  $\underline{K}' \text{МИН}' \text{УСЛ}' (\underline{EP}) \underline{E1}, \underline{E2} \supseteq \underline{E2}$

§ 5.3  $\underline{K}' \text{МИН}' \underline{EH}, * \supseteq \underline{EH}$

§ 5.4  $\underline{K}' \text{МИН}' \underline{EA} \supseteq \underline{K}' \text{МИН}' \text{УСЛ}' (\underline{K}' \text{ВУСЛ}' - \underline{EA} \dots) \underline{EA} \dots$

§ 6.1  $\underline{K}' \text{АКТН}' \underline{E1} (*) \supseteq \underline{E1}$

§ 6.2  $\underline{K}' \text{АКТН}' \underline{E1} (\underline{EH}) \supseteq \underline{K}' \text{АКТН}' \underline{K}' \text{АКТМ}' \underline{EH} \underline{E1} \dots$

§ 6.3  $\underline{K}' \text{АКТН}' \underline{E1} (\underline{EH}) \supseteq \underline{K}' \text{АКТН}' \underline{K}' \text{АКТМ}' (\underline{EH}) \underline{E1} \dots$

§ 7 Активизация символов 'K' с минимальным номером.

• 1  $\underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} \underline{K}' \underline{WH} (\underline{E2}) \underline{E3} \supseteq$

$\underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} \dots \underline{K}' \text{АКТМ}' \underline{WH} \underline{E2} \dots \underline{K}' \text{АКТМ}' \underline{WH} \underline{E3} \dots$

§ 7.2  $\underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} \supseteq \underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} \dots$

§ 8.1  $\underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} (\underline{E2}) \underline{E3} \supseteq \underline{E1} (\underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} \dots)$

$\underline{K}' \text{АКТМ}' \underline{WH} \underline{E3} \dots$

§ 8.2  $\underline{K}' \text{АКТМ}' \underline{WH} \underline{E1} \supseteq \underline{E1}$

6.3.

Текст на ИОЯ:

```

ГОТ1 = (.); ГОТ2 (.); СПИС = (.);
М1: 'ЕСЛИ' 'ИЛИ' ('РАВ' (АРГ1)(.)) ('РАВ' (АРГ2)(.))
      'ТО' ('ИДНА' КОНЕЦ);
      (Т1) АРГ1 = 'ОТЦГ' АРГ1;
      (Т2) АРГ2 'ОТЦГ' АРГ2;
      'ЕСЛИ' 'НЕ' 'РАВ' (Т1)(Т2) 'ТО' ('ИДНА' М2);
      (Т3)(Т1) 'Д' (Т1);
      СПИС СПИС, Т3;
М2: ГОТ1 ГОТ1, Т1; ГОТ2 = ГОТ2, Т2;
      'ИДНА' М1;
КОНЕЦ: АРГ1 = ГОТ1, АРГ1;
      АРГ2 = ГОТ2, АРГ2;

```

Используются функции "отцененные термины"

$$\xi \in \text{'ОТЦГ'} W1 \in E2 \Rightarrow (W1) \in E2$$

6.4.

§1  $\underline{K}$ 'АКТН'  $\underline{E1} \geq \underline{K}$ 'ноя' АРГ ( $\cdot \underline{E1}$ );

$M1$ : минном  $*$ );

АРГ = 'опрмин' АРГ ;

'ЕСЛИ' 'РАВ' (МИННОМ) ( $(\cdot *)$ ) 'ТО' ('ИДНА'  $M2$  ;

ТНОМ 'ТЕРМ' МИННОМ;

АРГ = 'АКТМ' ТНОМ, АРГ ;

'ИДНА'  $M1$ ;

$M2$  : ;  $\underline{\cdot}$   $\underline{K}$ 'ок' АРГ  $\underline{\cdot}$

§ 2.1  $\underline{K}$ 'УЧЕТ'  $\underline{\Omega}H \geq \underline{K}$ 'УЧЕТ' ( $\underline{\Omega}H$ )  $\underline{\cdot}$

§ 2.2  $\underline{K}$ 'УЧЕТ' ( $\underline{E}H$ )  $\geq \underline{K}$ 'ноя' АРГ = ( $\cdot \underline{E}H$ );

'ЕСЛИ' 'или' ('РАВ' (МИННОМ) ( $(\cdot *)$ ))

('МЕНЬШЕ' (АРГ) (МИННОМ))

'ТО' (МИННОМ  $*$ ) = АРГ (МИННОМ) ;  $\underline{\cdot}$

Кроме функций 'ОПРМИН' и 'АКТМ', описания которых не меняются, используются следующие функции:

§ 3.1  $\underline{K}$  'ТЕРМ'  $\underline{\leq} H \supseteq \underline{\leq} H$

§ 3.2  $\underline{K}$  'ТЕРМ'  $\underline{E}H \supseteq (\underline{E}H)$

§ 4  $\underline{K}$  'МЕНЬШЕ'  $(\underline{E}1)(\underline{E}2) \supseteq \underline{K}$  'МЕНЬШЕ'  $\underline{K}$  'БОЛЬШЕ'  $-\underline{E}1, \underline{E}2 \perp$   
 $(\underline{E}1)(\underline{E}2) \perp$

§ 5.1  $\underline{K}$  'МЕНЬШЕ'  $-\underline{E}P \underline{W}1 \underline{W}2 \supseteq (\underline{W}1 \underline{W}2)$

§ 5.2  $\underline{K}$  'МЕНЬШЕ'  $\underline{E}P \underline{W}1 \underline{W}2 \supseteq \neg \underline{W}1 \underline{W}2$

## ЛИТЕРАТУРА

1. В.Ф.Турчин, Метаязык для формального описания алгоритмических языков, в сб. "Цифровая вычислительная техника и программирование", Сов.Радио, 1966, стр.116.
2. В.Ф.Турчин, Метаалгоритмический язык, "Кибернетика", № 4, 1968.
3. В.Ф.Турчин, В.И.Сердобольский, Язык РЕФАЛ и его использование для преобразования алгебраических выражений, "Кибернетика" № 3, 1969, стр.58-62
4. С.Н.Флоренцев, В.Ю.Олюнин, В.Ф.Турчин, РЕФАЛ-интерпретатор, Труды I-й всесоюзной конф. по программированию, Киев, 1968.
5. В.Ф.Турчин, Транслятор с АЛГОЛа, написанный на языке РЕФАЛ, Труды I-ой всесоюзной конференции по программированию, Киев, 1968.
6. В.Ф.Турчин, Алгоритмический язык рекурсивных функций (РЕФАЛ), Препринт, ИПМ АН СССР, 1968.
7. С.Н.Флоренцев, В.Ю.Олюнин, В.Ф.Турчин, Эффективный интерпретатор для языка РЕФАЛ, Препринт ИПМ АН СССР, 1969.
8. С.А.Романенко, В.Ф.Турчин, РЕФАЛ-компилятор, Труды 2-ой всесоюзной конференции по программированию, Новосибирск, 1970.
9. А.П.Будник, Е.В.Гай, А.В.Климов, Н.С.Работнов, В.Ф.Турчин, И.Б.Щенков, Волновые функции и матрицы операторов феноменологической коллективной модели ядра, "Ядерная физика", 1971 г., (в печати) .



## ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

к следующим выпускам, посвященным языку РЕФАЛ:

1. Неформальное введение в программирование на языке РЕФАЛ.
2. Формальное описание и принципы реализации рефала.
3. Программирование на базисном рефале.
4. Использование рекурсивных переменных в языке РЕФАЛ.
5. Использование метафункций на языке РЕФАЛ.

абстрактный рефал 2/3

'АДЦЮП' (знак аддитивной операции) 1/49, 4/3

'АКТ' (активизация символов 'К') 5/10

'АКТН' (активизация нумерованных символов 'К') 5/11

алгоритмическая полнота рефала 2/17

базисная свободная переменная 1/44, 2/10

зависший рефал 3/3

ведущий знак конкретизации 1/27, 2/5

вертикальное соединение функций 3/39

'ВК' (см. выкапывание)

'ВП' (выполнить последовательно) 5/7

'ВЧСЛ' (вычислить) 2/53

'ВЧЦ' (вычисления над целыми числами) 2/55

'ВЫВ' (вывести на печать) 2/49

'ВЫЦАЖ' (выделение цепочки А...Ж) 3/12, 3/14

вызов функции 3/38, 4/19

выкапывание 2/49

выражение 1/22, 1/25, 2/4

    глагольное 3/37

    глагольно-именное 3/37

    именное 3/37

    конкретизируемое 2/7

    объектное 2/5

    рабочее 2/5, 2/7

    типовое 1/43, 2/5

главное вхождение свободной переменной 2/10

глагольно-именные выражения и функции 3/37

- глагольные выражения и функции 3/36  
 горизонтальное соединение функций 3/39  
 граф вызовов 3/38  
 группа предложений I/35  
 дерево языковых объектов I/I9  
 детерминатив I/35  
     , ограничения при реализации 2/3I  
 закапывание 2/49  
 закрытая переменная I/53, 2/II  
 'ЗАПИСЬ' (в поле памяти) 5/4  
 звено 2/I9  
 'ЗК' см. закапывание  
 знак I/I7, I/25, 2/3  
 знак конкретизации I/I6  
 значение свободной переменной 2/I0  
 'ИД' (идентификатор в АЛГОЛ'е-60) 4/8, 4/20  
 идентификатор (в АЛГОЛ'е-60) I/46, 4/6, 4/8, 4/I9  
 идентификатор свободной переменной I/30, 2/4  
 иерархия понятий в естественных языках I/I4  
     в формальных языках I/5, I/I7  
     функций 3/39  
     языков в сравнении с иерархией макрокоманд или проце-  
         дур I/3, I/4, I/7  
 именные выражения и функции 3/37  
 интерпретация и компиляция при описании семантики языка I/II  
     при трансляции 2/I8  
 интерпретируемый операторный язык (ИОЯ) 5/II  
 интерпретирующая метафункция 5/7  
 'ИОЯ' (интерпретируемый операторный язык) 5/I4  
 квазипредикативная переменная 4/45, 5/5  
 квазипредложение 4/30  
 комментарий к предложению I/26, 2/6  
 компилирующая метафункция 5/3  
 конкретизационная точка I/I6  
 конкретизационные (функциональные) скобки I/I6, 2/4  
 конкретизация I/I4, I/I5, I/26  
 конкретизируемое выражение 2/7  
 'ЛИКППР' (ликвидация повторных пробелов) 3/4, 3/30

- машинные процедуры 2/46
- метафункция 5/3
- метакод-А 2/32
  - варианты 2/39
  - , таблица 2/37
- метакод-Б 2/43
  - , таблица 2/45
- метакодовое преобразование I/54, 2/7, 2/36
- метаязык программирования I/3, I/10, I/13
- метаязыковые по отношению к языку рефал объекты I/35
- набор предложений 2/6, 2/7
- натуральный код 2/34, 2/41
- непредикативные переменные 4/43
- нумерация предложений I/27, I/35
- область действия знака конкретизации I/28, 2/5
- область определения функции 2/16
- объектная цепочка 2/4
- объектные (обычные) скобки I/22, 2/4
- объектный знак I/17, 2/4
  - , представление в машине 2/30
- однопробная функция 3/41
- операторные языки I/23
- 'ОПИСФ' (описания функций) 5/5
- опорный узел рекурсивной переменной выражения 4/11
- 'ОПРФ' (определить функции) 5/5
- оптимизация программы путем использования процедур ЗК, ВК 3/18
  - использования скобок в качестве разделителей 3/6
  - прогонки 4/34, 4/45
  - расщепления функций 3/43
  - устранения просмотров высших размерностей 3/12, 4/43
  - устранения рекурсивных переменных 4/16, 4/34
- остановка рефал-машины нормальная 2/7
  - аварийная 2/7, 2/8, 2/13
- открытая переменная I/53, 2/14
- отождествление см. синтаксическое отождествление
- параграф I/25, 2/6
- 'ПЕРВСИМ' (первый символ выражения) I/31, I/32, I/33, I/37

- перестройка 2/9, 2/14, 2/15  
 'ПЕЧ' (печать) 2/48  
 повторное вхождение переменной I/53, 2/10  
 'ПОЛ' (перевод алгебраического выражения в польск. форму) 4/18  
 поле зрения I/26, 2/7  
     памяти I/26, 2/7  
 понятия естественного языка I/14, 3/47  
 'ПОРЦЮК' 3/18, 4/30  
 'ПОСЛСМ' (последний символ выражения) I/35  
 правило конкретизации I/16, I/25  
 предикативная переменная 4/9  
 предикаты классические 3/19  
     , рефал-предикаты 3/23  
 предложение I/16, I/23, I/25, 2/6  
 предложение-комментарий 2/6, 2/59  
 прогонка 3/44, 4/36, 4/45, 5/16  
 проектирование (в процессе отождествления) I/44, I/51, I/52, 2/9  
 проекционный номер I/52, 2/10  
 просмотр сквозной 3/4, 3/28, 5/8  
     слева направо I/50, 3/4  
     справа налево 3/15, 4/16  
 равенство (классический предикат) I/43  
 'РАВ' (стандартный рефал-предикат) 5/26  
 размерность просмотра 3/10  
 разложение переменных 3/16  
 'РАЗСК' (разъединение скобок) 3/34  
 расцепление функции 3/41  
 рекурсивная переменная I/48, 2/10  
 рекурсивная функция глагольная 3/37  
     глагольно-именная 3/37  
     именная 3/37  
     однопробная 3/41  
     описание синтаксиса 4/30  
     , формат аргумента 3/7  
 рекурсивное сложение I/41  
     умножение 1/42  
     усеченное вычитание I/42  
 рекурсия в правой части предложения I/37

- рекурсия в левой части предложения 4/18  
 косвенная I/38  
 рефал-машина I/16, 2/6  
 рефал-объект 2/6  
 рефал-предикат 3/23  
 стандартный 3/24  
 рефмодуль 2/57, 2/58  
 свободная переменная базисная I/44, 2/10  
 закрытая I/53, 2/11  
 квазипредикативная 4/45, 5/5  
 непредикативная 4/43  
 открытая I/53, 2/14  
 , повторное вхождение I/53, 2/10  
 предикативная 4/9  
 рекурсивная I/48, 2/10  
 сентенциальные языки I/24  
 символ I/18, I/25, 2/4  
 символьные преобразования I/10, I/12  
 символьные скобки I/22, 2/4  
 символы обмена I/54, 2/4  
 'СИМОВ' (отщепить символов) 4/42  
 'СИМИ' (симметричность цепочки) 3/28  
 синтаксис функций 4/30  
 синтаксические типы специальные I/44  
 указатели I/45, 3/27  
 синтаксическое отождествление I/31, 2/8  
 'СКВ' (сквозной просмотр) 3/4, 3/28, 5/8  
 скобка 2/3  
 слияние упорядоченных списков 4/26, 4/28, 4/31  
 слияние функций 3/40  
 'СЛИУС'- слияние упорядоч. списков объектов 4/31  
 символов 4/28  
 слов 4/38  
 собственный знак I/17, 2/3  
 составной символ I/18, 2/4  
 представление в машине 2/30  
 'СПАРСК' (спаривание скобок) 3/50  
 спецификатор I/45, 2/4

- списковая структура памяти 2/19  
 старшинство знаков конкретизации 2/26  
 стек конкретизаций 2/27  
 сумка 3/7  
 терм I/23, 2/4  
 терм-объект 4/30  
 типовое выражение I/43  
 трансляторы с рефала 2/18  
     , эффективность 2/29, 2/52, 2/57  
 'УБРФ' (убрать функции) 5/6  
 удлинение открытой переменной I/51, 2/15  
 узел 2/9  
 указатель свободной переменной I/30, 2/4  
 'УПОРЯД' (упорядочивание объектов) 4/28, 4/32  
 упорядочивание объектов, алгоритм 4/24  
 управляющий знак 2/4  
 условные выражения классические 3/21  
 'УСТРЗАП' (устранение запятой) I/50  
 устранение открытых переменных 3/13  
     рекурсивных переменных 4/13, 4/35, 4/45  
 формальное описание семантики алгоритмических языков I/10  
 формат аргумента функции 3/7  
 функциональные скобки см. конкретизационные скобки  
 функция I/35, см. также рекурсивная функция  
 'ХВОСТ' 4/20, 4/23  
 числа, представление в машине 2/52  
     , выделение процедурой 'N' 2/55  
 'ЧИСЛО' (в АЛГОЛ'e-60) 4/10  
 'ЧИСЛО' (в рекурсивной арифметике) I/41  
 числовые символы 2/54  
     распознавание процедурой 'NS' 2/55  
 шаг рефал-машины I/26, 2/7  
 элементы левой части предложения /47, 2/8  
 • N, 2/55  
 • NS, 2/55  
 • R, 4/17

## СО Д Е Р Ж А Н И Е

МЕТАФУНКЦИИ	3
1. Компилирующие метафункции	3
2. Интерпретирующие метафункции	7
3. Интерпретируемый операторный язык	II
РЕШЕНИЯ ЗАДАЧ	29
ЛИТЕРАТУРА	48
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	49

№ T12647 от " 2 " viii 1971 г. Заказ № 805 Тираж 250 экз.

---

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