

Государственный комитет Совета Министров СССР
по делам строительства

Центральный научно-исследовательский и проектно-
экспериментальный институт автоматизированных
систем в строительстве
ЦНИИИАСС

УДК: 681.322.06:721.011:69.003:65.014.011.56

№ гос. регистрации

Инд. №

"Утверждаю"

Директор ЦНИИИАСС

А.Г.Н., доцент

А.А.Гусаков

3003 1977 г.

Технический проект экспериментальной
системы пакетов прикладных программ
автоматизированного проектирования
систем организационного управления
/логико-интерпретационный блок про-
ектирования/

Всего томов 3. Том 3, книга 3

Шифр 15-2-76

И.о. зам. руководителя отделения АСУС *М.М.Ливши* И.М.Ливши

Руководитель темы,
зав.отделом № АСУС

С.П.Никамор С.П.Никамор

Ответственный исполнитель,
с.н.с., к.ф.-ч.н.

Д.Б.Персиц Д.Б.Персиц

Москва - 1977 г.

Л6

АКАДЕМИЯ НАУК УССР
 ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР ОДЕССКОГО ОТДЕЛЕНИЯ
 ИНСТИТУТА ЭКОНОМИКИ

ЭКСПЕРИМЕНТАЛЬНАЯ СИСТЕМА ППП АВТОМАТИЗИРОВАННОГО
 ПРОЕКТИРОВАНИЯ СИСТЕМ ОРГАНИЗАЦИОННОГО УПРАВЛЕНИЯ

Д 4. СОДЕРЖАНИЕ ЗАДАЧ ПРОЕКТИРОВАНИЯ
 И АЛГОРИТМЫ ИХ РЕШЕНИЯ

на 60 листах

"Согласовано"

Руководитель темы

С. П. НИКАНОРОВ
 Ст. научный сотрудник к. ф.-и. н.

Д. Б. ПЕРСИЦ

Ответственный испол-
 нитель ст. инженер

А. В. АЙЗЕНГАТ
 Ст. инженер

Б. А. ЗАКС

Зав. отделом модели-
 рования и системотех-
 ника

Г. Я. ПОРТНОВ

15-276

ТПТЗмЗ

ОГЛАВЛЕНИЕ

| | Стр |
|--|-----|
| Д.4. Содержание задач проектирования и алгоритмы их решения | 2 |
| 5. Управление процессом машинного проектирования (главная управляющая программа - ГУП) | 4 |
| 5.1. Постановка задачи | 4 |
| 5.2. Схема функционирования | 4 |
| 5.3. Алгоритм ГУП | 7 |
| 6. Управление данными (СУБД) | 10 |
| 6.1. Постановка задачи | 10 |
| 6.2. Описание алгоритмов основных программ СУБД | 12 |
| 7. Обоснование принятых решений | 54 |
| 7.1. Выбор языков программирования | 54 |
| 7.2. Обоснование расчетов трудоемкости | 55 |
| 7.3. Обоснование общей архитектуры | 56 |
| Приложение. Номенклатура модулей системы ШПД | 58 |

4

5. Управление процессом машинного проектирования (главная управляющая программа-ГУП).

5.1. Постановка задачи.

Выполнение машинного проектирования требует присутствия в системе ППП ГУП-главной управляющей программы.

Эта организующая программа должна осуществлять проектирование в любом режиме, определяя взаимодействия пакетов внутри системы и системы с проектировщиком, используя текущее состояние информационной базы СУБД.

Функциями ГУП в режиме РСР ЭС ЭВМ являются следующие:

- организация управления заданием;
- генерация системы ППП;
- распределение устройств ввода-вывода;
- связь с проектировщиком;
- выполнение загрузки модулей из библиотеки в основную память и передача управления на точки входа;
- обработка программных прерываний;
- управление созданием контрольных точек, рестарт системы.

В связи с намечаемым развитием системы предусматривается возможность расширения запросов к ГУП, в частности:

- организация работы системы в мультипрограммном режиме;
- обеспечение непосредственного доступа проектировщика к ЭВМ путем создания информационно-поискового языка общения пользователя с системой на уровне терминологии МЕТОДА.

5.2. Схема функционирования.

Предполагаемый состав ГУП имеет следующие укрупненные блоки:

- ГЕНЕРАТОР;
- ПЛАНИРОВЩИК;
- АНАЛИЗАТОР;
- СОВЕЩНИК.

ГЕНЕРАТОР определяет различные режимы работы системы.

генерирует систему в соответствии с конкретным заданием проектировщика.

ПЛАНИРОВЩИК осуществляет контроль и анализ задания, планирует выполнение задания.

АНАЛИЗАТОР - блок, контролирующий выполнение задания, вызывающий необходимые функциональные блоки, осуществляет обработку программных прерываний, управляет созданием контрольных точек и рестарта системы, обеспечивает связь с операционной системой.

СОБЕСЕДНИК обеспечивает диалоговый режим для уточнения и исправления задания, управление формированием и выдачей сообщения о ходе проектирования, содержит интерпретатор языка директив.

Схема функционирования ГУП приведена на Рис. 5.1.1.

Взаимодействие функциональных блоков в соответствии со схемой функционирования заключается в следующем:

Программа системного ввода вводит управляющие операторы задания, в основную память вызывается ГУП. ГЕНЕРАТОР ГУП определяет режим проектирования. ПЛАНИРОВЩИК обрабатывает управляющие операторы задания программой ИНТЕРПРЕТАТОР. Эта программа осуществляет чтение одного входного потока. Она читает операторы управления заданиями, анализирует их содержание и строит таблицы, которые используются во время инициализации и выполнения пунктов задания. Программа ИНИЦИАТОР выбирает для выполнения в входного потока очередную пункт задания. ИНИЦИАТОР анализирует требования на устройстве ввода-вывода от пунктов задания, выделяет им устройства, используя дополнительные средства автоматического распознавания томов, и передает управление пункту задания, сформированному в ИНТЕРПРЕТАТОРЕ. При этом ИНИЦИАТОР "присит" АНАЛИЗАТОР наблюдать за выполнением задачи. Управляющая программа выделяет место в основной памяти в соответствии с информацией, находящейся в элементе справочника библиотек, который относится к данному загрузочному модулю, а затем загружает программу в основную память. Когда модуль загрузки появляется в основной памяти, управление передается в точку входа. АНАЛИЗАТОР своей программой управления задачами передает управление первому загрузочному модулю и наблюдает за его вы-

15-2-76

ТПТЗмЗ

15.2.16
ГПЗ-343

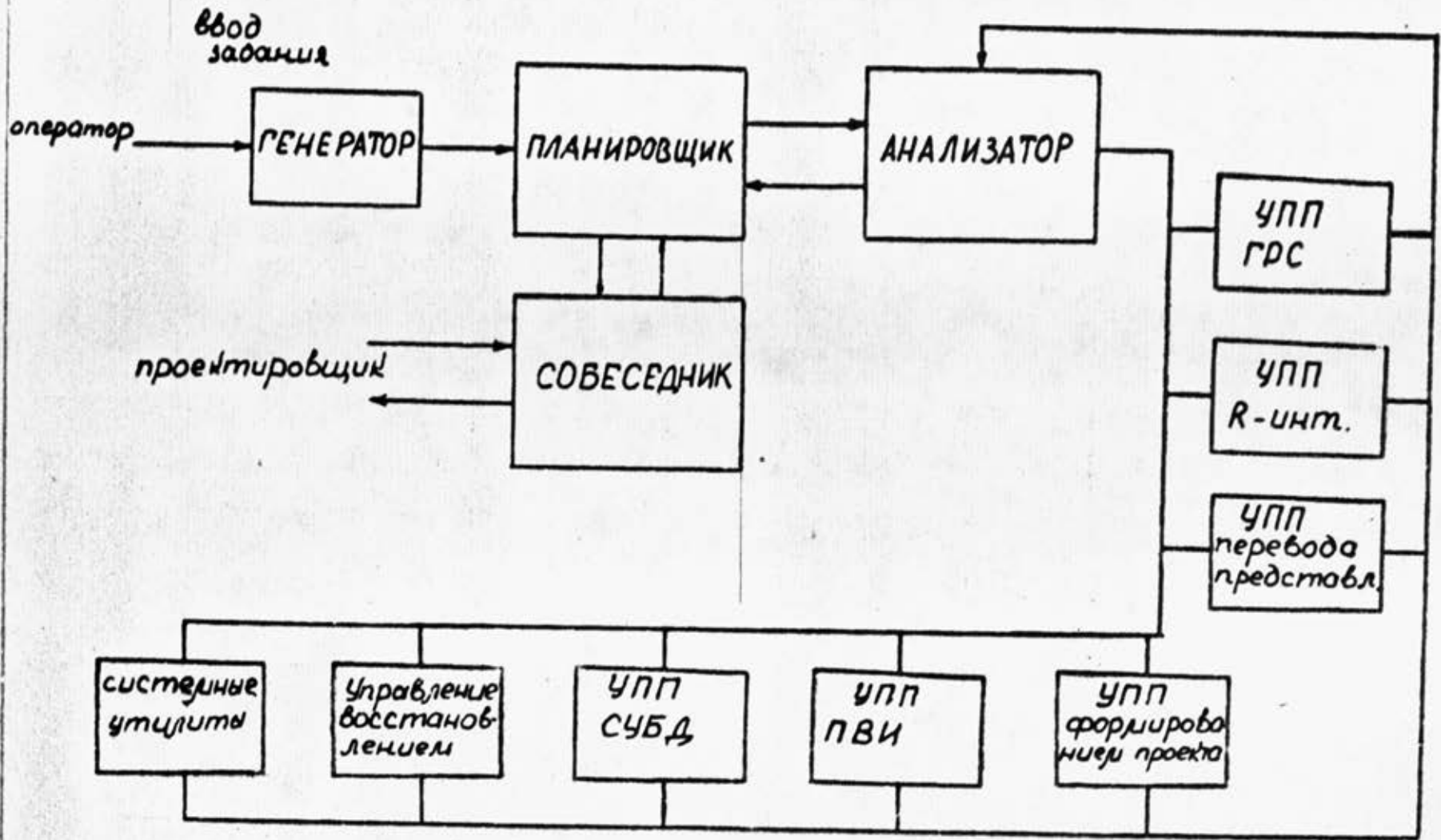


Рис. 5.2.1. Схема функционирования ГУП.

полнением. Для контроля за задачей управляющая программа группирует всю управляющую информацию, относящуюся к этой задаче, в блок управления задачей. АНАЛИЗАТОР информируется о прекращении выполнения задачи, независимо от того, завершилась она нормально или аварийно. Задача может перейти в состояние ожидания прямо или косвенно. Окончание использования ресурса всегда указывается прерыванием, в результате которого соответствующий распорядитель ресурса получает управление. Поэтому, если активная задача прерывается, она получит управление после того, как АНАЛИЗАТОР обработает прерывание. По завершении пункта задания АНАЛИЗАТОР информирует об этом ПЛАНИРОВЩИК задания независимо от того, было ли окончание нормальным или аварийным.

Когда завершается задание или пункт задания, ИНИЦИАТОР возвращает управление программе системного ввода и обработка повторяется.

При однопрограммном режиме только одно задание может поступать из входного потока в основную память.

СОБЕСЕДНИК выполняет команды оператора, которые управляют системными функциями, обрабатывает сообщения проектировщика по уточнению задания. По указанию проектировщика выполнение процесса проектирования может быть отменено, временно приостановлено. СОБЕСЕДНИК обрабатывает запросы проектировщика и обеспечивает прием или выдачу дополнительной информации относительно статуса системы.

5.3. Алгоритм ГУЦа.

ГУЦ вызывается в основную память под управлением ОС ЭС. Набор данных, обрабатываемых ГУЦ, - это некоторая последовательность управляющих операторов, определяющих конкретный режим работы системы ППП.

Блок-схема алгоритма ГУЦ описывает прохождение этих операторов.

Блок А2 проверяет, возможна ли генерация системы, указанная набором управляющих операторов. ГЕНЕРАТОР осуществляет эту проверку по таблице, в которой для каждого из режимов описаны необходимые компоненты системы и технические средства.

15-1-78
ТПТЗинЗ

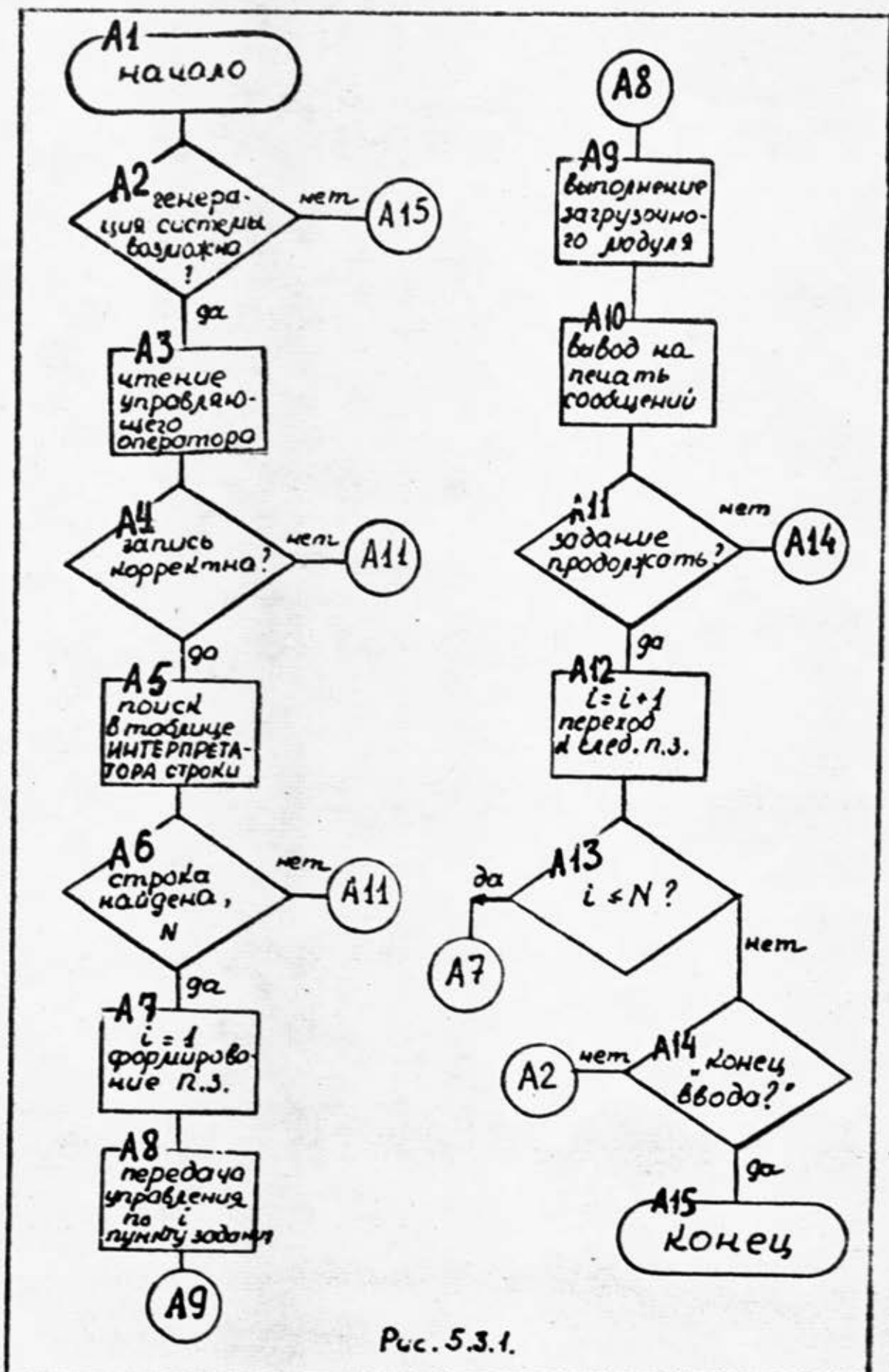


Рис. 5.3.1.

Набор управляющих операторов формирует задание проектирования, а каждый из управляющих операторов - пункт задания, соответствующий определенному этапу проектирования.

Блок А3 осуществляет чтение очередного управляющего оператора.

Блок А4 проверяет корректность управляющего оператора.

Блок А5 осуществляет поиск таблицы ИНТЕРПРЕТАТОРА, соответствующей этому управляющему оператору строки. Строка должна определить перечень управляющих сигналов, передаваемых модулям, их идентификаторы и содержание работы, которую они должны выполнить.

Блок А6 проверяет наличие строки, запоминает количество управляющих сигналов N .

Блок А7 формирует пункт задания.

Блок А8 осуществляет загрузку в основную память i -того модуля и передачу управления на его точку входа.

Блок А9 производит выполнение загрузочного модуля с выделением всех устройств и используемых массивов СМД.

Блок А10 определяет состояние задания средствами СОБЕСЕДНИКА.

Блок А11 анализирует причину завершения программы.

В случае нормального завершения осуществляется передача управления на блок А12 для перехода к следующему пункту задания. В нем индекс сигнала увеличивается на 1.

Блок А13 производит проверку $i \leq N$. Если $i \leq N$ управление передается на формирование нового пункта задания, цикл блоков А7-А13 повторяется, пока весь перечень управляющих сигналов не будет обработан.

В случае $i > N$ блок А14 осуществляет проверку на признак "конец ввода", и если он отрицателен, то передает управление на блок А12, для повторения процедуры системного ввода.

Алгоритм подлежит уточнению и расширению по мере того, как будет осуществляться стыковка пакетов.

15.2.28
Т/ПтЗмЗ

6. Управление данными (СУБД).

6.1. Постановка задачи.

Программное обеспечение банка данных - СУБД - предназначено для создания, обновления и использования массивов БД.

При вводе и пополнении информационных массивов производится перекодировка во внутренний код, контроль вводимых данных и выдача сообщений об ошибках, преобразование корректных данных к виду, удобному для работы пакетов. Контроль данных состоит из контроля перфорации (например, проверка контрольных сумм) и некоторого синтаксического и семантического контроля, при этом в случае обнаружения ошибок производится дальнейший контроль всей информации.

При выводе информационных массивов производится преобразование их в соответствии с выходными формами.

Внесение изменения средствами СУБД ограничивается удалением базисных родов структуры, дополнения и отображения и удалением стандартных подграфов из графа.

Необходимо обеспечить выборку следующих данных из информационных массивов:

- родов структур и дополнений из массивов БРС, БДП, БОТ или РСД;
- конститuent из рода структуры или дополнения;
- подграфов из графа конститuent;
- слоя из графа в расширенном представлении;
- R-интерпретации конститuent.

СУБД должен обеспечить для данных преобразование их из одного представления в другое, в частности, приведение графа к расслоенному виду.

В СУБД имеется также ряд сервисных средств, в частности, средства формирования списка начальных вершин, списка всех вершин и списка конечных вершин графа.

Для решения этих задач предлагается следующая архитектура СУБД. СУБД состоит из управляющей программы (УП) и набора модулей (перечень модулей и описание их алгоритмов приведены в п.6.2.). При необходимости вызвать модуль СУБД формируется запрос в пакете или проектировщиком, который поступает в УП

15-2-76
ТПГЗамЗ

СБД. После анализа и проверки запроса вымывается соответствующий модуль **СБД** для выполнения заданной работы, а затем ^{снова} возвращает управление **ПП СБД**.

СБД выполняет следующие функции:

- 6.1. Ввод и пополнение информационных массивов;
- 6.2. Вывод массивов;
- 6.3. Внесение изменений в массивы (совместно с **ПП** внесения изменений);
- 6.4. Выборка данных из информационных массивов;
- 6.5. Преобразование данных.

6.2. Описание алгоритмов основных программ СУБД.

В данном пункте представлены следующие группы модулей СУБД:

- управляющая программа СУБД.
- модули ввода-вывода,
- модули выборки данных,
- модули изменения данных,
- ~~служебные~~^{сервисные} модули.

Из модулей ввода-вывода в данном пункте описаны только модули ввода и контроля для ППП ГРС и некоторые средства контроля, используемые при вводе для ППП *R* -интерпретации. Для остальных модулей ввода-вывода в качестве задания на программирование может служить описание соответствующих форм в 5.3. документа ДЗ и информационных массивов в 1.4.2.4, 3.4.4.4 документа Д4.

Из модулей изменения данных представлены модули удаления базисных родов структур, базисных дополнений и отображений, удаления подграфов из графа, при внесении изменений в ^{другие} входящие данные используется ППП внесения изменений.

Приведено описание алгоритмов следующих основных модулей СУБД:

- 1) УП СУБД.
- 2) модуль ввода и контроля базисных родов структур и дополнения (МВБРД),
- 3) модуль ввода базисных отождествляющих отображений (МВБОТ),
- 4) модуль ввода и контроля операционной схемы (МВОСХ),
- 5) модуль ввода и контроля отображения (МВЭТА),
- 6) модуль контроля типов аргументов операций операционной схемы (МКТАР)
- 7) модуль ввода массивов РСД, ОТСТ и КОПС (МВРДО),
- 8) модуль выборки родов структур или дополнения,
- 9) модуль выборки конституент рода структуры или дополнения,
- 10) модуль выборки подграфов из графа конституент,
- 11) модуль выборки слоя из графа в расслоенном представлении,
- 12) модуль выборки *R* -интерпретации конституент.

- 13) модуль удаления базисных родов структур,
 14) модуль удаления стандартных подграфов из графа,
 15) модуль формирования графа конституэнт,
 16) модуль приведения графа к расслоенному виду,
 17) модуль формирования списка начальных вершин ^{и списка всех вершин графа}
 (№ НВВ),
 18) модуль формирования списка конечных вершин графа,
 19) модуль изменения ориентации ссылок.

6.2.1. Описание алгоритма управляющей программы СУБД (УП СУБД).

УП СУБД предназначена для анализа запросов, поступающих от пакетов или проектировщика, и организации их выполнения.

Исходные данные: запросы к СУБД.

В запросе указывается идентификатор модуля и аргументы, необходимые для работы модуля.

УП анализирует очередной запрос, проверяет его и либо сообщает об ошибочности запроса, либо вызывает соответствующий модуль для выполнения запроса. После выполнения запроса, УП сообщает об этом и переходит к следующему запросу.

Блок-схема алгоритма не приводится.

6.2.2. Описание алгоритма модуля ввода и контроля базисных родов структур или дополнений.

Модуль предназначен для ввода и/или контроля базисных родов структур и дополнений и в случае их правильности формирования соответствующих массивов и каталогов.

Исходные данные: массив соответствующий форме 2^x .

Блок-схема алгоритма приведена на рис. 6.2.1.

В блок-схеме приняты следующие обозначения:

МАС2- массив, соответствующий форме 2,

Σ - контрольная сумма, указанная в МАС2,

$\Sigma 1$ - контрольная сумма, вычисляемая при вводе МАС2,

КС- контрольная сумма,

И- признак правильности (К=1- ошибка не обнаружена, К=0- есть ошибка),

ТАБП- таблица номеров конституэнт типа П,

ТАБК- таблица номеров конституэнт типа К,

ТАБА- таблица номеров конституэнт типа А,

^{х)} Описание формы приведено в 5.1. документа Д3.

15-2-76

Т/ПтЗмЗ

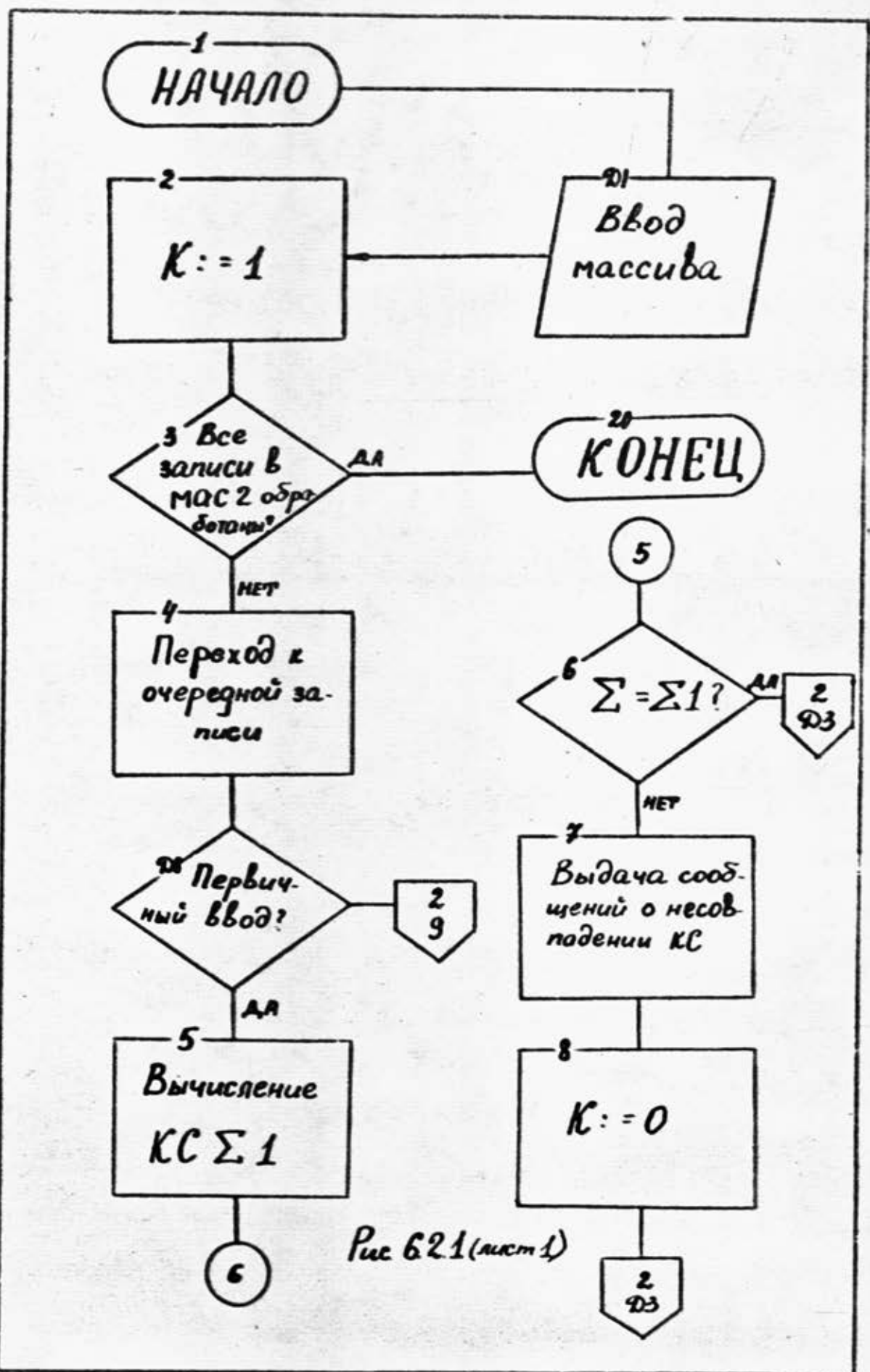


Рис 621 (лист 1)

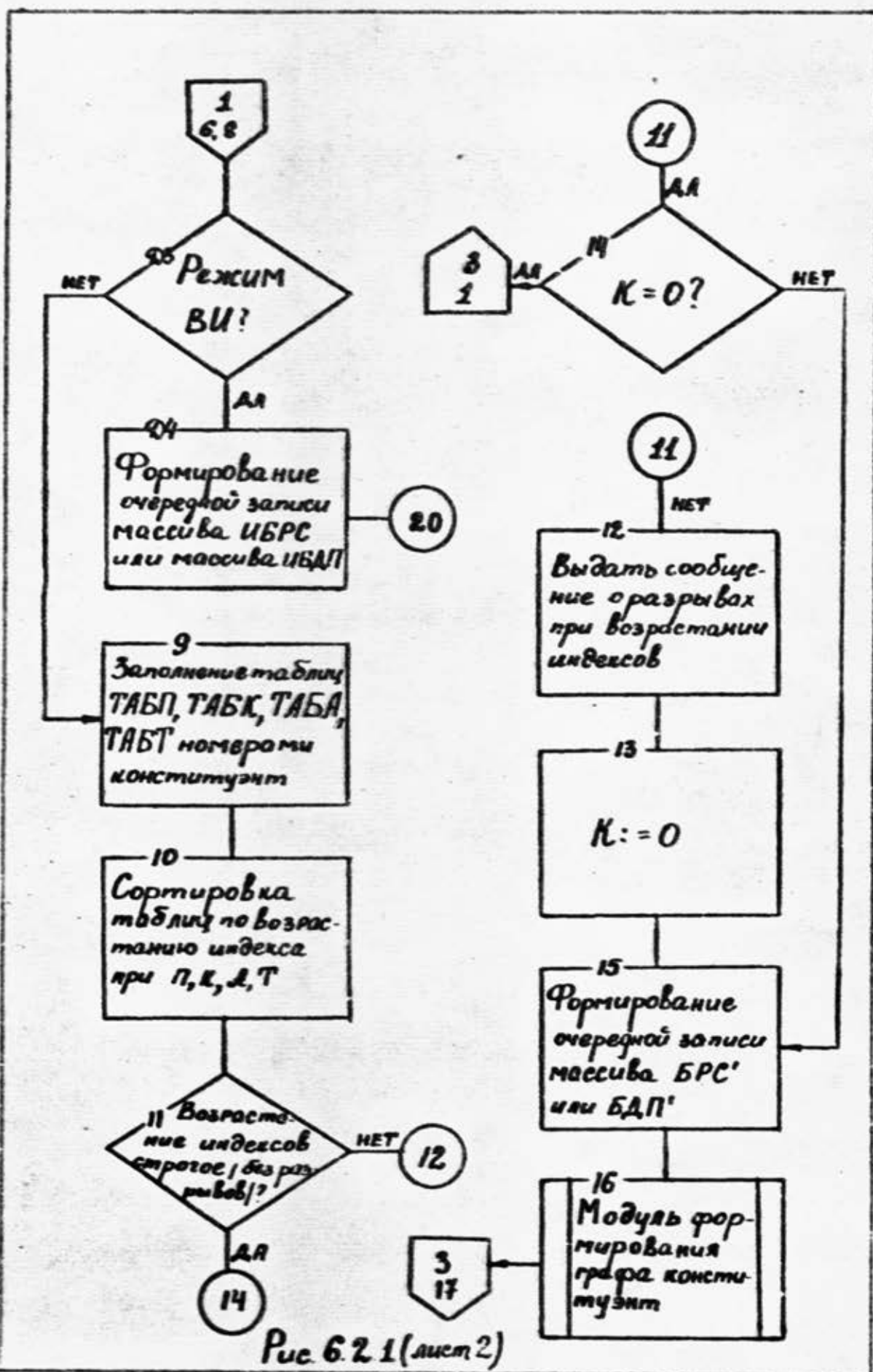


Рис. 6.21 (лист 2)



Рис. 6.2.1 (лист 3)

ТАБТ- таблица номеров конститuent типа Т.

БРС' - рабочий массив базисных родов структур

БДП' - рабочий массив дополнений

БДП - массив базисных дополнений

БРС- массив базисных родов структур

Т - признак правильности записи рода структуры и дополнения, выдаваемый модулем формирования графа конститuent.

ВИ- внесение изменений

ИБРС - массив изменений в базисные роды структур

ИБДП- массив изменений в дополнения.

Опишем отдельные блоки алгоритма:

Блоки Д1, Д2. В случае первичного ввода осуществляется проверка контрольных сумм, в случае, если массив уже имеется в системе, осуществляется переход на блок 9.

Блоки Д3, Д4. В случае режима внесения изменения проверяется только совпадение контрольных сумм.

Блок 6. Проверки контрольных сумм Σ , заданной в форме 2, и ΣI , вычисленной (блок 5) при вводе формы 2- контроль перфорации.

Блок 9, 10. Заполнение соответствующих таблиц номерами конститuent и сортировка номеров.

Блок 11, 12. Контроль отсутствия разрывов в нумерации конститuent, выдача соответствующего сообщения в случае разрывов.

Блок 12. Формирование очередной записи рабочего массива БРС' или БДП', структура записи описана в 1.4.2.

Блок 13. Вызов модуля формирования графа конститuent^{x)}, модуль передается запись результатов работы модуля используется признак Т- результат контроля записи (Т=I- если запись правильная).

Блок 14, 15. Если запись правильная, то она записывается в массив БРС (если она соответствует роду структуры) или БДП (если она соответствует дополнению), а ее имя записывается в соответствующий каталог.

Результатом работы модуля являются массивы БРС, БДП и их

^{x)} Описание модуля приводится в 6.2.

15-2-78

ТЛТЗмЗ

каталоги КАТЭ, КАТЕ, содержащие правильные (т.е. прошедшие контроль) записи формы 2 и их имена.

6.2.3. Описание алгоритма модуля ввода базисных отождествляющих отображений.

Модуль предназначен для ввода базисных отождествляющих отображений и в случае их правильности формирования соответствующего массива БОТ.

Исходные данные: массив, соответствующий форме 3^x).

Блок-схема алгоритма не приводится ввиду простоты.

Результатом работы модуля является массив БОТ^{xx}) тех базисных отождествляющих отображений, которые удовлетворяют требованиям контроля перфорации и сообщения об ошибочных записях; для записей массива БОТ формируется каталог КАТГ отождествляющих отображений - список соответствующих им базисных индексов.

6.2.4. Описание алгоритма модуля ввода и контроля операционной схемы.

Модуль предназначен для ввода и/или контроля операционной схемы.

Исходные данные:

1) массив, соответствующий форме I,^{xxx})

2) признак первичного ввода или вторичного ввода,

3) признак режима внесения изменений.

Блок-схема алгоритма представлена на рис. 6.2.2.

В блок-схеме приняты следующие обозначения:

МАС - массив на ПК (в случае первичного ввода) или в системе, соответствующий форме I.

КС - контрольная сумма.

ВИ - внесение изменений,

НОПСХ - массив изменений в массив ОПСХ^{xxxx})

x) описание формы приведено в 5.2. документа ДЗ.

xx) структура массива БОТ приведена в 1.4 (аналогична ОТ0Т).

xxx) описание формы I приведено в 5.2.

xxxx) описание массива приведено в 1.4

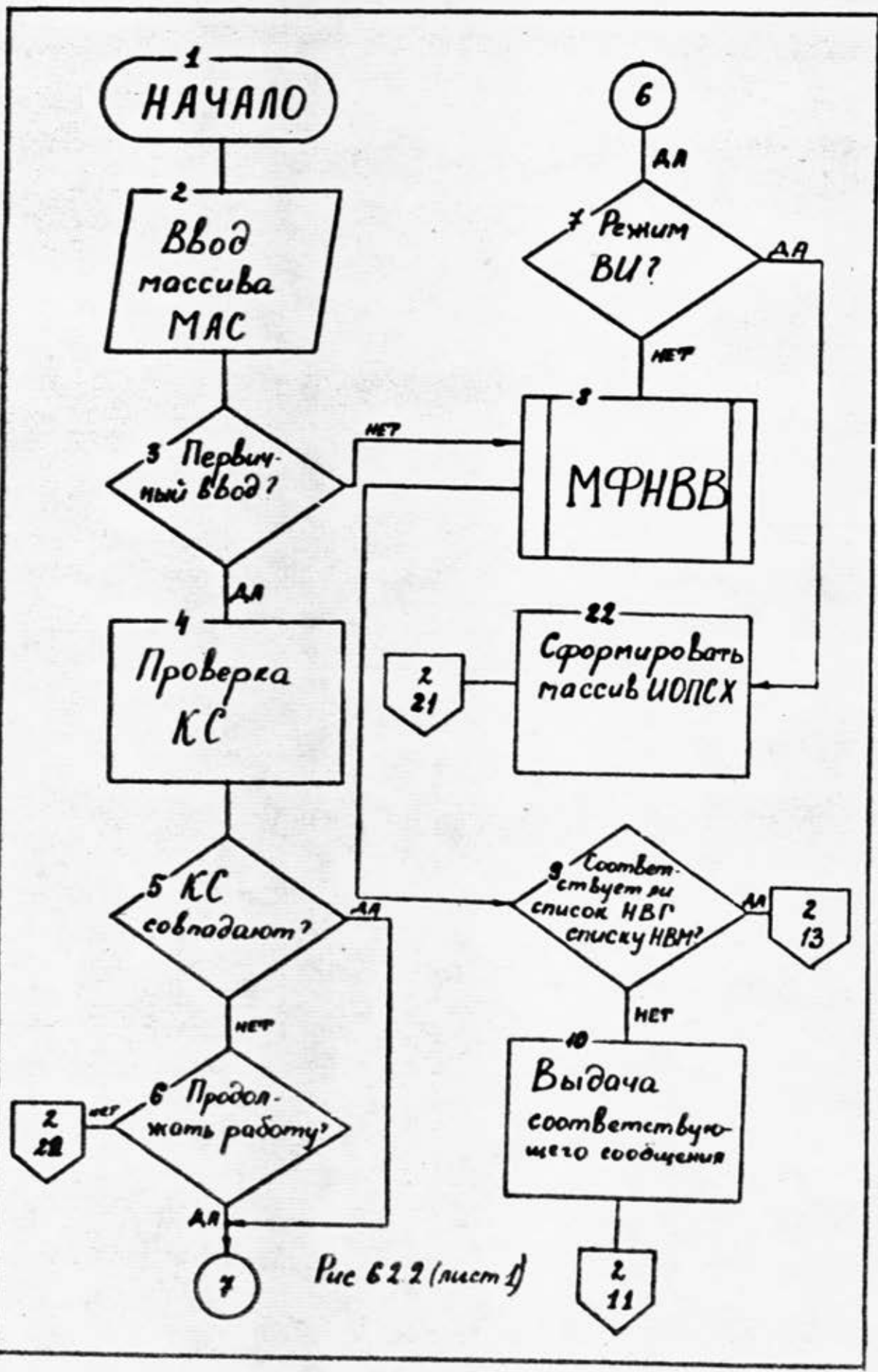


Рис 622 (лист 1)

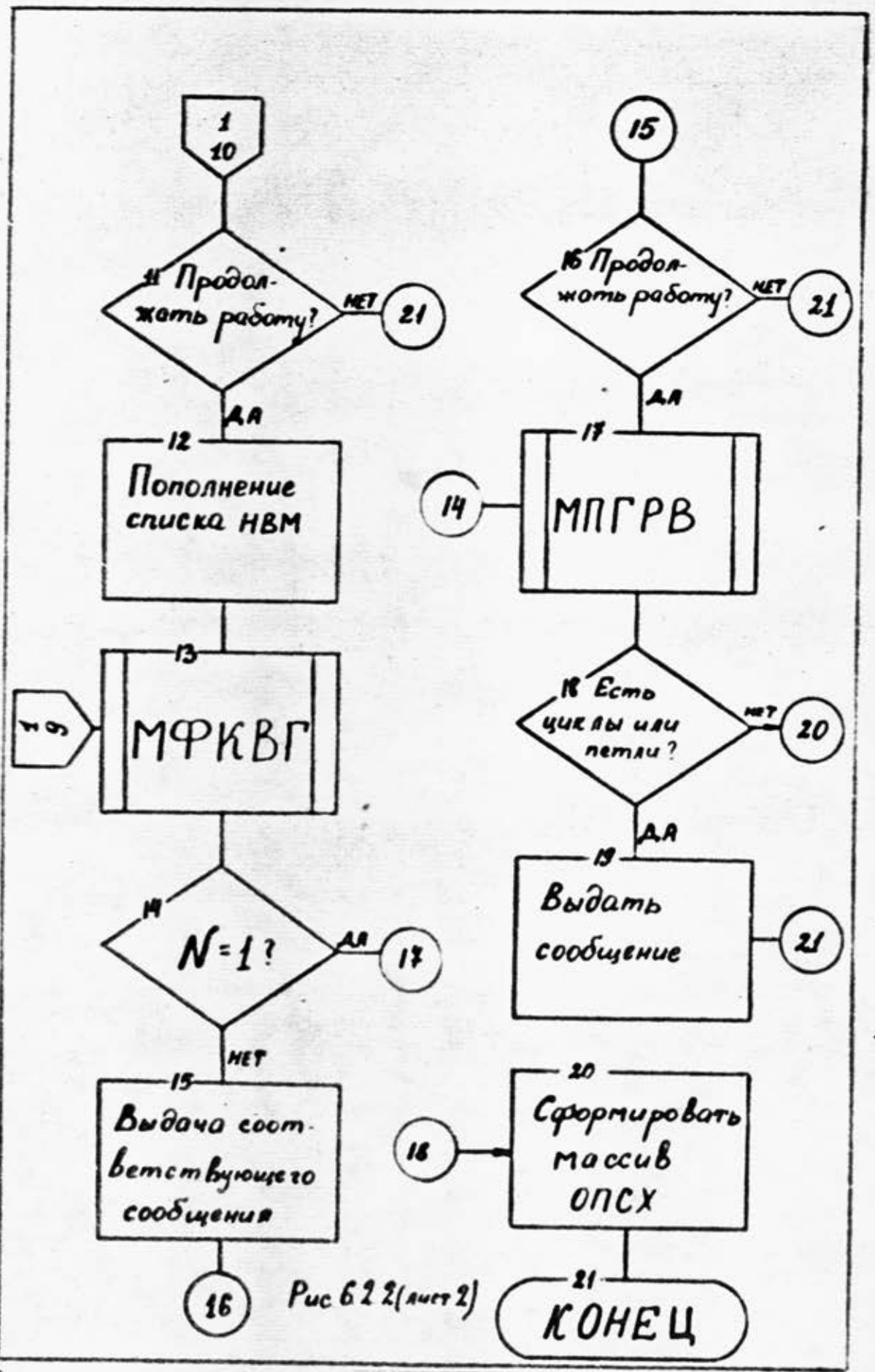


Рис 6.2.2 (лист 2)

10-2-76
Т.П.Зенз

МФНВВ- модуль формирования списка начальных вершин и списка всех вершин графа.

НВМ- нижние вершины в массиве.

НВГ- нижние вершины графа.

МФКВГ- модуль формирования списка конечных вершин графа.

N - число конечных вершин.

МПГРВ- модуль приведения графа к расслоенному виду.

ОПСХ- массив, описанный в I.4.

Опишем отдельные блоки алгоритма:

Блок 3. В случае первичного ввода сперва осуществляется контроль перфорации- проверка контрольных сумм, в случае вторичного ввода (в частности, после внесения изменений или при контроле уже имеющейся в системе операционной схемы) проверка контрольных сумм не производится.

Блок 7. В случае режима внесения изменений при вводе изменений производится только контроль перфорации.

Блок 8-12. Вызов модуля формирования списка начальных вершин и списка всех вершин графа, после чего списки начальных вершин массива и начальных вершин графа сверяются, и в случае продолжения работы, список начальных вершин массива пополняется не входящими в него начальными вершинами графа.

Блок 13-16. Вызов модуля формирования списка конечных вершин графа, модулю передается массив и признак $K=I$, в случае неединственности конечной вершины выдается список конечных вершин и по решению проектировщика работа прекращается или может быть продолжена.

Блок 17-19. Вызов модуля приведения графа к расслоенному виду, модуль либо приводит граф к расслоенному виду, либо выдает сообщения о наличии в графе петель и циклов (в этом случае работа прекращается). В случае отсутствия в графе петель и циклов, формируется массив ОПСХ, используемый в ППП ГРС.

Результатом работы модуля является либо массив ОПСХ, либо сообщение об обнаруженных ошибках.

6.2.5. Описание алгоритма модуля ввода и контроля отображения η .

Модуль предназначен для ввода отображения η и контроля задания отображением η соответствия между начальными вершинами

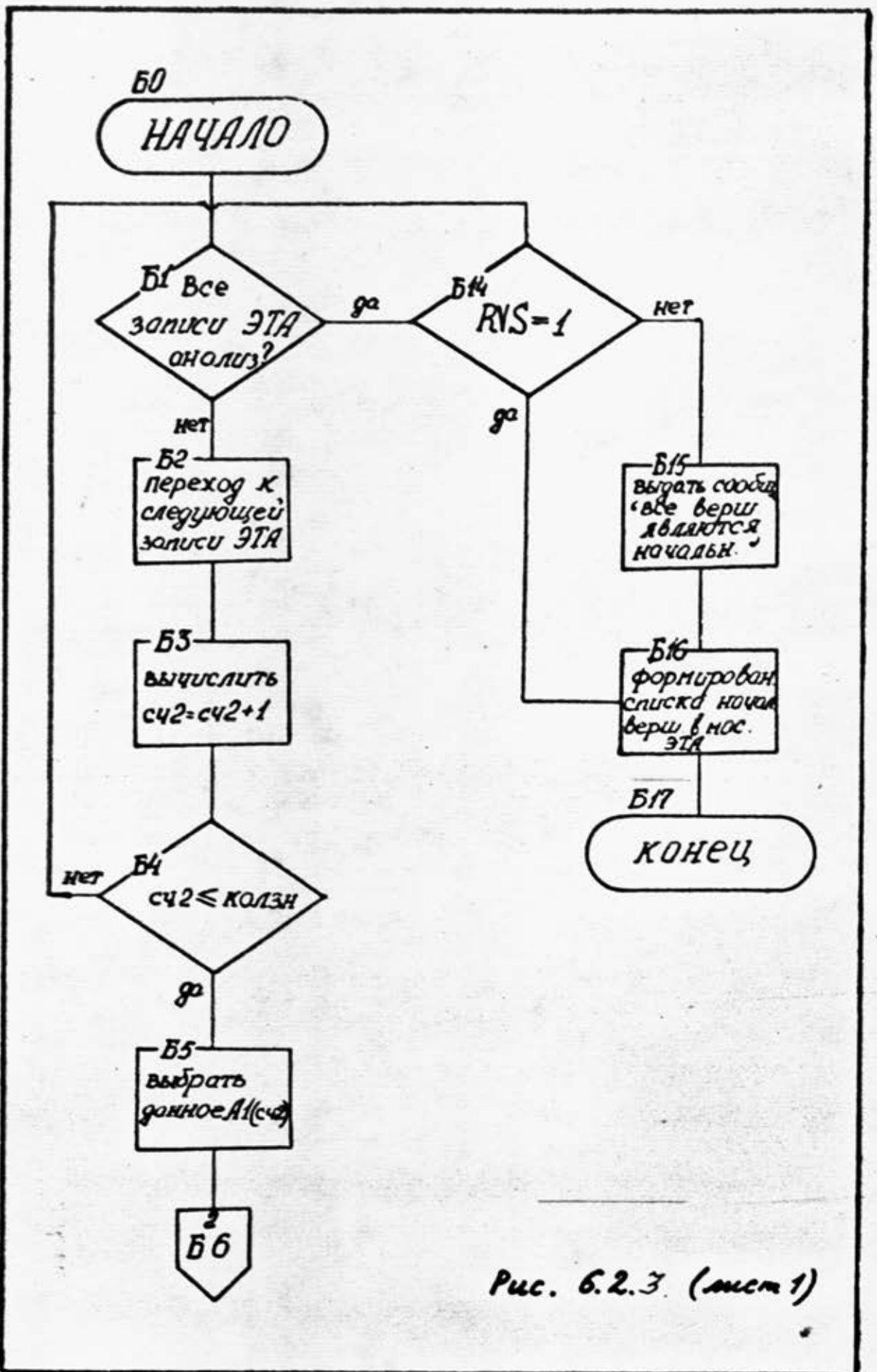


Рис. 6.2.3. (лист 1)

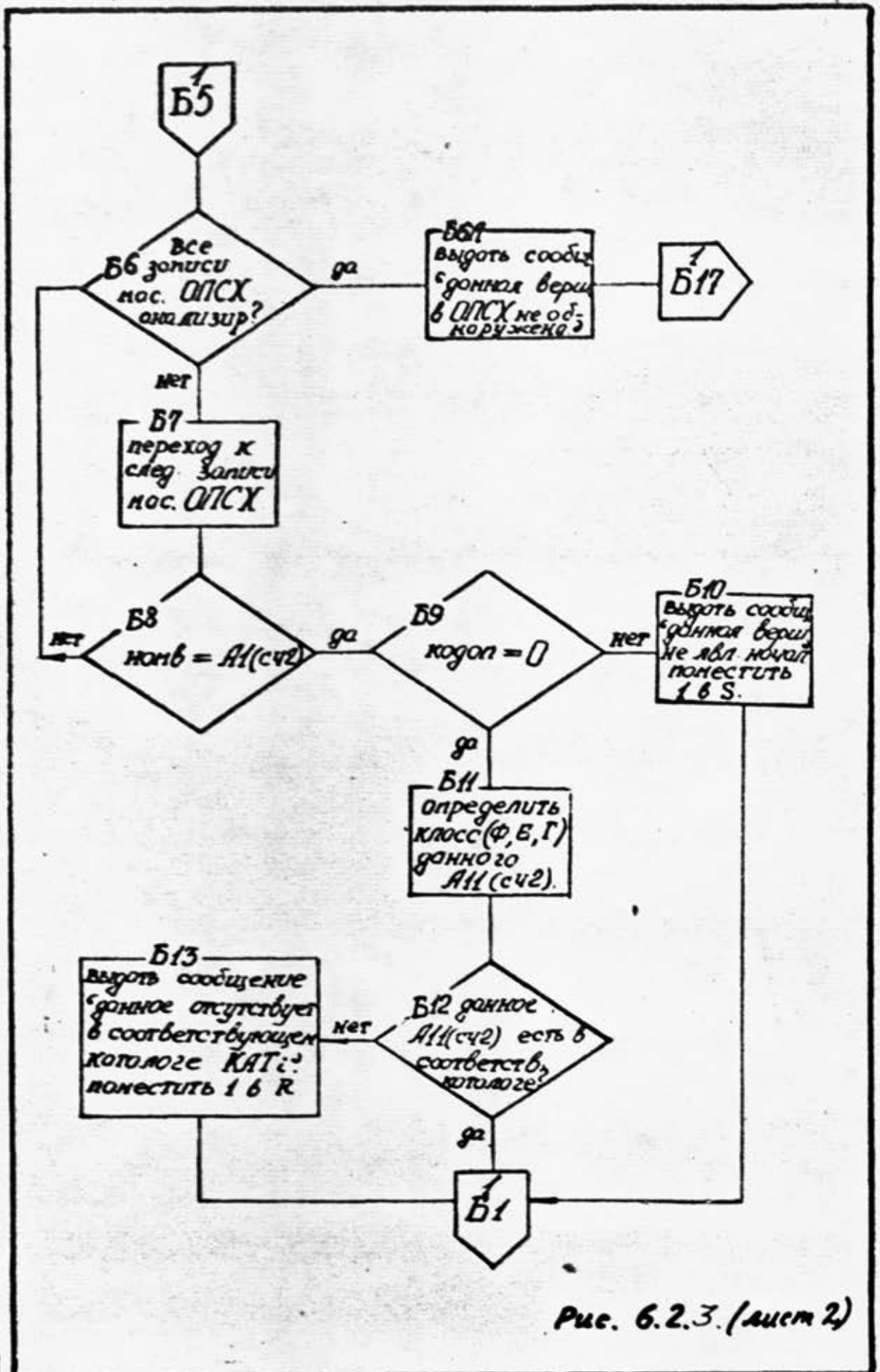


Рис. 6.2.3. (лист 2)

ми операционной схемы и базисными родами структур, дополнениями и отображениями.

Исходные данные: массив ЭТА, соответствующий отождествлению h (форма 3^{XX}), массив ОПСХ, каталоги базовых родов структур (КАТФ), базовых дополнений (КАТЕ), отображения (КАТГ).

В блок-схеме приняты следующие обозначения:

ОПСХ- массив записей операционной схемы, структура которого приведена в I.4.

ЭТА- массив отождествляющих отображений, структура приведена в 5. I.

OX 2- счетчик вершин в массиве ЭТА, в начале работы счетчик очищен.

колзн- данное массива ЭТА- количество значений.

$A_I(OX 2)$ - первое данное строки, номер вершины.

$A_{II}(OX 2)$ - второе данное строки, содержимое, относящееся к этой вершине.

КАТФ - каталог базовых родов структур.

КАТЕ - каталог дополнений.

КАТГ - каталог отображений.

НОМв - номер вершины, данное массива ОПСХ.

КОДоп - код операции- данное массива ОПСХ.

S - признак того, является ли данная вершина начальной в ОПСХ ($S = I$, если является, $S = 0$ - если не является), в начале работы $S = 0$.

A - признак наличия в соответствующем каталоге $A_{II}(OX 2)$ ($A = I$ - если да, $A = 0$ - если нет), в начале работы равно 0.

Блок схема алгоритма приведена на рис. 6.2.3.

Бл. I-4. Чтение записей массива ЭТА.

Бл. 5. Выбирается номер вершины $A_I(OX 2)$.

Бл. 6. Проверка, все ли записи массива ОПСХ проанализированы.

Бл. 7. Переход к следующей записи массива ОПСХ.

Бл. 8. Проверка соответствия номера вершины в массиве ОПСХ и в массиве ЭТА.

XX) Описание формы 3 приведено в 5. I документ ДЗ.

15-276
Т ПТЗ мЗ

Бл. 9. Проверка, является ли вершина начальной в ОПСХ.

Бл. 11. Определение, к какому классу принадлежит второе данное записи массива ЭТА.

Бл. 12. Проверка наличия соответствующей записи в каталоге (КАТФ, КАТЕ, КАТГ).

Бл. 14. Проверка наличия ошибок.

Результатом работы модуля является либо текст о правильности соответствия, либо сообщения об обнаруженных ошибках, которые могут быть двух вариантов: а) данная вершина не является начальной; б) данное АИ (СК 2) не обнаружено ни в одном из каталогов (дополнений, отображений, родов структур).

После окончания работы в случае правильности соответствия модуль вызывает модуль МКАР, описанный в 6.2.6.

6.2.6. Описание алгоритма модуля контроля соответствия типов аргументов операциям операционной схемы.

Модуль предназначен для контроля типов аргументов операция операционной схемы.

Исходные данные: 1-ый аргумент: операционная схема (ОПСХ), 2-ой аргумент - таблица операций (ТАБОР).

Таблица операций ^{ТАБОР} приведена на рис. 6.2.4.

Блок-схема алгоритма приведена на рис. 6.2.5.

В блок-схеме приняты следующие сокращения и обозначения: ОПСХ - массив операционной схемы, структура приведена в 1.4.1.

ЭТА - массив отождествляющих отображений, структура приведена в 5.1.

ТАБОР - таблица соответствия типов аргументов операциям операционной схемы.

СК 3 - счетчик аргументов массива ОПСХ, в начальный момент счетчик очищен.

Коларг - количество аргументов, данное массива ОПСХ.

В1 (СК 3) - аргумент, данные массива ОПСХ.

Т - признак соответствия типов аргументов операциям (Т=1 есть соответствие, Т=0-нет).

РТ - рабочая строка типов аргументов.

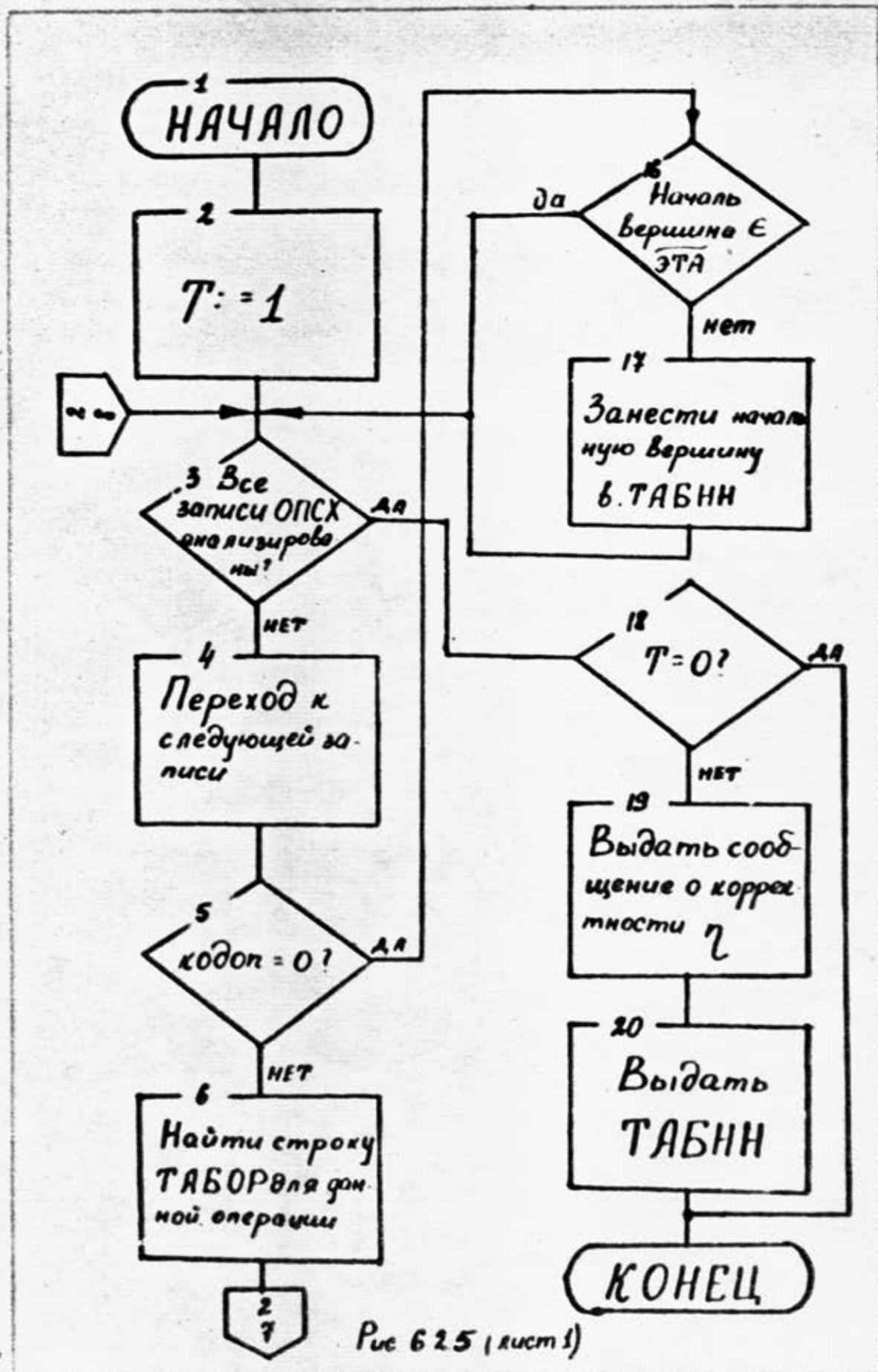
Кодоп - код операции в ОПСХ.

ТАБНК - таблица начальных вершин, входящих в массив

Таблица операция ТАБОР.

| Код, соответствующий операции | Количество аргументов | Аргументы | | | |
|-------------------------------|-----------------------|-----------|----|-----|-----------|
| | | I | II | III | Последний |
| OP _I | 2 | ⊙ | ⊙ | | |
| OP ₂ | 2 | ⊙ | ⊙ | | |
| OP ₃ | 3 | ⊙ | ⊙ | Г | |
| OP _{3.I} | 3 | ⊙ | ⊙ | Г | |
| OP ₅ | 1 | ⊙ | | | |
| OP ₇ | 2 | ⊙ | Г | | |
| OP ₉ | 4 | ⊙ | ⊙ | Г | Г |
| OP _{II.I} | 3 | ⊙ | ⊙ | Г | |
| OP _{II} | 3 | ⊙ | ⊙ | Г | |
| OP _{IS} | 1 | ⊙ | | | |

Рис. 6.2.4.



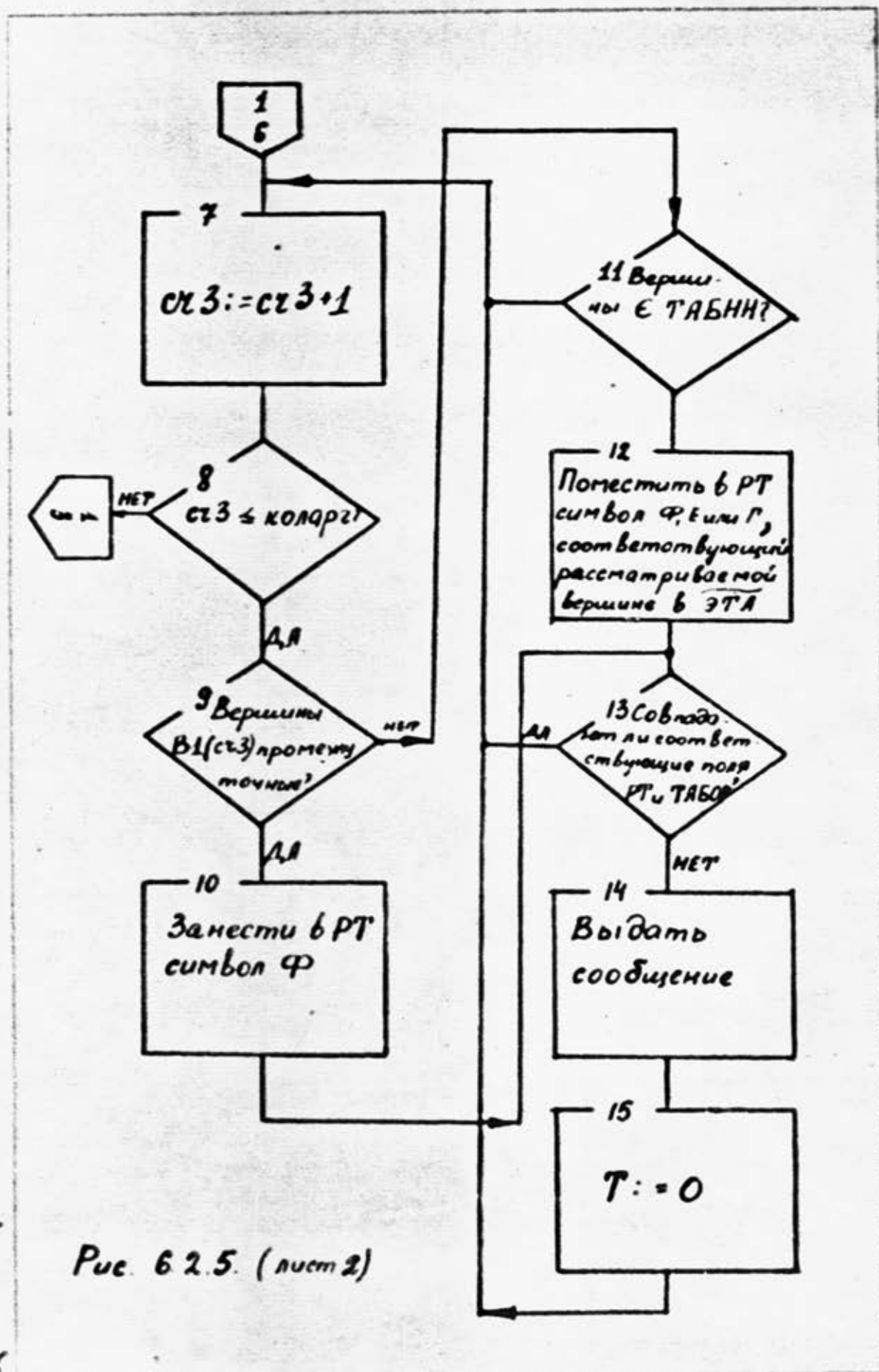


Рис. 6.2.5. (лист 2)

ОПСХ, но не вошедших в массив ЭТА.

Опишем отдельные блоки алгоритма.

Блоки 3-5, 16, 17. Анализ начальных вершин ОПСХ, формирование таблицы ТАБНК.

Блок 6. Нахождение строки таблицы ТАБОР, задавший типы аргументов операции (для сравнения с типами аргументов операции ОПСХ).

Блоки 7-15. Нахождение типов аргументов операции ОПСХ (если вершина, соответствующая аргументу, промежуточная, то ему приписывается тип Ф, если начальная, то соответствующий тип выяснится в массиве ЭТА, после чего проверяются поля РТ и ТАБОР, соответствующие рассматриваемым аргументам и в случае несовпадения типов выдается сообщение.

Блоки 18, 19. Если при анализе аргументов ошибки не обнаружены, то выдается сообщение о корректности соответствия

h.

Результатом работы модуля является текст с указанием правильности соответствия типов аргументов операциям операционной схемы, а также список начальных вершин, не вошедших в массив ЭТА, но имеющих в операционной схеме.

6.2.7. Описание алгоритма модуля ввода массивов РСД, ОТОТ и КОПС.

Модуль предназначен для формирования из массивов БРС, БДП, БОТ базисных родов структур, дополнений и отображений по отображению *h* массивов РСД, ОТОТ и КОПС, используемых в ППП ГРС.

Исходные данные: массивы БРС, БДП, БОТ; массив ЭТА; признак режима (пополнение или начальный ввод), массивы РСД, ОТОТ, КОПС (в случае режима пополнения).

Блок-схема алгоритма приведена на рис. 6.2.6.

В блок-схеме приняты следующие сокращения и обозначения:

РС- род структуры.

Д- дополнение.

ОТ- отображение.

БРС- массивы базисных родов структур

БДП- массив базисных дополнений.

БОТ- массив базисных отображения.

13-2-86

ТПТЗки3

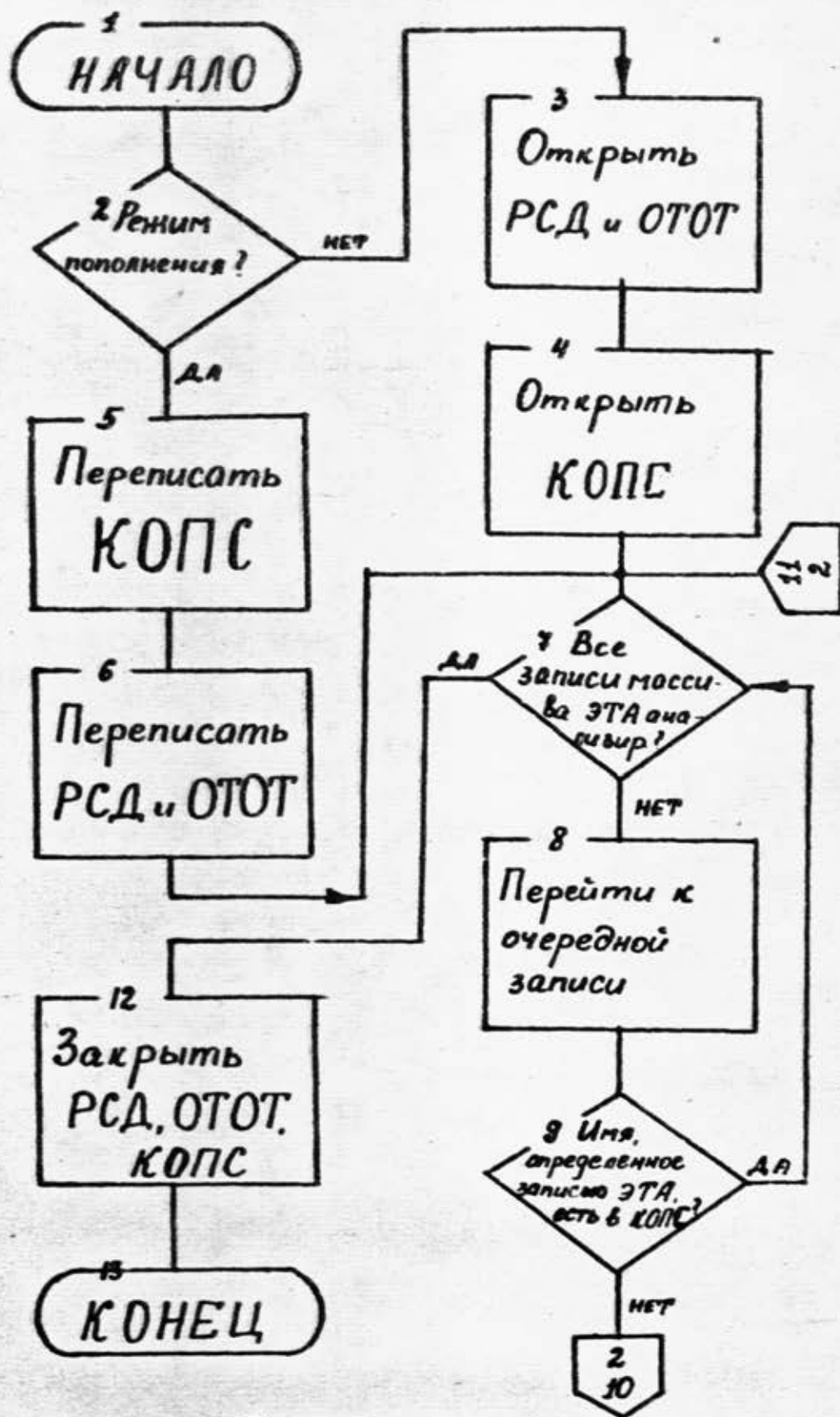


Рис. 6.2.6. (лист 1)

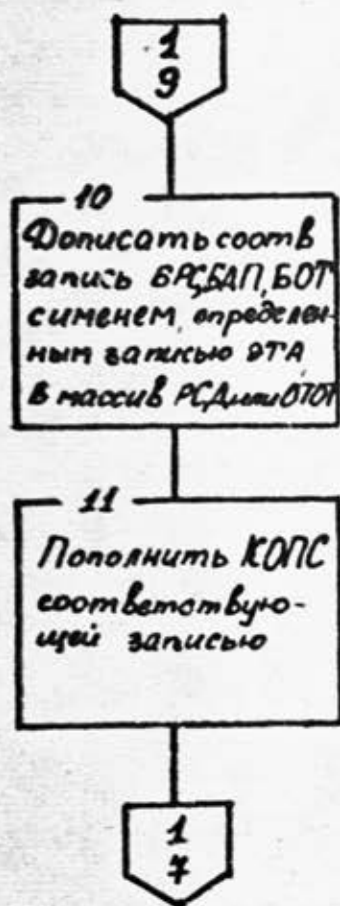


Рис 6.2.6. (лист 2)

РСД- массив родов структур и дополнений.
 ОТОТ- массив отождествляющих отображений.
 КОПС- каталог массива ОПСХ.

Опишем отдельные блоки алгоритма.

Блоки 2-6. В случае режима начального ввода открываются массивы РСД, ОТОТ и КОПС, в случае пополнения- переписываются.

Блок 7-12. Если все начальные вершины ЭТА проанализированы, то массивы РСД, ОТОТ, КОПС закрываются, в противном случае- в массиве ЭТА по очередной записи определяется имя записи, БРС, БДП или БОТ, отвечающей данной вершине операционной схемы и соответствующая запись массива БРС, БДП или БОТ записывается в РСД или ОТОТ, если ее имя не находится в каталоге КОПС (т.е. она не была записана в соответствующий массив раньше).

Результатом работы модуля являются массивы РСД, ОТОТ, КОПС- входные массивы для ППП ГРС.

6.2.8. Описание алгоритма модуля выборки родов структур или дополнений.

Модуль предназначен для выборки из заданных массивов необходимых родов структур или дополнений.

Исходные данные: 1-й аргумент- массив БРС, или массив БДП, или массив РСД; 2-ой аргумент- список имен родов структур или дополнений.

Блок-схема алгоритма не приведена ввиду простоты.

Результатом работы модуля является набор записей, соответствующих требуемым родам структуры или дополнениям, либо сообщение об отсутствии всех или некоторых из них.

6.2.9. Описание алгоритма модуля выборки конститuent рода структуры или дополнения.

Модуль предназначен для выборки из массивов БРС или БДП или РСД конститuent: рода структуры или дополнения.

Исходными данными для работы модуля являются: 1-й аргумент: массив БРС или массив БДП или массив РСД, 2-й аргумент- ^{имя рода} структуры или дополнения, ^{список} 3-й аргумент- кодов конститuent.

Блок-схема алгоритма не приведена ввиду простоты.

Результатом работы модуля является набор записей, соответствующих конститuentам рода структуры или дополнения, либо

сообщение об отсутствии всех или некоторых из них.

6.2.10. Описание алгоритма модуля выборки подграфов из графа конституэнт.

Модуль предназначен для выборки по списку верхних вершин подграфов из графа конституэнт.

Исходными данными являются: 1-й аргумент: идентификатор графа конституэнт, 2-й аргумент - список имен вершин подграфа.

Блок-схема алгоритма не приводится ввиду простоты.

Результатом работы модуля является набор записей, соответствующих требуемым подграфам или сообщение об отсутствии всех или некоторых из них.

6.2.11. Описание алгоритма модуля выборки слоя из графа в расслоенном представлении.

Модуль предназначен для выборки слоя из графа, заданном в расслоенном представлении.

Исходные данные следующие: 1-й аргумент - граф в расслоенном представлении, 2-й аргумент - номер слоя.

Блок-схема данного алгоритма не приводится ввиду простоты.

Результатом работы алгоритма является ^{набор} записей всех подграфов, соответствующих слою либо сообщение об их отсутствии.

6.2.12. Описание алгоритма модуля выборки \mathcal{A} -интерпретации конституэнты.

Модуль предназначен для выборки \mathcal{A} -интерпретации запрашиваемых конституэнт из массива ЗНАЧР.

Исходными данными являются: 1-й аргумент - массив ЗНАЧР, 2-й аргумент - список кодов конституэнт, 3-й аргумент - массив КРВГ.

Сначала код конституэнты ищется в каталоге КРВГ и если она там есть, то по адресу ее значение ищется в массиве ЗНАЧР.

Блок-схема не приводится ввиду простоты.

Результатом действия алгоритма является запись массива ЗНАЧР, соответствующие значениям \mathcal{A} -интерпретации запрашиваемых конституэнт, либо сообщением об отсутствии всех или

некоторых из них.

6.2.13. Описание алгоритма модуля удаления базисных родов структур.

Модуль предназначен для удаления базисных родов структур по списку имен, соответствующих записей из массивов базисных родов структур.

Исходные данные: 1) массив базисных родов структур, 2) список имен удаляемых базисных родов структур.

Блок-схема алгоритма не приводится ввиду ее простоты.

Результатом работы модуля является ^{соответствующих} удаление базисных родов структур или сообщение об их отсутствии.

Контроль всех изменений, связанных с данным удалением лежит на проектировщике или ППП внесения изменений.

Модуль удаления базисных дополнений и базисных отображений аналогичен модулю удаления базисных родов структур, при этом ^{изменяется} лишь исходные данные (т.е. работа идет с другими массивами и списками имен).

6.2.14. Описание алгоритма модуля удаления стандартных подграфов из графа.

Модуль предназначен для удаления подграфов из графа.

Исходные данные: 1-й аргумент: граф конституэнт, 2-й аргумент: список верхних вершин, для которых соответствующие им подграфы графа должны быть удалены.

Блок-схема алгоритма не приводится ввиду простоты.

Модуль удаляет подграфы, верхние вершины которых принадлежат заданному списку, а также получившиеся в результате пустые слои графа (если граф в расслоенном представлении), либо выдает сообщение об их отсутствии.

6.2.15. Описание алгоритма модуля формирования графа конституэнт.

Модуль формирования графа конституэнт (МФГК) предназначен для проверки записи рода структуры или дополнения массива РСД^X и в случае ее правильности построения соответствующего графа конституэнт^{XX} этого рода структуры или дополнения структуры записи массива описана в 1.4.

^{XX} Определение графа конституэнт, соответствующего роду структуры, приведено в 1.1.

15-2-76
ТПЗкиЗ

ния.

(ЗРС)

Контроль записи рода структуры производится по следующим пунктам:

1) в выражениях классов М и К могут быть только конститутенты типов Х и С и должна быть хотя бы одна конститuenta этого типа,

2) в выражениях классов П, А, Т могут быть только конститутенты типов Х, С, Д, П и должна быть хотя бы одна конститuenta одного из этих типов.

3) в i -м выражении класса П_с в выражении не может быть конститuenta П_с с $j \geq i$.

4) число выражений данного класса должно совпадать с соответствующим числом N_j с индексом класса в ЗРС.

Исходные данные: I-й аргумент-ия ЗРС в массиве РСД.

Блок-схема алгоритма представлена на рис. 6.2.7.

В блок-схеме алгоритма приняты следующие обозначения:

В- часть записи РСД^X, в которой записаны все выражения

РСД.

Т- признак правильности ЗРС, (Т=ЗРС- правильная)

К- счетчик числа классов в В,

i - счетчик числа выражений в классе,

j - счетчик числа символов в выражении,

N_j - количество выражений соответствующего класса, записано по классам в РСД.

ВВ- верхняя вершина,

НВ- нижняя вершина,

ПГР- подграф

S - признак наличия стандартных символов (S=I- есть ст.симв.; S=0- нет ст.симв.).

Алгоритм носит четко выраженный структурный характер, в нем выделяются три уровня анализа:

1) анализ В,

2) анализ классов выражений

3) анализ выражений.

Опишем отдельные блоки алгоритма:

Блоки 2, 3, 10, 20, 21- присвоение начальных значений при-

х) структура записи массива РСД описана в I.4.

15-2-76

ТПТЗкиЗ

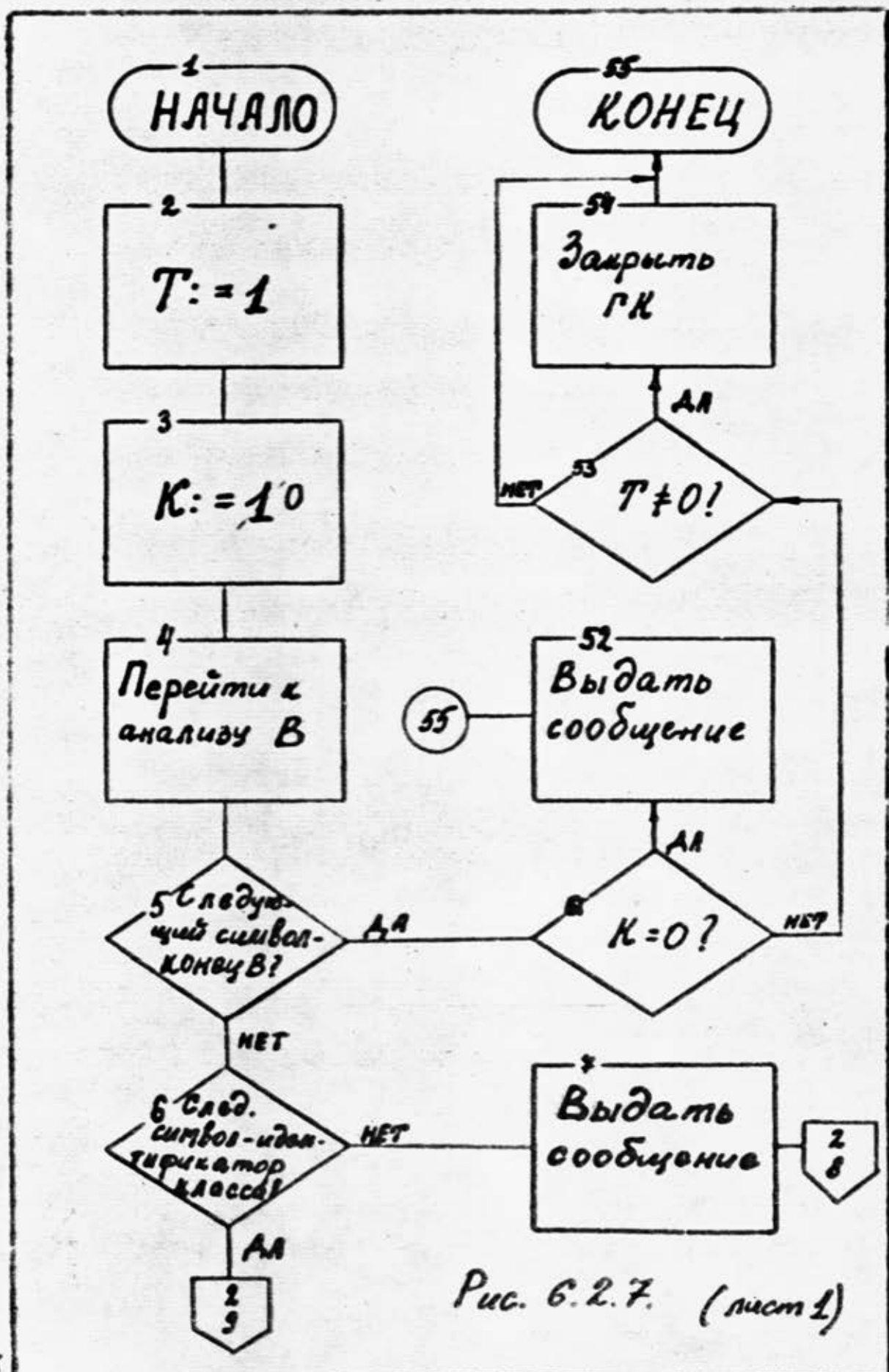


Рис. 6.2.7. (лист 1)

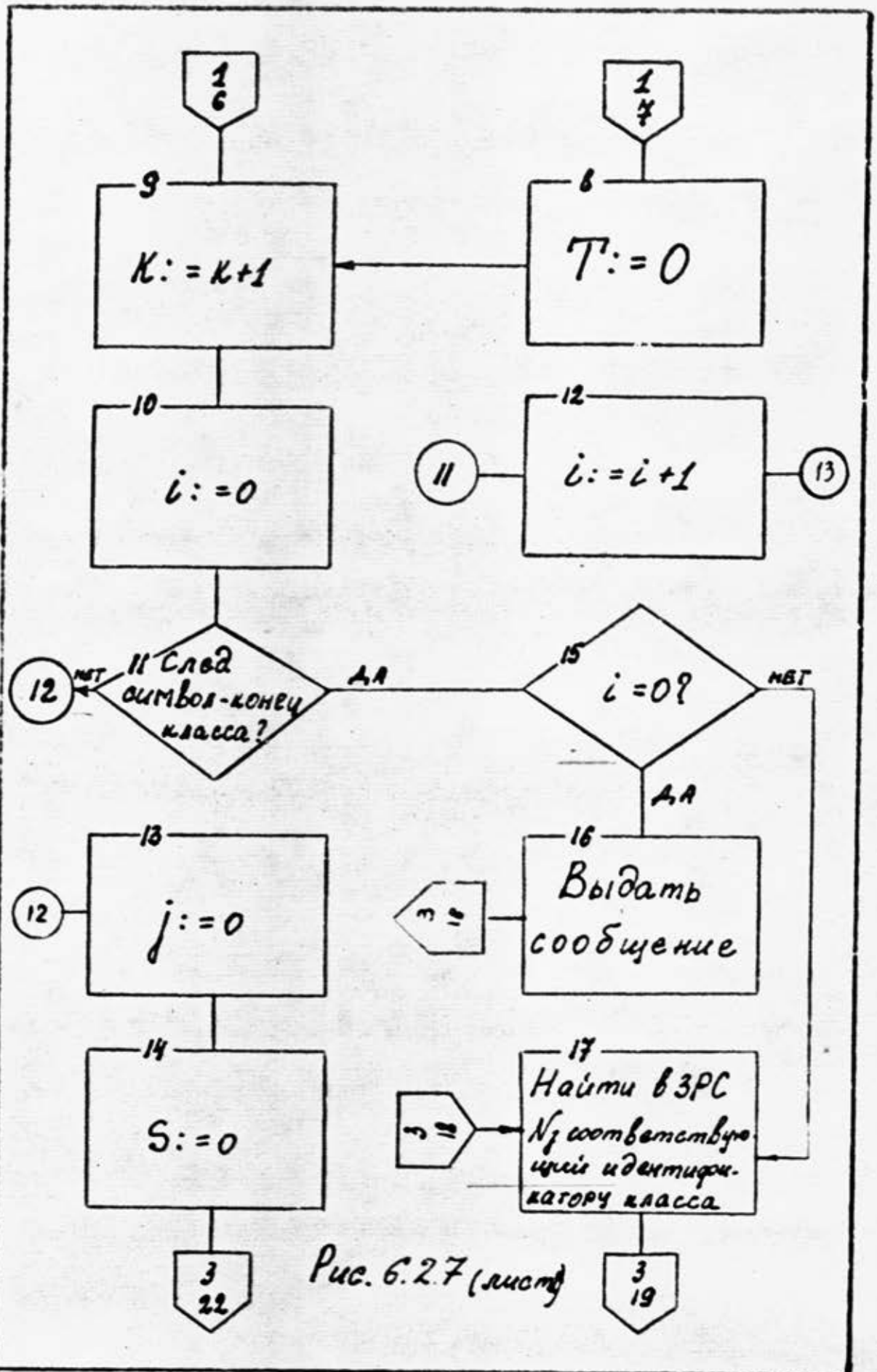


Рис. 6.2.7 (лист)

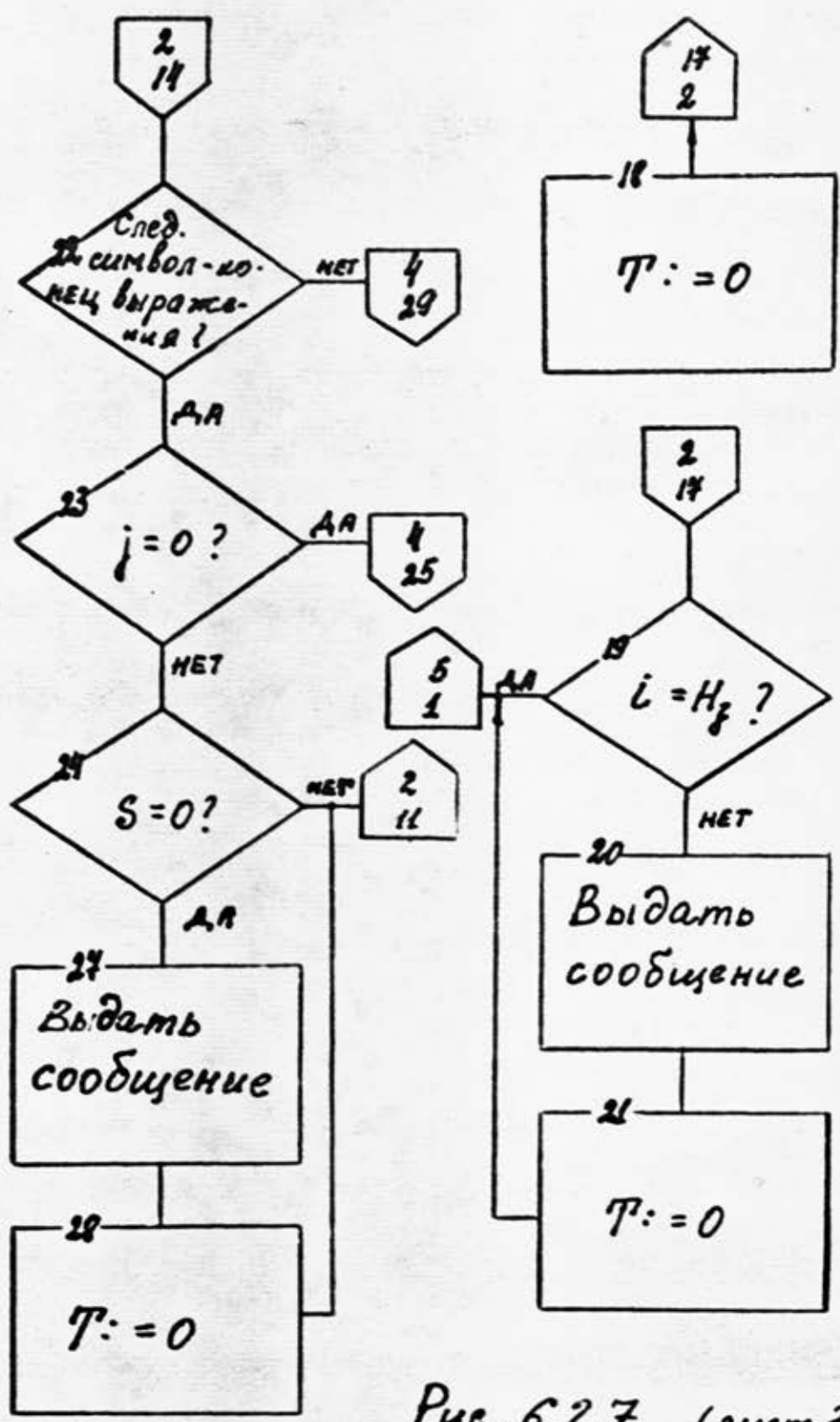


Рис. 6.2.7. (лист 3)

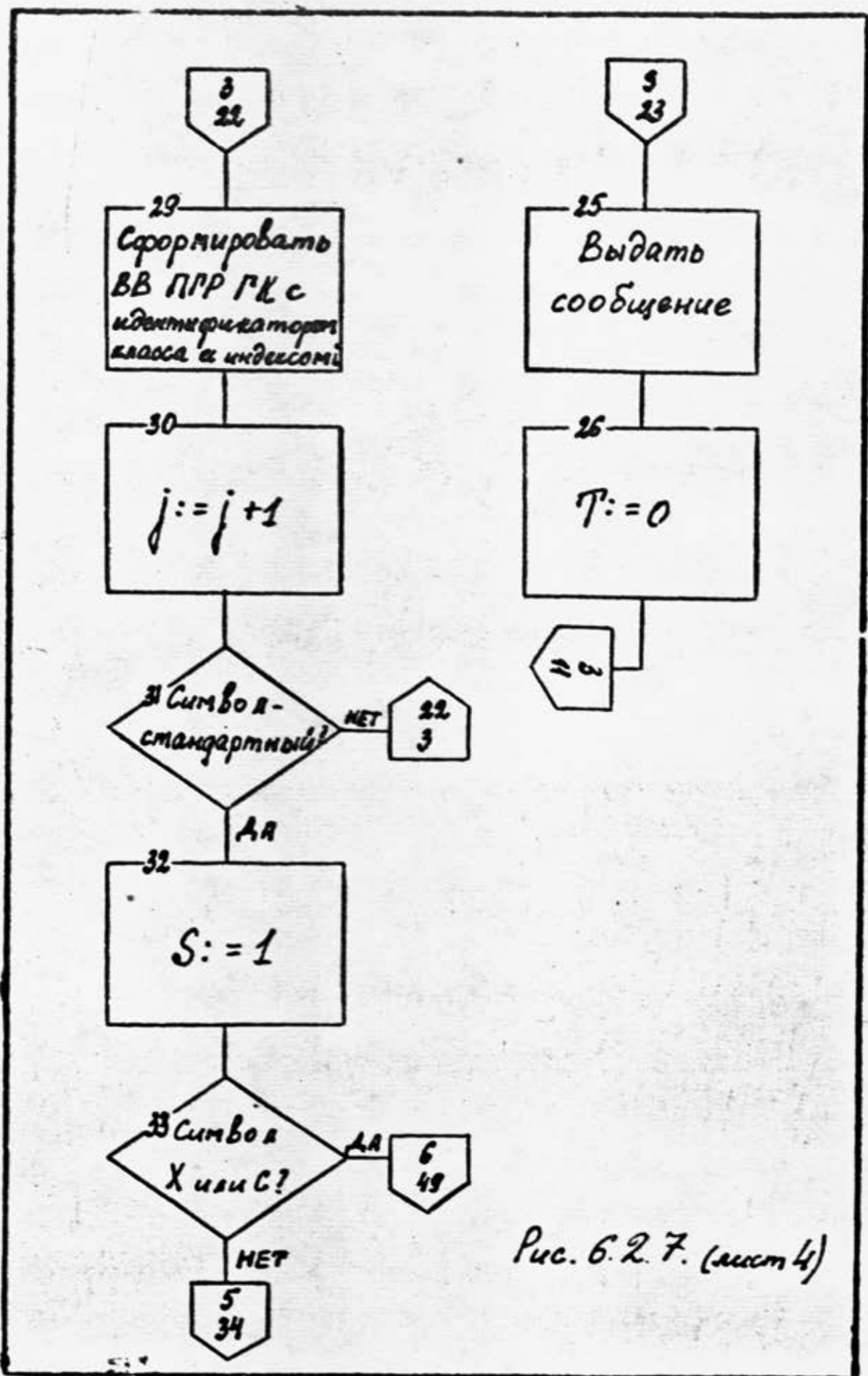


Рис. 6.2.7. (лист 4)

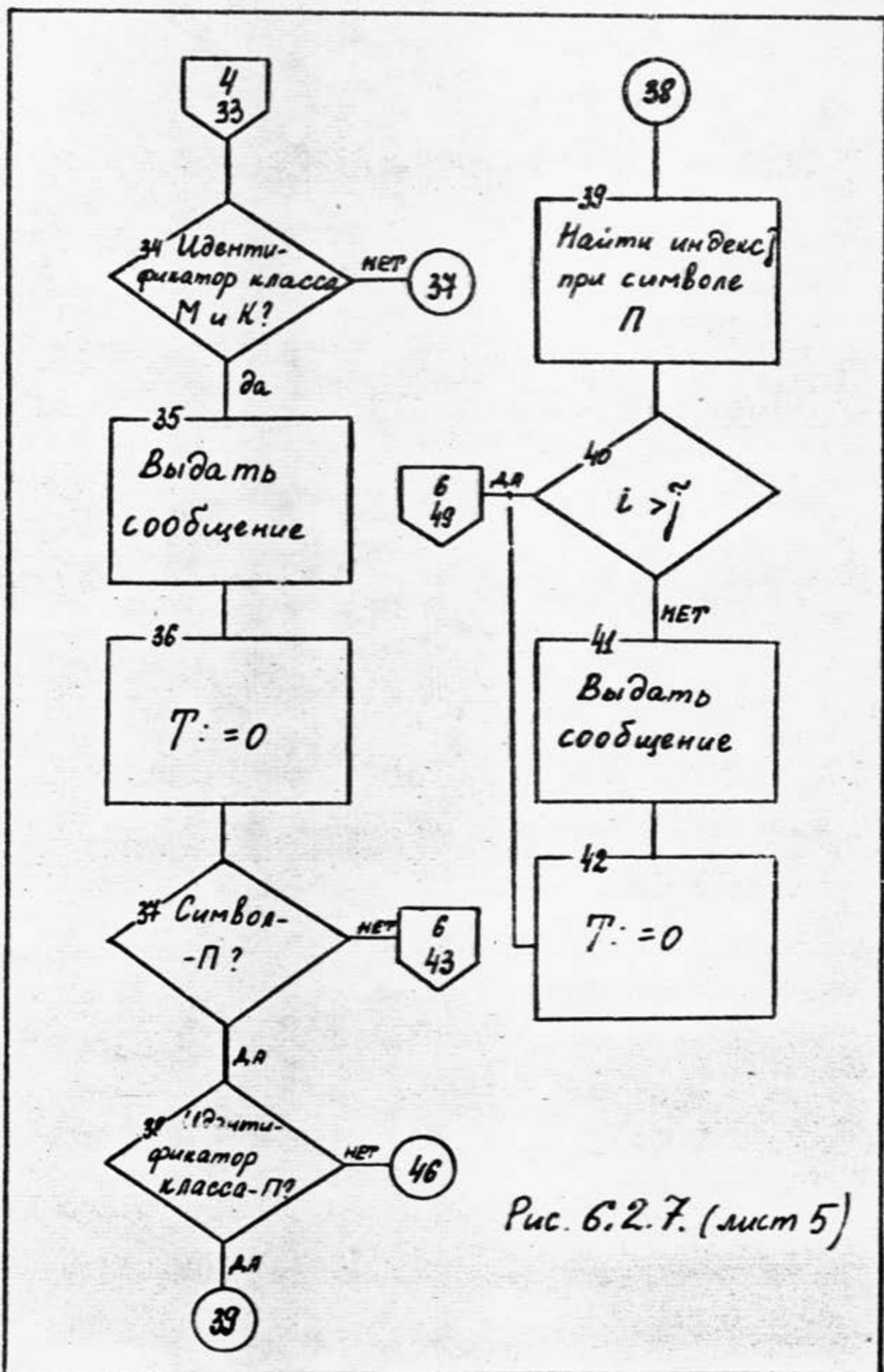


Рис. 6.2.7. (лист 5)

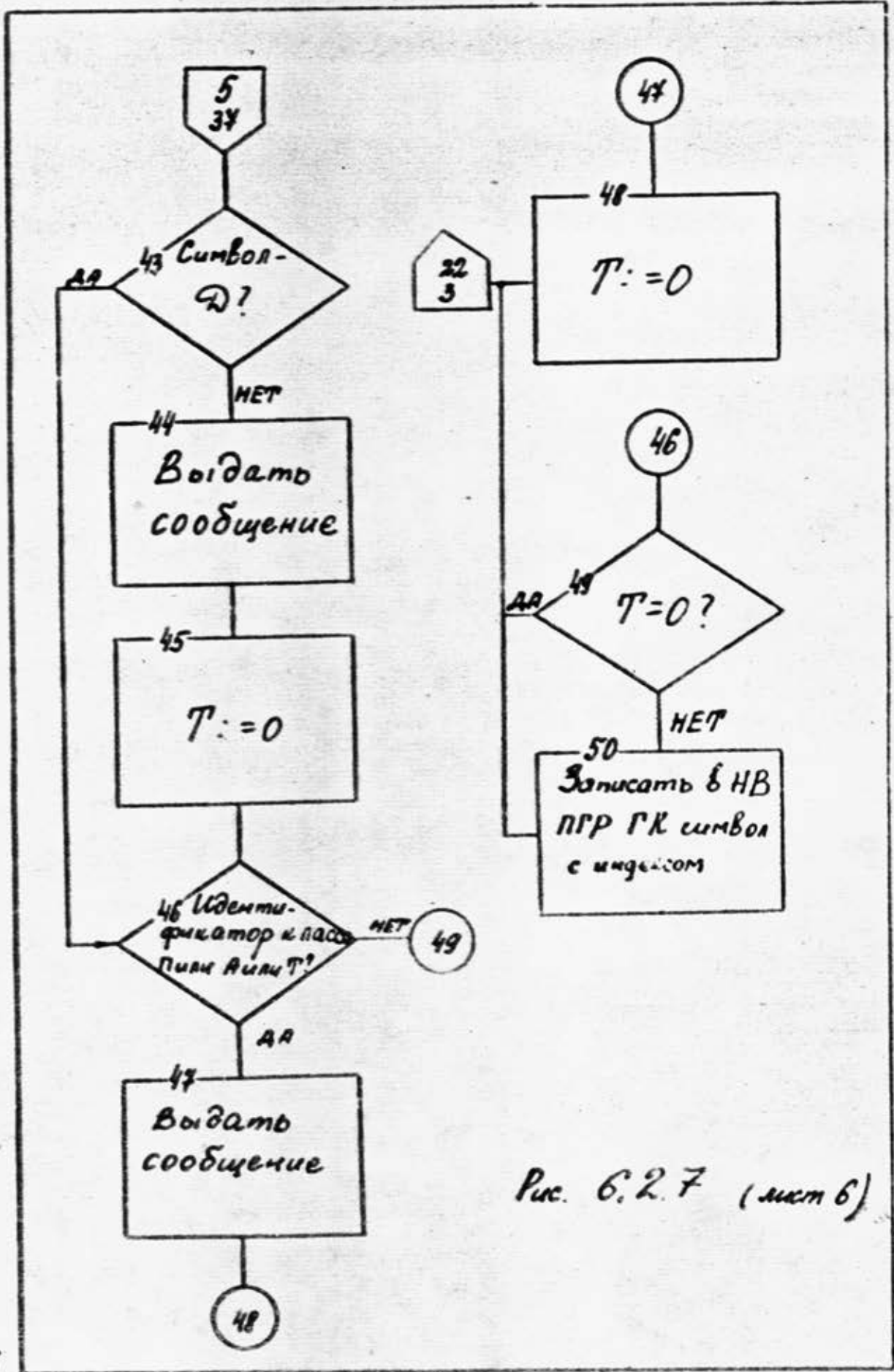


Рис. 6.2.7 (лист 6)

15-2-28
ТПТ.ЗенЗ

накам T, S и счетчикам K, i, j .

Блоки 9, 12, 30. Вычисление текущих значений счетчиков K, i, j .

Блоки 5-10. Переход в очередному классу.

Блоки 51-54-. Обработка конца В.

Блоки 11-14. Переход к очередному выражению.

Блоки 15-21. Обработка конца класса.

Блоки 22, 29-50. Анализ выражения.

Блоки 23-28. Обработка конца выражения.

Блок 29. Формирование верхней вершины подграфа графа кон-
ституэнт, в качестве кода конституэнта записывается идентифи-
катор класса с индексом номера выражения i .

Блок 31. Проверка стандартности символа, т.е. является
ли символ символом типа X, C, D, П.

Блок 39, 50. нахождение индекса при символе сводится к
выявлению всех цифровых символов следующих за символом П.

Блок 7, 15, 23, 24. Выдача сообщений об ошибках, считается
ошибочной ситуация, когда пусто В, пуст класс, а идентифика-
тор класса есть (в случае пустого класса должен отсутствовать
соответствующий идентификатор), выражение пусто или содер-
жит только ^{нестандартные} символы.

Аналогично производится контроль дополнения, соответству-
ющие блоки в алгоритме не представлены.

Результатом работы модуля является граф конституэнт, ес-
ли запись правильная и признак $T=1$, или сообщения об ошибках
и признак $T=0$.

6.2.16. Описание алгоритма модуля приведения графа к расслоенному виду. (МПГРВ).

Модуль приведения графа к расслоенному виду предназначен
для приведения графа к ^{расслоенному} представлению X).

Исходные данные: I-й аргумент- граф.

Блок-схема алгоритма представлена на рис. 6.2.8.

На блок-схеме приняты следующие обозначения:

Кат 1- каталог вершии предшествующих слоев формируемо-
го графа.

Кат 2- каталог ВВ текущего слоя формируемого графа

X) Описание расслоенного представления графа приведено в
2.3.

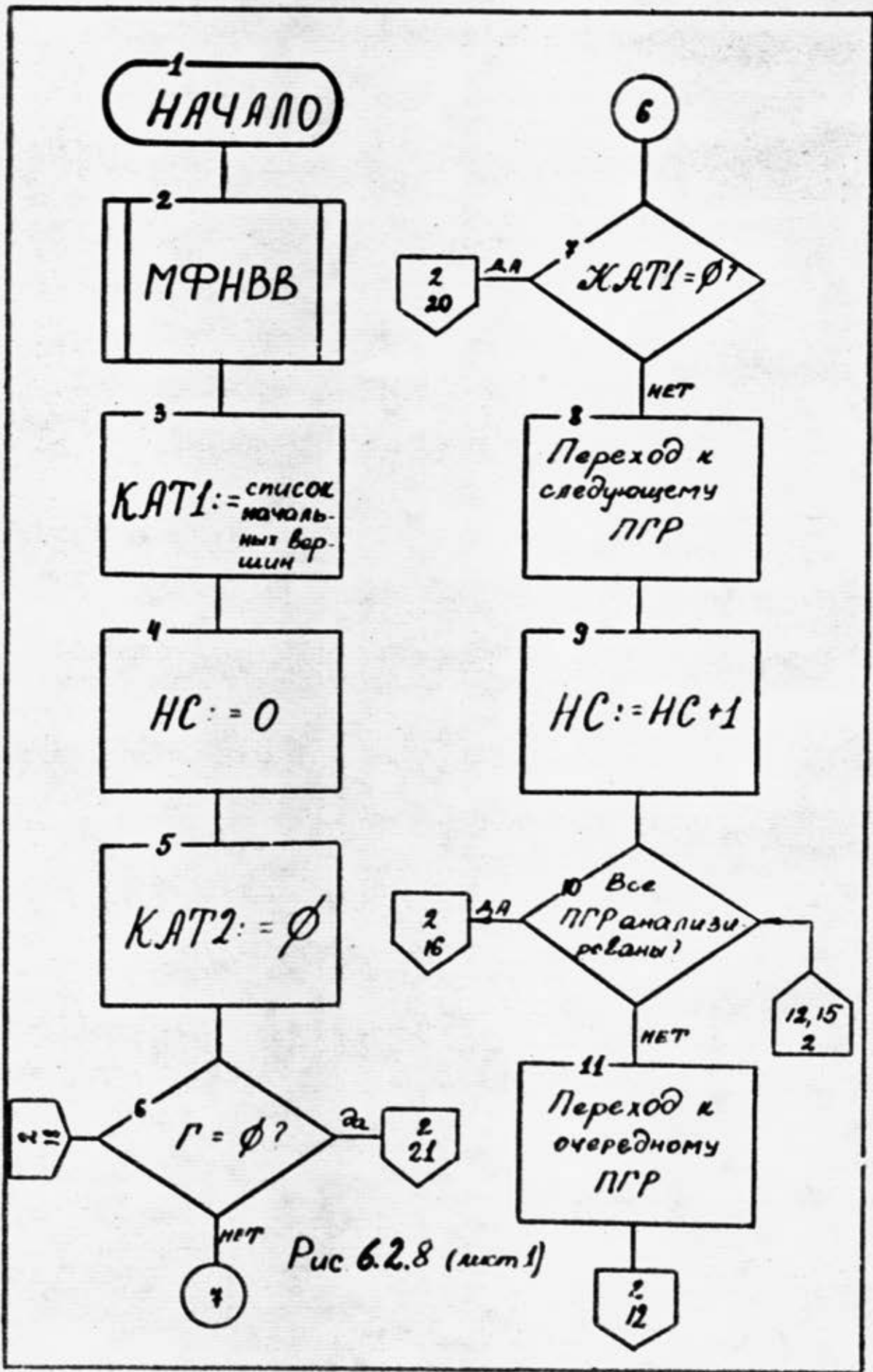


Рис 6.2.8 (лист 1)

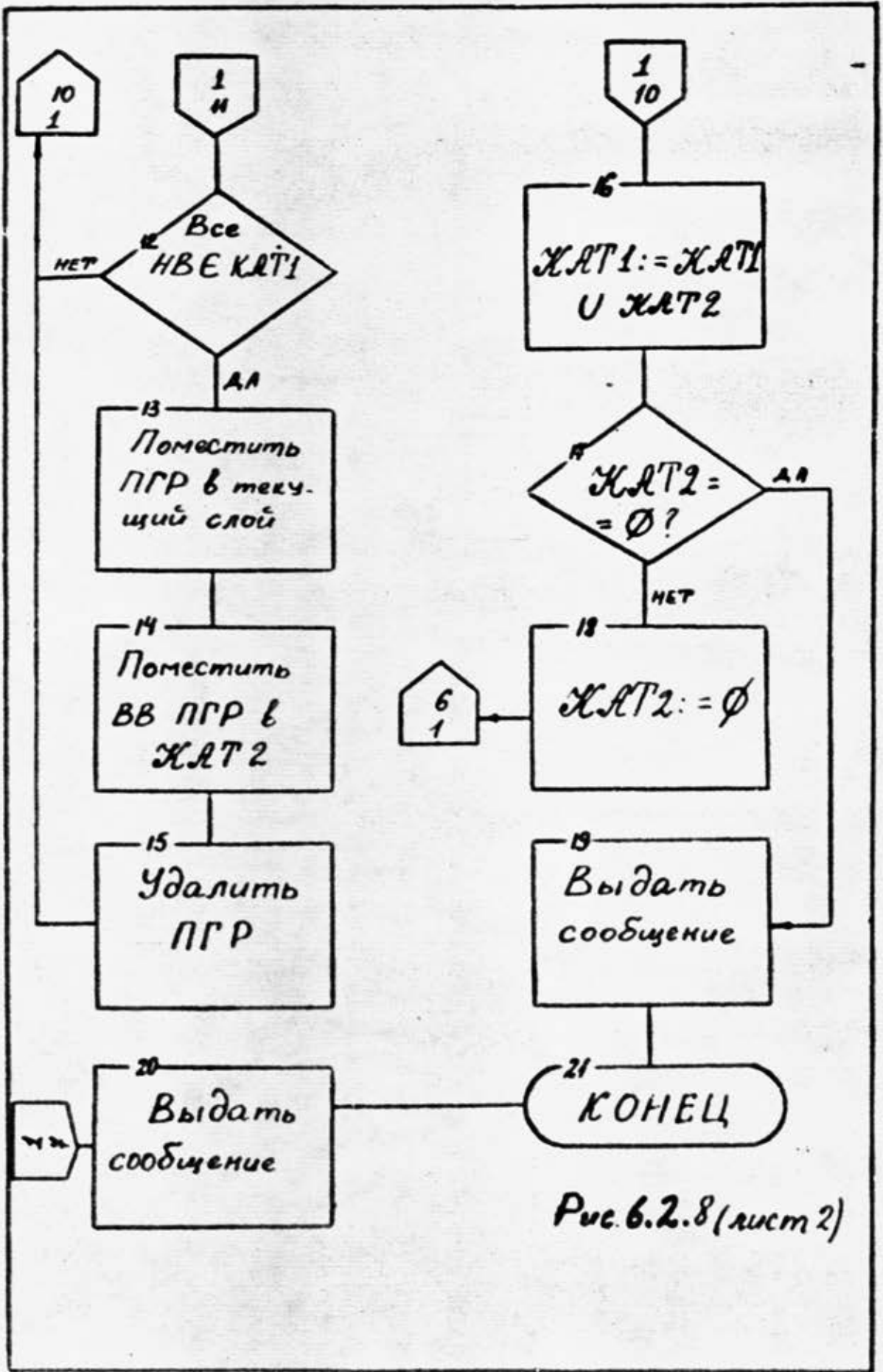


Рис. 6.2.8 (лист 2)

15.2.26
РПТЗМЗ

ВВ- верхняя вершина,

НВ- нижняя вершина,

ПГР- подграф,

НС- счетчик слоев,

Опишем отдельные блоки алгоритма.

Бл. 2. Вызов модуля формирования списка начальных вершин и списка всех вершин графа, модуль передается граф и параметр N_{-I}^{XX} .

Бл. 3. занесение в каталог I списка начальных вершин графа.

Бл. 4. Очистить счетчик слоев.

Бл. 5. Открыть кат. 2.

Бл. 6. Проверить, имеются ли непроанализированные подграфы.

Бл. 7. Проверить, не является ли $кат. I$ пустым, т.е. в графе нет начальных вершин, если их нет, перейти на блок 20.

Бл. 8. Перейти к анализу следующего подграфа.

Бл. 9. Начало формирования следующего слоя.

Бл. 10. Проверить, все ли подграфы проанализированы.

Бл. 11. Перейти к очередному ПГР.

Бл. 12. Проверить, все ли НВ принадлежат кат. I.

Бл. 13. Поместить ПГР в текущий слой.

Бл. 14. Поместить ВВ анализируемого ПГР в кат. 2.

Бл. 15. Удалить ПГР либо логически (т.е. поместить как уже обработанный), либо физически, т.е. удалить из графа.

Бл. 16. Пополнить кат. I списком вершин, вошедших в сформированный слой.

Бл. 17. Проверить, не является ли каталог 2 пустым.

Бл. 18. ^{очистить} кат. 2 для перехода к следующему слою.

Бл. 19. Выдать сообщение: в графе имеются циклы или петли.

Бл. 20. Выдать сообщение: в графе все вершины неначальные, имеются циклы или петли.

Результатом работы модуля является граф в расслоенном представлении или сообщения о наличии в графе циклов и петель и подграф^х графа в расслоенном представлении.

XX) Описание модуля приведено в 6.2.17

X) Здесь подграф- в смысле теории графов (а не только длины I).

5.2.17. Описание алгоритма модуля формирования списка начальных вершин и списка всех вершин графа.

Модуль предназначен для формирования и выдачи списка начальных вершин графа, а также списка всех вершин.

Исходные данные: 1-й аргумент- граф Γ в формализованном представлении, 2-й аргумент- признак N .

Блок-схема алгоритма представлена на рис. 6.2.9.

В блок-схеме приняты следующие обозначения:

Кат 1- каталог вершин графа.

Кат 2- катало вершин графа

ПГР- подграф.

ВВ- верхняя вершина

НВ- нижняя вершина

N -1- признак того, что необходимо выдать список НВ и всех вершин.

Блок 1. Начало работы.

Блок 2. Открыть кат.1.

Блок 3. Открыть кат.2.

Блок 4. Проверка того, есть ли еще непроанализированные ПГР.

Блок 5. Переход к следующему подграфу и анализ его вершин.

Блок 6. Проверка принадлежности ВВ каталогу 1.

Блок 7. Выдача сообщения о том, что повторились ВВ в ГК.

Блок 8. Верхняя вершина из анализируемого подграфа помещается в Кат.1 и Кат.2 (если в кат.2 она не встречалась).

Блок 9. Нижние вершины анализируемого подграфа записываются в кат.2, при этом уже встречавшиеся НВ не заносятся.

Блок 10. Формирование нового кат.2- списка начальных вершин графа.

Блок 11. Формирование кат.1- списка всех вершин графа, записанных в упорядоченной форме.

Блоки 12,13. Выдача на печать сформированных списков.

Блок 14. Проверка признака N .

Результатом работы модуля является список всех вершин графа или список всех начальных вершин и список всех вершин графа.

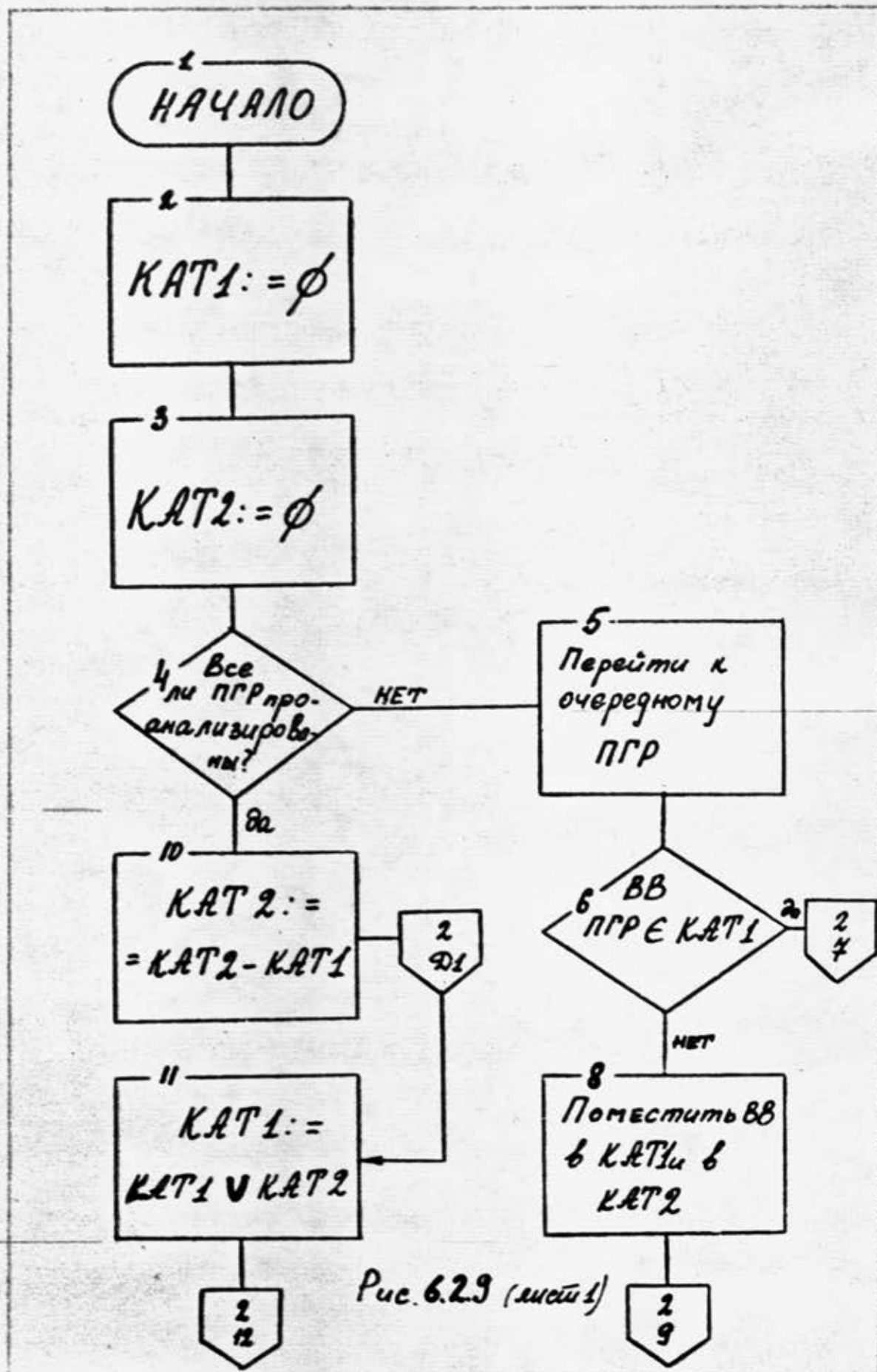


Рис. 6.29 (мат. 1)

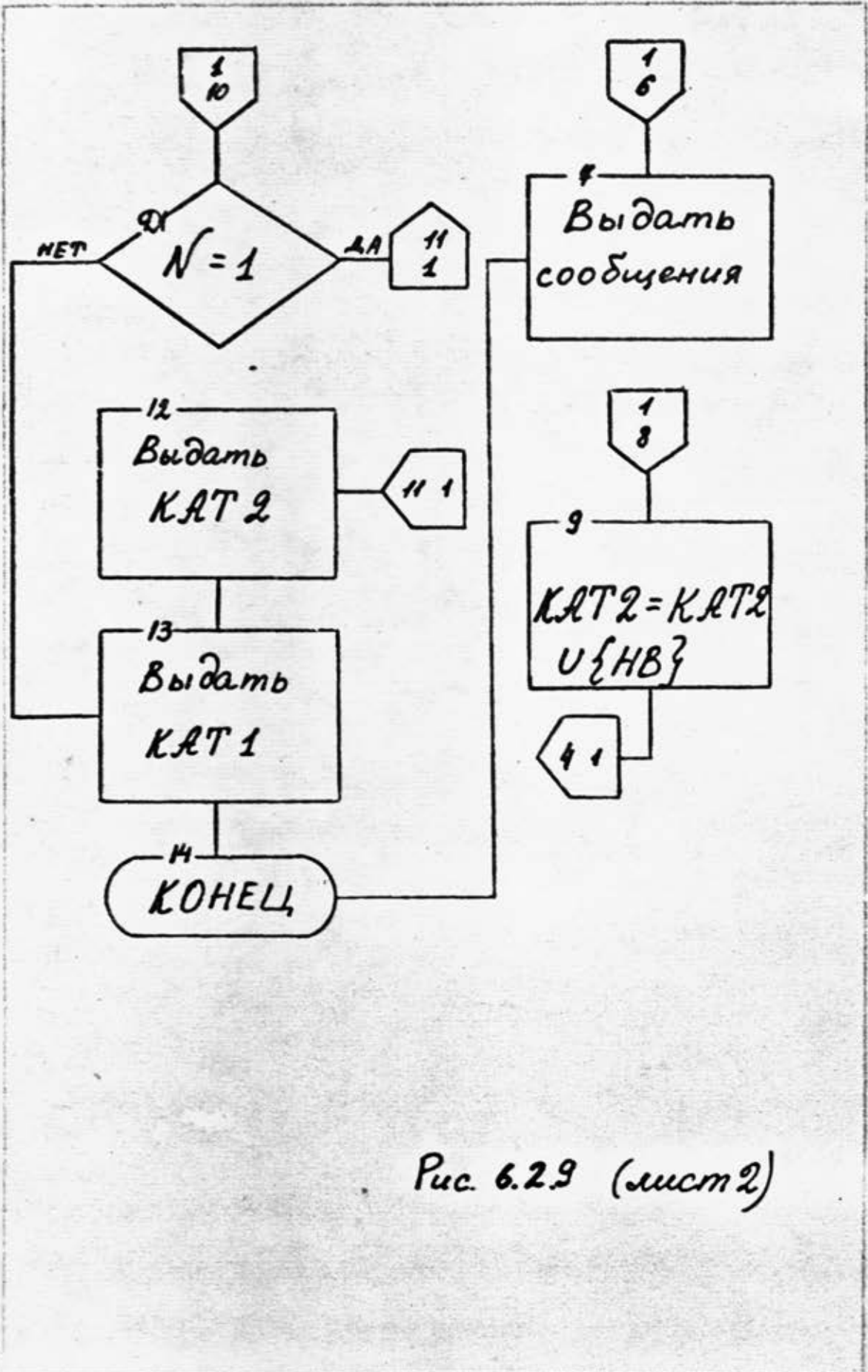


Рис. 6.29 (лист 2)

15-2-76
ТПТЗМЗ

6.2.18. Описание алгоритма модуля формирования списка конечных вершин графа.

Модуль предназначен для формирования и выдачи списка и числа конечных вершин.

Исходные данные: 1-й аргумент- граф в форматизированном представлении, 2-й аргумент- значение ключа К.

На блок-схеме приняты следующие обозначения:

К- признак того, выдавать ли только число верхних вершин, или число верхних вершин и список верхних вершин.

N- число верхних вершин.

ВВ- верхняя вершина,

Г- граф.

Г (U) - нижнее замыкание (Ux).

Блок-схема представлена на рис.6.2.10.

Блок 2. Очистка счетчика N .

Блок 3. Открыть каталог ВВ.

Блок 4. Проверка, не является ли анализируемый граф пустым, т.е. все ПГР обработаны.

Блок 5,6. Переход к очередной ВВ, при этом граф анализируется сверху вниз.

Блок 7. Применение модуля формирования нижнего замыкания передача ему графа Г и вершины U .

Блок 8. Исключение проанализированных ПГР.

Блок 9. Проверка признака К.

Блок 10. Выдача числа ВВ.

Блок 12. Выдача списка ВВ.

Результатом выполнения модуля будет список и число конечных вершин, либо только число ВВ.

6.2.19. Описание алгоритма модуля изменения ориентации ссылок.

Модуль изменения ориентации ссылок предназначен для формирования по списку всех или заданных вершин графа подграфов, указывающих, в каких подграфах исходного графа используются (в качестве нижних вершин) запрашиваемые вершины.

Исходные данные: 1-й аргумент- граф в форматизированном представлении, 2-й аргумент- признак П.

Определение нижнего замыкания приведено в 2.3.

Описание модуля приведено в

15.2.76
ТТТЗенЗ

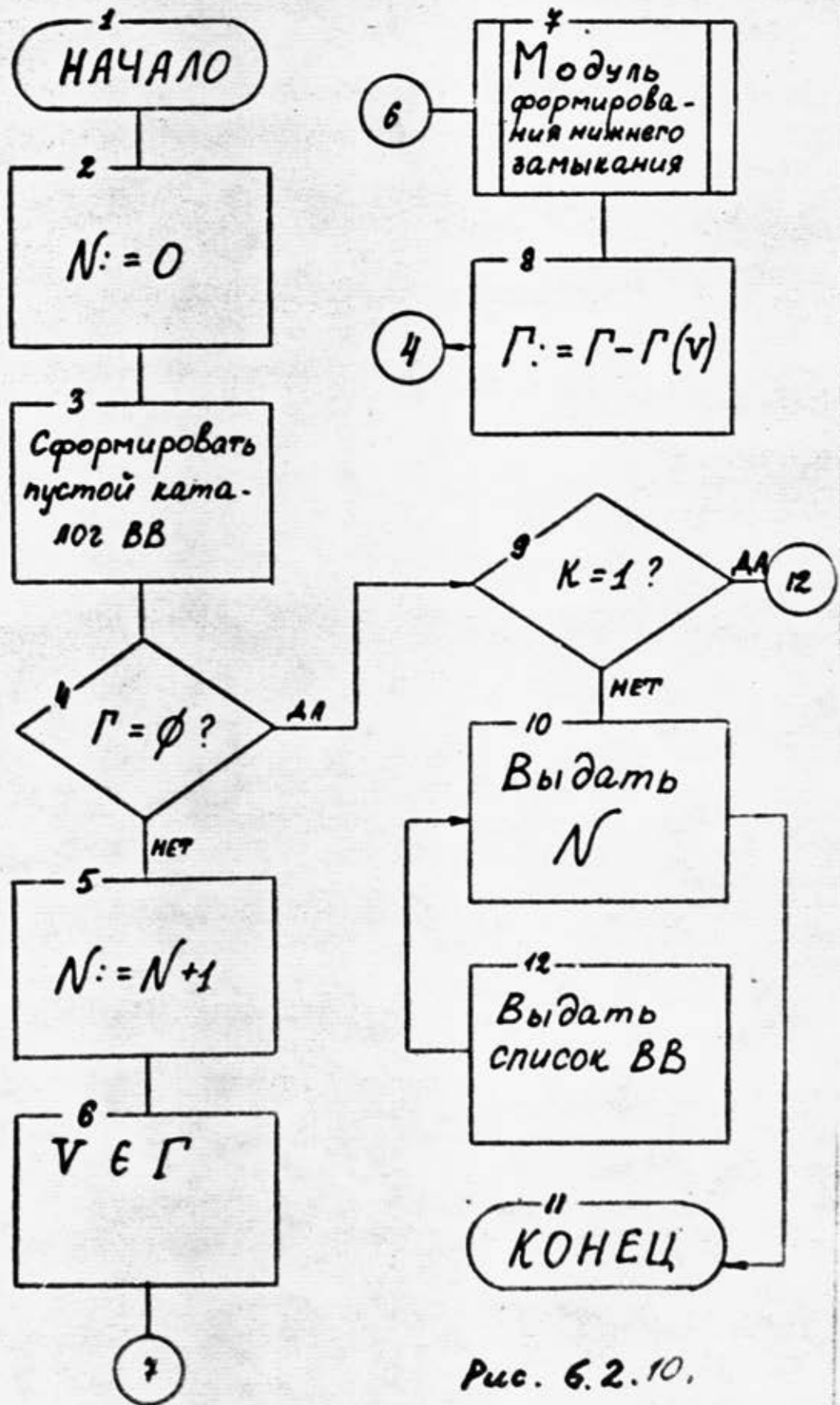


Рис. 6.2.10.

3-й аргумент - список запрашиваемых вершин (отсутствует, если $P=1$), должны быть частью списка всех вершин графа.

Блок-схема алгоритма приводится на рис. 6.2. II.

На блок-схеме приняты следующие обозначения:

P - признак полного ($P=1$) или частичного запроса ($P=0$),

$МНВВ$ - модуль формирования списка начальных вершин и списка всех вершин графа^{x)}.

V - вершина .

$ВВ$ - верхняя вершина .

$НВ$ - нижняя вершина .

$ПГР$ - подграф.

Опишем отдельные блоки алгоритма.

Блоки 2, 3, 4. ^{СМАСМЕННЕ} Внесение режима запроса и соответственно ввод списка запрашиваемых вершин или формирование списка всех вершин ($МНВВ$ передается граф и параметр $N=0$).

Блоки 5-12. Для вершин списка проверяется, совпадает ли она с некоторой $ВВ$ подграфов исходного графа, если совпадает, то соответствующая $ВВ$ приписывается в качестве $НВ$ к формируемому $ПГР$; если же V совпала с некоторой $ВВ$ $ПГР$, то в силу фермативности графа, она не может совпадать с $НВ$ следующих $ПГР$ (анализ проводится сверху вниз) и производится переход к следующей вершине списка.

Результатом работы модуля является список подграфов, указывающих для запрашиваемых (или всех)^{вершин соответ} в каких подграфах они являются нижними вершинами.

^{x)} Описание модуля приведено в 6.2.17 .

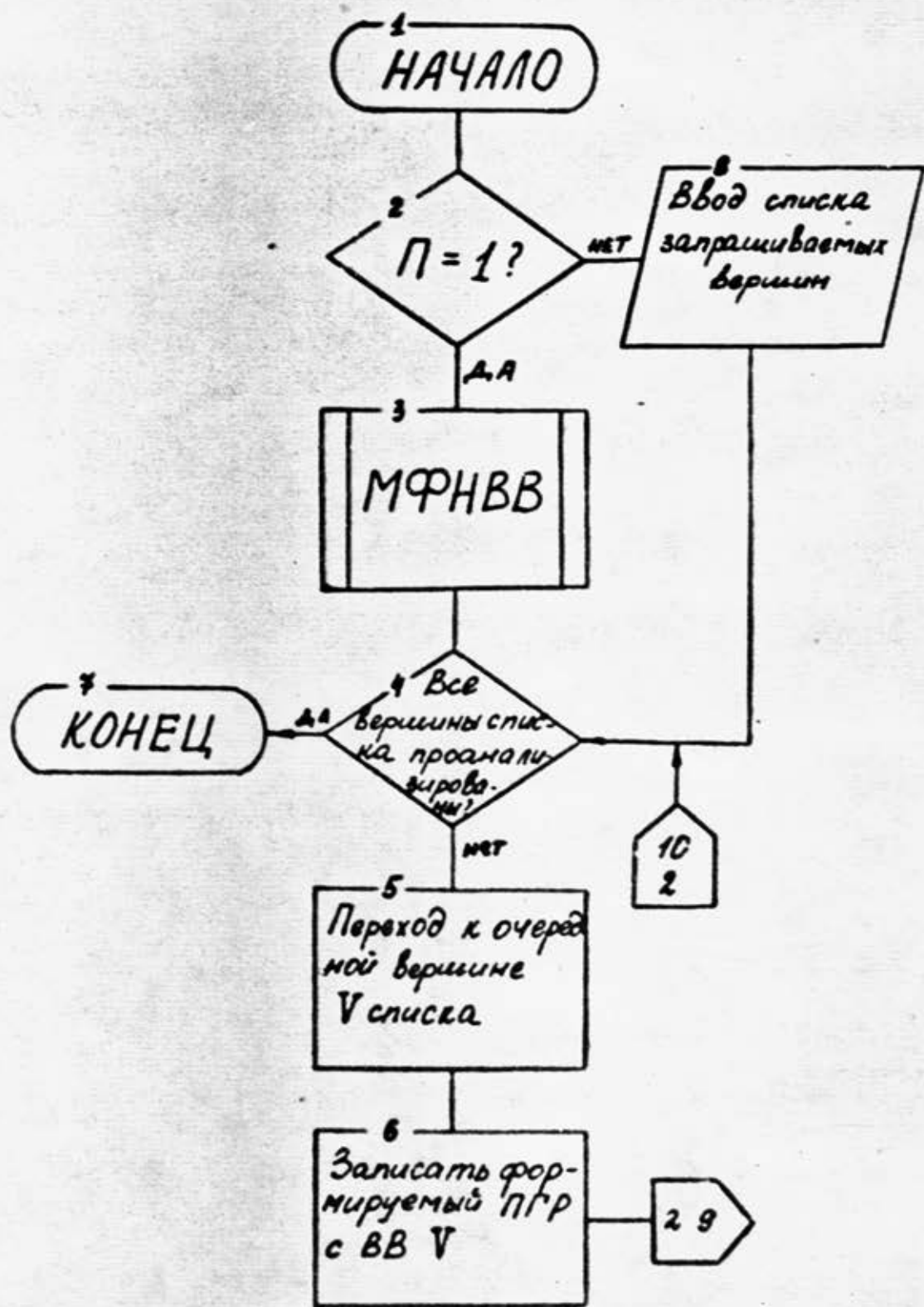


Рис. 6.2.11. (лист 1)

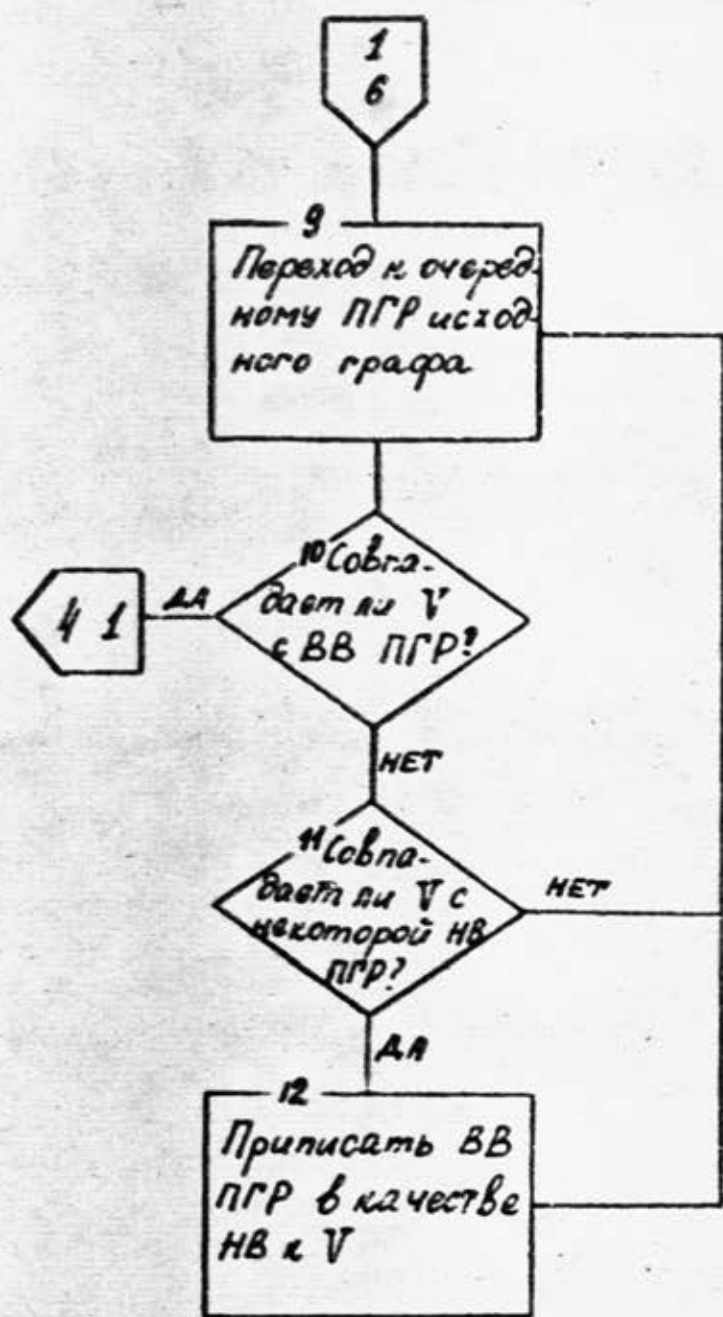


Рис. 6.2.11 (лист 2)

7. Обоснование принятых решений.

7.1. Выбор языков программирования.

В основном, принимается язык PL/I по следующим соображениям.

PL/I представляет собой эффективное объединение языков ФОРТРАН и КОБОЛ и обладает важными свойствами для обработки списковых структур, в том числе:

- а) развитые типы данных и способов их организации, в частности наличие операторов определения длины строки, оператора конкатенации строк и выделения части строки;
- б) возможности выполнения операций над битовыми строками;
- в) хорошие возможности описания сложных структур данных (важно при формировании проекта);
- г) более высокая выраженность по сравнению с языком АССЕМБЛЕРА.

Средства языка PL/I обеспечивают динамическое распределение и резервирование необходимой статической и автоматической и управляемой памяти. Блочная структура языка позволяет фиксировать область действия переменных в рамках отдельных блоков.

Важные свойства для СПП - гибкость доступа к данным и возможность организации рекурсии.

Для эффективной организации работы ГУП, СУБД, мультипрограммного режима весьма важны возможности выполнения в PL/I асинхронных операций и расширяемость макрокоманды времени компиляции.

Транслятор с PL/I имеется в ОС ЕС ЭВМ, в версии 1У.0, требует 50 Кбайт основной памяти, строит весьма эффективные программы, обладающие хорошими диагностическими возможностями.

АССЕМБЛЕР целесообразно использовать для ГУП, УПП и отдельных модулей пакета А - интерпретации. Макроязык, являющийся расширением языка АССЕМБЛЕРА, предоставляет программисту удобные средства для определения последовательности операторов, которые могут использоваться в программе неоднократно. В ОС включаются наборы системных макроопределений для связи запросов на обслуживание с супервизором, для связи с подпрограммами ввода-вывода и для проверки программ.

РЕЗАН - универсальный метаалгоритмический язык рекурсивных функций. Рекурсивные функции языка определяются на множестве произвольных символьных выражений. Структура данных представляет собой скобочные последовательности произвольной сложности. Порядок рассмотрения РЕЗАН объектов задается скобочной структурой. Достоинства языка - мощные возможности рекурсии и простота в изучении. Целесообразно применение для модулей ППП формирования ГРС.

7.2. Обоснование расчетов трудоемкости.

Используются данные, приведенные Д.А.Мартиним в книге "Программирование для вычислительных систем реального времени" М., "Наука", 1975, см. таблицу и Э.А.Трахтенгерца "Программное обеспечение АСУ", М., "Статистика", 1974, см. таблицу

Оценка производительности труда программистов.

| ----- | | :Количество команд на I |
|--|---|-----------------------------|
| ----- | | :программиста в день(усред- |
| ----- | | :ненное по всему периоду |
| ----- | | :написания программ) |
| ----- | | ----- |
| I | : | 2 |
| ----- | | ----- |
| Программы пакетной обработки данных | | 12-15 |
| Прикладные программы | | |
| Подготовка спецификаций | | 20 |
| Кодировка и отладка на основе деталь- | | 20 |
| ных спецификаций: | | |
| Управляющие программы | | |
| Подготовка спецификаций, кодировка и | | |
| отладка программ общим объемом: | | |
| от 1000 до 2000 команд | | 3-5 |
| Обеспечивающие программы | | |
| Средства отладки и управления в ре- | | |
| жиме реального времени (например, | | |
| программы прокрутки, режим анализа, | | |
| кодировка и отладка | | 4-8 |
| Обеспечивающие программы, работающие в | | |
| автономном режиме: спецификации, коди- | | |
| ровка и отладка | | 10-20 |
| ----- | | ----- |

Ниже приведены характеристики производительности работы программистов по составлению программ в зависимости от их сложности.

Таблица

| Характеристики программы | Среднее число команд в день |
|--|-----------------------------|
| Автономные программы объемом до 10000 команд, составленные разработчиками алгоритмов | 41 |
| Автономные вычислительные программы объемом до 1000 команд, составленные по заданному алгоритму | 27 |
| Частично взаимосвязанные задачи по разработке программного обеспечения. Объем программы до 1000 команд | 11 |
| Взаимосвязанные программы обеспечения объемом до 3000 команд каждая | 6 |

7.3. Обоснование общей архитектуры.

Система ППП с единым управлением и единой информационной базой (ЕИ) выбрана в соответствии с требованиями системного программирования, в частности:

- возможности простой сегментации программ за счет их модульной структуры;
- упрощения процедур составления и отладки программ;
- выделения задач основных функциональных блоков проектирования;
- эффективности контроля сохранности входной, промежуточной и выходной информации;
- максимальное использование средств операционной системы, позволяющей осуществить режим пакетной обработки, режим мультипрограммирования и повысить эффективность выполнения трудоемких операций ввода-вывода и отладки.

ППП представляют собой совокупность алгоритмических и программных средств с проблемной ориентацией на решение классов задач. ППП - мощное средство автоматизации программирования, рассчитано на определенного потребителя, функционирует под управлением операционной системы и относится к ППП общего назначения. ППП по системной организации относятся к сложной струк-

15-2-26

ТПТЗмЗ

туре, так как содержит ведущую программу (УПП) и обслуживающие программы.

Специальная часть системы ППП может возможным автономно-последовательное функционирование отдельных пакетов, обеспечивает возможность оперативного внесения изменений и допускает развитие системы.

15-2-76
ТМЗ мЗ

ПРИЛОЖЕНИЕ

15-2-76
ТМЗ-3

Номенклатура модулей системы ШП.

- 1.1. Модуль организации процесса формирования ГРС.
 - 1.2. Модуль, выполняющий операции свободного, прямого и смешанного производства.
 - 1.3. Модуль, выполняющий операции усиления и родового усиления.
 - 1.4. Модуль, выполняющий операции простого и общего расширения базы.
 - 1.5. Модуль, выполняющий операции терм-вложения.
 - 1.6. Модуль, выполняющий операции конкретизирующего вложения.
 - 1.7. Модуль, выполняющий операции булеанизации.
- 2.1. Модуль организации вычислительного процесса.
 - 2.2. Модуль автоматической \mathcal{R} -интерпретации.
 - 2.3. Модуль формирования графа конституэнт.
 - 2.4. Модуль формирования нижнего замыкания.
 - 2.5. Модуль удаления подграфа.
 - 2.6. Модуль управления детализацией графа.
 - 2.7. Модуль формирования графа по формуле.
 - 2.8. Модуль операционного замыкания графа.
 - 2.9. Модуль приведения графа к расслоенному виду.
 - 2.10. Модуль удаления дублирующихся подграфов.
 - 2.11. Модуль формирования списка начальных вершин и списка всех вершин графа.
- 3.1. Управляющая программа пакета перевода представлений.
 - 3.2. Модуль одношаговой \mathcal{T} -интерпретации.
 - 3.3. Модуль одношаговой антиинтерпретации.
 - 3.4. Модуль \mathcal{T} - и антиинтерпретации по схемному коду.
 - 3.5. Модуль расшифровки обобщенного схемного кода.
 - 3.6. Модуль ввода задания на перевод представлений.
 - 3.7. Модуль перевода $\text{OOOII} \rightarrow \text{O} \alpha_2 \text{OII}$.
 - 3.8. Модуль перевода $\text{OOOII} \rightarrow \text{IOOII}$.
 - 3.9. Модуль нахождения базового индекса.
- 4.1. Управляющая программа пакета внесения изменений.
 - 4.2. Модуль расширения операционной схемы.
 - 4.3. Модуль сокращения операционной схемы.

154-26

ТПЗКЗ

- 4.4. Модуль замены операндов.
- 4.5. Модуль изменения базового рода структуры (дополнения).
- 4.6. Модуль замены имени конститuent в базовом роде структуры (дополнения).
- 4.7. Модуль изменения отождествляющего отображения .
- 4.8. Модуль сокращения сильного нижнего замыкания.
- 4.9. Модуль дополнительного нижнего замыкания.
- 4.10. Модуль верхнего замыкания.
- 4.11. Модуль исключения вершин из графа \mathcal{R} -интерпретации.
- 4.12. Модуль расширения графа \mathcal{R} -интерпретации.
- 4.13. Модуль расширения множества интерпретируемых конститuent.
- 4.14. Модуль сокращения множества Rel .
- 4.15. Модуль расширения множества автоматически интерпретируемых конститuent.
- 4.16. Модуль сокращения множества разрешающих термов.
- 4.17. Модуль расширения множества разрешающих термов.
- 4.18. Модуль изменения операционного заполнения.
- 4.19. Модуль изменения индивидуальных имен конститuent.
- 4.20. Модуль изменения сокращенных схемных кодов.

- 5.1. Главная управляющая программа.

- 6.1. Управляющая программа СУБД.
- 6.2. Модуль ввода и контроля базисных родов структур и дополнения.
- 6.3. Модуль ввода базисных отождествляющих отображений.
- 6.4. Модуль ввода и контроля операционной схемы.
- 6.5. Модуль ввода и контроля отображения \mathcal{Z} .
- 6.6. Модуль контроля типов аргументов операции операционной схемы.
- 6.7. Модуль ввода массивов РСД, ОТТ и КОПС.
- 6.8. Модуль выборки родов структур или дополнения.
- 6.9. Модуль выборки конститuent рода структуры или дополнения.
- 6.10. Модуль выборки подграфов из графа конститuent.
- 6.11. Модуль выборки слоя из графа в расслоенном представлении
- 6.12. Модуль выборки \mathcal{R} -интерпретации конститuent.

5-22
 ТПЗсз3

- 6.13. Модуль удаления базисных родов структур.
- 6.14. Модуль удаления стандартных подграфов из графа.
- 6.15. Модуль формирования графа конституэнт.
- 6.16. Модуль приведения графа к расслоенному виду.
- 6.17. Модуль формирования списка начальных вершин и списка всех вершин графа.
- 6.18. Модуль формирования списка конечных вершин графа.
- 6.19. Модуль изменения ориентации ссылок.
- 6.20. Модуль ввода отображения γ .
- 6.21. Модуль ввода индивидуальных представлений конституэнт.
- 6.22. Модуль ввода сокращенных представлений конституэнт.
- 6.23. Модуль ввода видов представлений конституэнт.
- 6.24. Модуль вывода представлений конституэнт.
- 6.25. Модуль вывода операционной схемы.
- 6.26. Модуль вывода родов структур и дополнений.
- 6.27. Модуль вывода отображения η и базисных отождествляющих отображений.
- 6.28. Модуль ввода списков R_{int}, S_{An} .
- 6.29. Модуль вывода списка Rel и списка начальных вершин.
- 6.30. Модуль ввода списка S_{int} и Alg .
- 6.31. Модуль вывода графа конституэнт.
- 6.32. Модуль ввода значений специнтерпретации конституэнт.
- 6.33. Модуль вывода значений специнтерпретации конституэнт и запросов на специнтерпретацию и пополнение библиотеки модулей.

Б-1-26
ТПТЗ-13