МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ

(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

--------------------------------------------------------

Кафедра 22

Пояснительная записка к

КУРСОВОЙ РАБОТЕ

на тему

"Работа с таблицами общего пользования

на этапе лексического анализа языка Норма"

студента группы К7-02а

Жучкова Александра Викторовича

Научный руководитель:

Комиссия:

Оценка:

Москва 1995г.

1. ВВЕДЕНИЕ

Задание, полученное мной на УИР и КП в данном семестре состояло

в следующем:

- ознакомиться с языком программирования Норма и изучить его;

- изучить структуру транслятора с языка программирования Норма;

- ознакомление с языком пограммирования РЕФАЛ;

- разработать структуру данных для реализации работы с таблицами

общего пользования;

- написать функции работы стаблицами.

2. Общее описание языка Норма

Язык программирования Норма является декларативным (непроцедурным)

языком и предназначен для спецификации численных методов решения задач

математической физики. Изначально он был ориентирован на решение задач

математической физики разностными методами, однако может быть использован

для решения более широкого класса вычислительных задач.

Типичный процесс решения задачи из области математической физики

состоит из следующих этапов.

1. Постановка задачи. Выходом этого этапа является обычно система

дифференциальных уравнений, описывающих задачу.

2. Выбор пространственно-временной сетки и дискретизация уравнений с

помощью одного из разностных методов.

3. Выбор метода решения дискретных уравнений. В результате получаются

формулы (соотношения), описывающийе необходимые вычисления в узлах сетки.

4. Программирование полученных формул на некотором языке, который

обеспечивает решение задачи на вычислительной машине.

Главная идея, положенная в основу языка Норма, заключается в том,

что полученные специалистом в процессе решения прикладной задачи расчетные

формулы почти непосредственно используются для ввода в вычислительную систему

и проведения счета.Таким образом, язык Норма дает прикладному математику

возможность сформулировать свою задачу в привычных для него терминах.

Организация процесса вычислений с учетом архитектуры ЭВМ (возможностей

параллельной, векторной обработки и т. п.) - зто задача транслятора с языка

Норма.

Существенным фактом является возможность реализации одной программы

на языке Норма различными вычислительными процессами. Именно разработка

алгоритма с характеристиками, близкими к оптимальным и эффективно

учитывающими особенности конкретных ЭВМ, является наиболее узким местом

создания высококачественного программного обеспечения.

Запись на языке Норма - это, по существу, строгая запись численных

методов решения математической задачи, запись еще не алгоритмов, а просто

расчетных формул и остальной необходимой информации, которую необходимо

знать, чтобы написать программу для ЭВМ.

Отметим, что в записи на Норме не требуется никакой информации о

порядке счета, способах организации вычислительных (циклических) процессов.

Порядок предложений языка может быть произвольным - информационные

взаимосвязи будут выявлены и учтены при организации процесса счета

транслятором.

Выбор уровня языка Норма определяет характерную его черту - в этом

языке нет необходимости вводить такие понятия, как оператор присваивания и

возможность переприсваивания значений (типа х:=х+1) и операторы перехода.

Наличие таких понятий в традиционных языках программирования объясняется

необходимостью формулировки конкретного алгоритма с учетом вопросов

экономии и распределения памяти, порядка выполнения операторов и т. п.

Побочный эффект в языке Норма отсутствует по определению.

Понятно, что многие из этих вопросов появляются снова на этапе синтеза

рабочей программы. Однако, здесь они решаются автоматически по строгим

правилам, гарантирующим правильность синтезируемой программы.

Непроцедурность языка Норма позволяет преодолеть еще одну трудность,

связанную с распараллеливанием алгоритма при счете на ЭВМ, допускающих

совмещение операций. Известные методы распараллеливания последовательных

алгоритмов основаны на выявлении, при некоторых ограничениях, частей

алгоритма, которые можно выполнять независимо, в соответствии с заданным

критерием параллелизма - асинхронные вычисления, синхронные и т. п. Однако,

выявление взаимосвязей в уже сформированном последовательном алгоритме

является неестественной и трудной задачей, так как анализируемая

формулировка, как правило, насыщена избыточными взаимосвязями (типа введения

рабочих переменных для экономии памяти, конкретных способах организации

циклов и т. п.). Вообще говоря, ни откуда не следует, что последовательный

алгоритм надо транслировать в параллельный, а не определять параллельный

сразу по непроцедурной записи.

Эти свойства, и некоторые другие ограничения, позволяют строго

обосновать разрешимость синтеза выходной программы, так как в достаточно

общей постановке решение этой задачи приводит к значительным математическим

трудностям - она может оказаться NP-полной либо вообще неразрешимой. С

другой стороны, исследования, связанные с разработкой и применением языка

Норма показывают, что имеющиеся ограничения приемлимы с практической точки

зрения.

3 Структура транслятора с языка Норма.

Транслятор с языка программирования Норма уже написан на языке Рефал.

И хотя язык программиорвания Рефал весма удобен для обработки символьной

информации, транслятор написанный на этом языке очень не экономно использует

ресурсы вычислительной машины, а именно оперативную память, что зачастую

правильно написанную программы невозможно оттранслировать из за нехватки

оперативной памяти. Поэтому было решено перевести транслятор с языка

программирования Норма на язык программирования Си, который был выбран по

следующим причинам:

- язык Си позволяет гораздо более эффективно использовать ресурсы

вычислительной машины;

- язык Си универсален и удобен для решения задач системного

программирования - разработке трансляторов, операционных систем, экранных

интерфейсов, инструментальных средств;

- разработчиками языка Норма уже написан интерфейс на языке Си,

позволяющий законченные части транслятора, написанные на Рефале, заменять на

законченные части транслятора, написанные на Си, для отладки транслятора.

В процессе трансляции, решаются как традиционные задачи - лексический

синтаксический, семантический анализ, генерация выходной программы, так и

задачи, определяемые спецификой языка Норма: организация вычислений по

непроцедурному описанию задачи с выявлением возможного параллелизма

вычислений, семантический контроль возможности организации вычислений с учетом

возможностей выходного языка и архитектуры компьютера. Выходными языкоми могут

быть языки Фортран ВП ориентированный на многопроцессорный вариант ЭВМ ЕС-1191

и Фортран JNS. Трансляция проводится каждого раздела, входящего в Норма

программу, проводится автономно: для каждого раздела либо выдается программа

на выходном языке, либо, если были обнаружены синтаксические или семантические

ошибки, выдается сообщение об ошибке, после чего осуществляется переход к

трансляции очередного раздела.

Транслятор с языка программирования имеет следующую структуру:

Вход:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|исходный текст | | Лексический анализ | | Синтаксический анализ |

|программы + | --> |(Выделение лексем, | |и частично семантический|

|опции командной| | гупприровка лексем, | -->| анализ описаний и |-->

|строки | | начальное заполнение | | операторов |

---------------- | таблиц имен и констант)| | (заполнение всех аблиц)|

------------------------- -------------------------

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

| Табл. |--- | | | Табл. |--

| | | | МЕНЕЖЕР ПАМЯТИ | | множеств| |

| | | |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | и т.п. | |

| | | | | |

| | | | | |

|\_\_\_\_\_\_\_| | |\_\_\_\_\_\_\_\_\_| |

---------| |\_\_\_\_\_\_\_\_\_|

Выход:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

|Построение графа | |Органинизация | | Генерация | |Текст |

-->|информационных | --> |параллельных | --> | Фортран- | ->|программы |

|зависимостей опе- | |вычислений | | программы | |на Фортране|

|раторов программы | -------------- ----------- -----------

------------------

На вход лексического анализатора поступает текст исходной программы.На выходе

- отсортированный (по описаниям, операторам и итерациям) список лексем,

начально заполненные таблици имен и констант. Далее этот список поступает на

вход синтаксического анализатора, где происходит разбор конструкций-описаний,

операторов, итераций и заполняются таблицы имен, констант, множеств, описания

операторов и др. Перед началом этапа синтаксического анализа, длямаксимальной

очистки памяти происходит конвертация начальных таблиц имен и констант. Эти

таблицы поступают на вход следующего этапа, где происходит построение графа

информационных зависимостей, который используется на следуещем этапе при

определение порядка вычислений и разделение на параллельные ветви, независящие

друг от друга. На выходе получаем внутренние коды. Внутренние коды поступают

на вход кодогенерации. Выходом кодогенерации является программа на языке

Фортран.

Для обеспечения доступа к верхней памяти,а также для возможного свопинга

все операции с оперативной памятью осуществляются с использованием библиотеки

функций менеджера памяти, написанной сотрудниками института прикладной

математики.

4 Описание и решение задачи.

4.1 Постановка задачи.

В свете вышесказанного перед нами (группой разработчиков) встала задача

напсания транслятора с языка Норма с использованием инструментальных средств

языка программирования Си и библиотеки функций работы с операивной памятью.

Передо нами (двумя разработчиками) была поставлена задача написания функций

работы с таблицами имен и констант.

4.2 Решение задачи.

Для начального заполнения таблиц имен и констант были выбраны хэш-таблици.

При этом мы проигрывали в использовании оперативной памяти, но выигрывали во

времени (по сравнению с "деревьями"). Учитывая повышенные требования к

использованию памяти нами было рпинято решение по окончанию этапа лексического

анализа (т.е. после того как завершится предварительное заполнение таблиц имен

и констант и нам будет вточности известно количество идентификаторов и констант

используемых в транслируемой программе) провести конвертацию поисковой части

хэш-таблиц в непрерывные массивы фиксированной длины. При этом мы увеличиваем

обшее время трансляции за счет времени конвертации, но выигрываем в использовании

оперативной памяти за счет уменьшения числа информационных полей (в хэш-таблице

сушествует поле содержащее код идентификатора(константы), который является

ключом этого идентификатора(константы) в дальнейшей работе транслятора, в

масиве роль ключа будет играть смешение зтого элемента от начала массива -

индекс масива), а так же за счет освобождения неисопльзуемых учасков в

хэш-таблице (в любой хэш-таблице из за случайности процесса хэширования

появляются неиспользованные области приблизительно 30% от величины всей таблици).

Плюсом также является умьшение времени поиска по ключу. В масиве эта операция

эквавалентна взятию индекса от масива. В хэш-таблице поик гораздо дольше.

4.3 Выбор структуры данных

3. ЗАКЛЮЧЕНИЕ

В результате проделанной работы мною были достигнуты следуюющие цели:

- ознакомилась и изучила непроцедурный язык программирования Норма, предназначенный для записи численных методов решения задач математической физики разностными методами;

- изучила структуру транслятора с языка программирования Норма;

- ознакомилась с методами лексического анализа;

- изучила структуру лексического анализатора;

- разработала структуру данных своей части задания для реализации лексического анализатора;

-написала функцию, на вход которой поступает список лексем, а на выходе получаем список списков. Элементом этого списка списков является список лексем, который представляет собой одно предложение программы, написанной на Норме, оканчивающееся точкой. Т.е. эта функция "режет" поступающий на вход список лексем по точкам, а на выходе получаем список предложений, каждое из которых заканчивается точкой и представляется списком лексем.

Программа находится на стадии разработки. Завершение работы планируется в следующем семестре. Задание на УИР и КП выполнила полностью.

Список литературы:

А.Н. Андрианов, К.Н. Ефимкин, И.Б. Задыхайло, Н.В. Поддерюгина "Язык Норма"

А.Н. Андрианов, К.Н. Ефимкин, И.Б. Задыхайло "Непроцедурный язык Норма и методы его реализации"

А.Б. Бугеря "Реализация математических функций языка Норма для распределенных высислительных систем"

Приложение 1.