Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Компилятор языка Oberon»

Студент группы **ИУ7-11М** _____ **Е.В. Брянская**
(Подпись, дата) (И.О.Фамилия)

Руководитель _____ **А.А. Ступников**
(Подпись, дата) (И.О.Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Компилятор – программное обеспечение, которое переводит поданный на вход текст программы, написанный на одном из языков программирования – исходном, в машинный код для исполнения на компьютере. В процессе преобразования команд выполняется оптимизация кода и анализ ошибок, что позволяет улучшить производительность и избежать некоторых сбоев при выполнении программы. [1]

Целью данной курсовой работы является разработка компилятора языка Oberon. Компилятор должен выполнять чтение текстового файла, содержащего код на языке Oberon и генерировать на выходе программу, пригодную для запуска.

Для достижения цели необходимо решить следующие задачи:

- проанализировать грамматику языка Oberon;
- изучить существующие средства для анализа исходных кодов программ, системы для генерации низкоуровневого кода, запуск которого возможен на большинстве из используемых платформ и операционных систем;
- реализовать прототип компилятора.

1 Аналитическая часть

1.1 Составляющие компилятора

Компилятор состоит из трёх частей.

1) *Frontend* преобразует текст программы на исходном языке во внутреннее представление, состоит из четырёх частей: препроцессор, лексический и синтаксический анализаторы, генератор внутреннего представления.
2) *Middle-end* занимается машинно-независимой оптимизацией полученного внутреннего представления.
3) *Backend* выполняет преобразование внутреннего представления в программу на языке целевой платформы (ассемблер или машинный код).

На рисунке 1.1 изображены основные фазы работы компилятора.

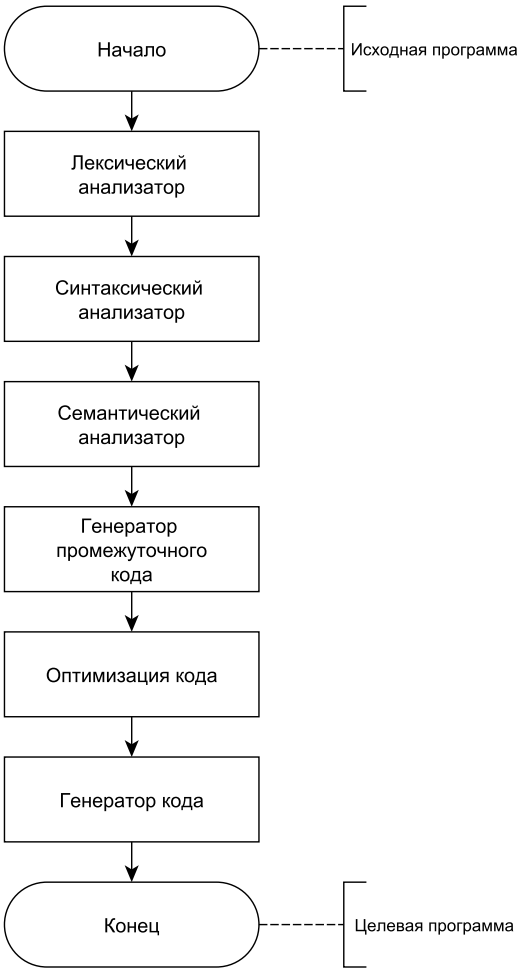

Рисунок 1.1 – Фазы компилятора.

1.2 Лексический анализатор

Цель – превратить поток символов в токены (этот процесс называется «токенизацией»). Выполняется группировка определённых терминальных символов в лексемы. Задаются конкретные правила в виде регулярных выражений, детерминированных конечных автоматов, грамматик.

Лексический анализ может представляться как один из этапов синтаксического анализа.

Обнаружение лексических ошибок, таких как недопустимые символы, ошибки идентификаторов или числовых констант, также является частью этого процесса. Кроме того, на этом этапе происходит удаление комментариев и обработка директив условной компиляции.

1.3 Синтаксический анализатор

Иерархический анализ называется разбором («parsing») или синтаксическим анализом, который включает группировку токенов исходной программы в грамматические фразы, используемые компилятором. Обычно они представляются в виде дерева.

Обычно это представление выражается в виде абстрактного синтаксического дерева, где каждый внутренний узел является оператором, а дочерние – его аргументами. Среди них можно выделить несколько групп связанных объектов:

- элементы арифметических выражений: каждый узел представляет собой операцию и содержит её аргументы;
- элементы системы типов: базовые типы (числовые, строковые, структуры и т.п.), указатели, массивы и функции;
- выражения пяти типов: арифметические, блочные и управляющие выражения, условные конструкции, циклы.

Разбор начинается со стартового нетерминала.

Условные конструкции описывают конструкцию «if», включающую в себя арифметическое выражение условия, выражение, выполняемое в случае его истинности, и альтернативное опциональное выражение.

Конструкции циклов (включают в себя «while», «do while», «for») описывают арифметическое выражение условия и выражение, исполняемое в цикле.

Управляющие выражения – «break», «continue», «return» и т.д.

Блочные выражения – последовательность других выражений, преимущественно используются в качестве тел функций и условных выражений и циклов.

Полученная грамматическая структура используется в последующих этапах компиляции для анализа исходной программы и генерации кода для целевой платформы.

Синтаксический анализ выявляет синтаксические ошибки, относящиеся к нарушению структуры программы.

1.4 Семантический анализатор

В процессе семантического анализа проверяется наличие семантических ошибок в исходной программе и накапливается информация о типах для следующей стадии – генерации кода. Используются иерархические структуры, полученные во время синтаксического анализа для идентификации операторов и операндов выражений и инструкций.

Как правило, семантический анализатор разделяется на ряд более мелких, каждый из которых предназначен для конкретной конструкции. Соответствующий семантический анализатор вызывается синтаксическим анализатором как только он распознает синтаксическую единицу, требующую обработки.

Семантические анализаторы взаимодействуют между собой посредством информации, хранящейся в структурах данных, например, в центральной таблице символов.

1.5 Генерация кода

Последний этап – генерация кода. Начинается тогда, когда во все системные таблицы занесена необходимая информация. В этом случае, компилятор переходит к построению соответствующей программы в машинном коде. Код генерируется при обходе дерева разбора, построенного на предыдущих этапах.

Для получения машинного кода требуется два отдельных прохода:

- генерация промежуточного кода;
- генерация собственно машинного кода.

Выводы

Таким образом, в данной работе

2 Конструкторская часть

2.1 Формат входных данных

В качестве входных данных выступает пользовательское описание какого-либо объекта. Оно должно соответствовать следующим требованиям:

1) определение даётся полностью на русском языке;
2) недопустимо использование аббревиатур, сокращений и т.д., все слова должны быть употреблены в полной форме;
3) описание должно укладываться в 1-2 предложения;
4) следует связно излагать свои мысли;
5) допускается голосовой ввод, результат которого в дальнейшем будет преобразован в текстовый формат.

2.2 Формат выходных данных

Выходные данные представляются как множество терминов из выборки, каждому из которых поставлены в соответствие величина косинусного сходства и её промасштабированное значение, выраженное в процентах.

Если в ходе работы метода дополнительно привлекалась сеть синтаксических графов, то пользователю также предоставляется информация о количестве совпавших слов (в запросе пользователя и терминах сети) и процентное соотношение.

Для интерпретации полученных значений косинусного сходства и количества совпавших слов используется функция softmax [31] позволяющая перевести множество полученных значений в вектор процентов уверенности в том, что был описан соответствующий термин.

2.3 IDEF0

Разрабатываемый метод состоит из нескольких этапов, которые представлены на рисунках **??-??**. Необходимо по изложению пользователя определить

описываемый объект.

Эта задача решается в несколько этапов: обработка запроса клиента, его преобразование, путём извлечения ключевых слов, сопоставление полученных данных с уже имеющимися онтологиями (сформированными на базе статистики и синтаксических графов), формирование промежуточного результата, и затем – принятие решения о том, какой именно термин (один или несколько) наиболее подходит под это описание.

В методе используется два подхода: статистический и на основе семантической сети, для каждого из них формируется онтология, процесс создания которой состоит из нескольких последовательных шагов.

Так формирование онтологии, в основе которой лежит статистические данные, представлено на рисунках **??-??**.

2.3.1 Алгоритм предобработки данных

На рисунке **??** представлена схема алгоритма обработки данных.

В процессе нормализации всё переводится в нижний регистр, удаляются все символы, кроме, букв русского языка, между словами выставляется фиксированно только один пробел.

2.3.2 Алгоритм создания онтологии на основе синтаксических графов

Из-за того, что в большинстве инструментов для определения словосочетаний заложены принципы, которые определил Теньер, для решения поставленной задачи необходимо идентифицировать служебные части речи.

В случае обнаружения и наличия зависимых от них слов, менять у детей этого элемента идентификатор родителя на идентификатор родителя для обрабатываемого слова. Таким образом, в синтаксическом дереве будут только слова, которые так или иначе несут смысловую нагрузку.

2.3.3 Алгоритм построения сети

Сеть строится из синтаксических графов. При их слиянии необходимо следить за тем, чтобы не было дублирующих узлов (то есть, узлов, ассоцииро-

ванных с одним и тем же словом).

При обнаружении повторяющихся единиц, необходимо слить полезную информацию в уже существующий в графе узел.

2.3.4 Алгоритм поиска косинусного сходства

2.3.5 Алгоритм поиска по сети

В качестве структуры данных используется дек, который позволяет добавлять элементы как в «голову», так и в «хвост».

По умолчанию сеть обходится в ширину (элемент снимается с головы, добавляется в хвост), причём каждое слово, связанное с обрабатываемым узел, проверяется на наличие в пользовательском вводе. Как только находится слово, присутствующее в обеих структурах, то потомки этого узла добавляются в голову дека (инициирование обхода в глубину) и увеличивается счётчик количества совпавших слов.

Так происходит до тех пор, пока дек не опустеет. На рисунке **??** подробно изложен этот алгоритм.

2.4 ER-диаграмма

На рисунках **??-??** представлены ER-диаграммы для двух рассматриваемых онтологий.

Сущность ключевого слова (Word) содержит такие поля, как название (name) и вес (weight). Из таких элементов состоит сущность Term (термин), в нём также хранится само название термина.

NodeTerm – узел графа, который помимо названия объекта, с которым он связан, содержит информацию о его признаках, действиях, свойствах и т.д. Кроме того, хранится информация о терминах, в которых было употреблено данное слово (mentions).

Согласно ранее установленным условиям, помимо того, что один граф может состоять из нескольких узлов, один узел может относиться к нескольким графам. Сущность сети включает как идентификационную информацию, так и информацию о её составляющих.

2.5 Use-case диаграмма

На рисунке **??** продемонстрирована Use-case диаграмма, на которой наглядно показаны возможности каждого из участников. Выделяются две роли: пользователь и администратор.

Для обоих предлагается два способа ввести запрос: через текстовое поле или через голосовой ввод. У обоих есть возможность просмотреть подробные результаты запроса и локальную сеть запроса, если она была использована в методе.

Администратор, в отличие от пользователя, может вносить изменения в обе онтологии, и менять данные как по отдельным терминам, так и по всем сразу.

Дополнительно он может увидеть наглядное изображение всей сети и определения терминов из словаря.

Выводы

В текущем разделе был определён формат входных и выходных данных, предоставлены IDEF0 схемы, подробные схемы основных алгоритмов, use-case диаграмма.

3 Технологическая часть

3.1 Выбор средств программной реализации

ЗАКЛЮЧЕНИЕ

Таким образом, в рамках текущей выпускной квалификационной работы был разработан и реализован метод определения объекта из ограниченной выборки по нечёткому описанию на естественном языке. Объём проделанной работы соответствует требованиям технического задания.

Разработанное приложение позволяет:

- определять объект из ограниченной выборки по нечёткому описанию на естественном языке, запрос может быть изложен как через текстовое поле, так и через голосовой ввод;
- обновлять из интерфейса онтологию, построенную как на статистических данных, так и на графовых структурах;
- наглядно демонстрировать строение синтаксических графов, как в составе сети, так и по отдельности.

В результате проделанной работы были выполнены все поставленные задачи.

- Была проанализирована предметная область, проведён сравнительный анализ существующих методов решения, выявлены основные преимущества и недостатки.
- Также, рассмотрены особенности работы с текстами на естественном языке, обоснована необходимость в предварительной обработке.
- Описан принцип формирования онтологии и основные ограничения для определения ключевых слов.
- В результате проведённого предварительного анализа, определены основные этапы поиска нечётких дубликатов, а также критерий принятия решения об использовании вспомогательного метода.
- Формализованы входные и выходные данные метода.
- Пошагово описана структура реализуемого алгоритма.
- Разработано и протестировано программное обеспечение, демонстрирующее работу данного метода.

- Проведено исследование поведения алгоритма при различных входных данных и онтологиях.

В качестве направлений дальнейшей работы можно выделить следующие:

- дальнейшее увеличение датасета;
- использование параллельных вычислений при нахождении косинусного сходства запроса и составляющих онтологии;
- определение критерия досрочного выхода из процедуры поиска в сети синтаксических графов в целях уменьшения времени обработки запроса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. АХО А.В, ЛАМ М.С., СЕТИ Р., УЛЬМАН Дж.Д. Компиляторы: принципы, технологии и инструменты. – М.: Вильямс, 2008.

2. Data Creation and Replication Will Grow at a Faster Rate than Installed Storage Capacity, According to the IDC Global DataSphere and StorageSphere Forecasts [Электронный ресурс]. – Режим доступа: https://www.idc.com/getdoc.jsp?containerId=prUS47560321 (Дата обращения: 12.11.2021).

3. Data Age 2025: the datasphere and data-readiness from edge to core [Электронный ресурс]. – Режим доступа: https://www.i-scoop.eu/big-data-action-value-context/data-age-2025-datasphere/ (Дата обращения: 12.11.2021).

4. How the pandemic impacted data creation and storage [Электронный ресурс]. – Режим доступа: https://www.i-scoop.eu/big-data-action-value-context/data-storage-creation/ (Дата обращения: 12.11.2021).

5. Еникеев, Р. Д. Двигатели внутреннего сгорания. Основные термины и русско-английские соответствия : учеб. пособие для вузов / Р. Д. Еникеев, Б. П. Рудой – М. : Машиностроение, 2004. – 383 с. – Библиогр.: с. 378-382. – ISBN 5-217-03267-7.

6. Гаврилова, Т.А. Базы знаний интеллектуальных систем [Текст] / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб: Питер, 2000. – 384 с. – ISBN 5-272-00071-4.

7. Autonomy. Autonomy Technology Whitepaper. – 1998. [Электронный ресурс]. – Режим доступа: http://www.autonomy.com (Дата обращения: 26.11.2021).

8. Олейник, Е. HP Autonomy IDOL: анализ совсем неструктурированных данных / Storage News – 2012. – № 3 (51). – С. 28-31 / [Электронный ресурс]. –

Режим доступа: http://www.storagenews.ru/51/HP_Autonomy_51.pdf (Дата обращения: 20.11.2021).

9. Громов, Ю.Ю. Интеллектуальные информационные системы и технологии: учебное пособие [Текст] / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 244 с. – ISBN 978-5-8265-1178-7.

10. Макеева, Л. Б. Язык, онтология и реализм. [Текст] / Л. Б. Макеева; Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011. – 310, [2] с. – 600 экз. – ISBN 978-5-7598-0802-2 (в пер.).

11. Gruber, T. R. A Translation Approach to Portable Ontologies. – Knowledge Acquisition – 1993. – № 5(2). – p. 199–220.

12. Noy, N.F. Ontology Development 101: A Guide to Creating Your First Ontology / N.F. Noy, D.L. McGuinness – Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report. – 2001. – SMI-2001-0880, p. 1-25.

13. Пальчунов, Д.Е. Решение задачи поиска информации на основе онтологий [Текст] / Бизнес-информатика – 2008. – № 1. – С. 3-13.

14. Лингвистический энциклопедический словарь [Текст] / Под ред. В. Н. Ярцевой. – М.: Советская энциклопедия, 1990. – 685 с.

15. Большакова, Е.И Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие [Текст] / Е.И. Большакова, К.В. Воронцов, Н.Э. Ефремова, Э.С. Клышинский, Н.В. Лукашевич, А.С. Сапин – М.: Изд-во НИУ ВШЭ, 2017. – 269 с. – ISBN 978-5-9909752-1-7.

16. Elizabeth, D. Natural Language Processing. – Center for Natural Language Processing. – 2001.

17. Srividhya, V. Evaluating Preprocessing Techniques in Text Categorization / V. Srividhya, R. Anitha – International Journal of Computer Science and Application Issue – 2010 – p. 49-51. – ISSN 0974-0767.

18. Большакова, Е.И. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие [Текст] / Е.И. Большакова, Э.С. Клышинский, Д.В. Ландэ, А.А. Носков, О.В. Пескова, Е.В. Ягунова —- М.: МИЭМ, 2011. —- 272 с. – ISBN 978–5–94506–294–8.

19. Пархоменко, П. А. Обзор и экспериментальное сравнение методов кластеризации текстов [Текст]/ П. А. Пархоменко, А. А. Григорьев, Н. А. Астраханцев – Труды ИСП РАН, 2017 – том 29, выпуск 2 – С. 161–200.

20. Akiko Aizawa An information-theoretic perspective of tf–idf measures – Information Processing and Management. – 2003 – p. 45-65.

21. Ramos, J. Using TF-IDF to Determine Word Relevance in Document Queries [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/228818851_Using_TF_IDF_to_determine_word_relevance_in_document_queries (Дата обращения: 02.12.2021).

22. Зиберт, А.О. Разработка системы определения наличия заимствований в работах студентов высших учебных заведений. Алгоритмы поиска нечетких дубликатов [Текст] / А.О. Зиберт, В.И. Хрусталев – Universum: Технические науки : электрон. научн. журн. – 2014. – № 3 (4).

23. Квашина, Ю.А. Методы поиска дубликатов скомпонованных текстов научной стилистики [Текст] / Ю.А. Квашина. – Технологический аудит. – 2013. – № 3/1(11) – С. 16-20.

24. Зеленков, Ю.Г. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов [Текст] / Ю.Г. Зеленков, И.В. Сегалович – Труды 9-ой Всероссийской научной конференции «Электронные библиотеки:

перспективные методы и технологии, электронные коллекции» RCDL’2007: Сб. работ участников конкурса. – Т. 1. – Переславль Залесский: «Университет города Переславля», 2007. – С. 166—174.

25. Цимбалов, А.В. Метод шинглов [Текст] / А.В. Цимбалов, О.В. Золотарев – Вестник. – 2016. – Серия «Сложные системы: модели, анализ и управление». Выпуск 4. – С. 72-79.

26. Преображенский, Ю.П. О методах создания рекомендательных систем [Текст] / Ю.П. Преображенский, В. М. Коновалов – Вестник Воронежского института высоких технологий. – 2019. – № 4(31). – С.75-79.

27. Бабкин, Э.А. Принципы и алгоритмы искусственного интеллекта: Монография / Э.А. Бабкин, О.Р. Козырев, И.В. Куркина. – Н. Новгород: Нижегород. гос. техн. ун-т. 2006. 132 с.

28. Теньер Л. Основы структурного синтаксиса: Пер. с фр. – Прогресс, 1988.

29. Национальный корпус русского языка. 2003—2022 [Электронный ресурс]. – Режим доступа: ruscorpora.ru (Дата обращения: 04.04.2022).

30. Национальный корпус русского языка. Синтаксический корпус. 2003—2022 [Электронный ресурс]. – Режим доступа: https://ruscorpora.ru/new/search-syntax.html (Дата обращения: 04.04.2022).

31. Маршаков Д. В. СРАВНЕНИЕ РЕЗУЛЬТАТОВ НЕЙРОСЕТЕВОЙ КЛАССИФИКАЦИИ С ПРИМЕНЕНИЕМ SOFTMAX И ФУНКЦИИ РАССТОЯНИЯ //Математические методы в технологиях и технике. – 2021. – №. 8. – С. 75-78.

32. Документация по Python 3 [Электронный ресурс]. Режим доступа: https://docs.python.org/3/ (Дата обращения 01.02.2022)

33. Документация по PyCharm [Электронный ресурс]. Режим доступа: https://www.jetbrains.com/pycharm/guide/tips/quick-docs/ (Дата обращения 01.02.2022)

34. Документация по Discord [Электронный ресурс]. Режим доступа: https://support.discord.com/hc/ru (Дата обращения 05.02.2022)

35. Аналитика трафика и доля рынка Discord [Электронный ресурс]. Режим доступа: https://www.similarweb.com/ru/website/discord.com/ (Дата обращения 20.05.2022)

36. Документация по Javascript [Электронный ресурс]. Режим доступа: https://javascript.ru/manual (Дата обращения 02.02.2022)

ПРИЛОЖЕНИЕ А

Таблица 1 – Используемые термины

№	Название термина	№	Название термина
1	агрегат наддува	22	конденсация
2	адиабатный процесс	23	контроллер
3	аккумуляторный элемент	24	коэффициент полноты сгорания топлива
4	блокировка пуска	25	к-т усиления регулятора
5	вибропрочность	26	лубрикатор
6	воздухозаборник	27	маховик
7	впускное отверстие	28	механическая мощность компрессора
8	выпускной коллектор	29	необратимый цикл
9	гистерезис регулятора	30	обменник давления
10	глушитель шума	31	обратимый цикл
11	двигатель внутреннего сгорания	32	охладитель
12	двухтактный двигатель	33	помпаж компрессора
13	дефлектор	34	привод распределительного вала
14	длинноходный двигатель	35	пусковая жидкость
15	зубчатый ремень	36	рабочая камера
16	изобарный процесс	37	рабочее тело
17	изотермический процесс	38	рабочий ход
18	изохорный процесс	39	распределительный вал
19	карбюратор	40	расширение
20	коленчатый вал	41	свеча зажигания
21	компрессор	42	сжатие

Продолжение на следующей странице

№	Название термина	№	Название термина
43	термодинамический процесс	47	тепловой двигатель
44	термодинамический цикл	48	ход поршня
45	топливный фильтр	49	цикл Карно
46	форсунка	50	электрическое напряжение