

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з
“Лінгвістичне забезпечення САПР”

1 МЕТА ТА ЗАВДАННЯ

Метою курсової роботи є отримання студентами практичних навичок у розробці транслятора як закінченого програмного продукту.

Курсова робота спирається на проведений цикл лабораторних робіт та становить собою транслятор для заданої мови.

Студент повинен розробити граматику заданої мови, визначити структуру свого транслятора, описати та реалізувати основні функціональні блоки транслятора, розробити інтерфейс користувача, провести тестування свого програмного продукту.

2 ЗАВДАННЯ НА КУРСОВУ РОБОТУ

Розробити та реалізувати транслятор з заданої мови, що включає наступні блоки:

Лексичний аналізатор.

Синтаксичний аналізатор.

Побудова проміжної форми подання програми – польського інверсного запису (ПОЛІЗ).

Виконання ПОЛІЗ.

Мови, що використовуються: Мова програмування повинна містити в собі оператор присвоювання, оператори вводу та виводу, структура яких пропонується самим студентом, а також оператори умовного переходу та циклу, що надаються згідно з варіантом.

3 СКЛАД, ОБСЯГ І СТРУКТУРА КУРСОВОЇ РОБОТИ

Курсова робота складається з програми, що реалізує транслятор з заданої мови, та пояснювальної записки.

Записка повинна містити в собі основну частину та додатки.

Основна частина складається з наступних розділів:

- 1) анотація:
- 2) вступ:
- 3) індивідуальне завдання:
- 4) граматики мови (у відповідності до індивідуального завдання);
- 5) структура транслятора (послідовність роботи блоків, число проходів та ін.);
- 6) лексичний аналізатор (дати опис алгоритму з поданням переліку лексем, та всіх допоміжних даних залежно від алгоритму реалізації, а саме: перелік роздільників для алгоритму 1; діаграма станів та таблиця класів символів для алгоритму 2; діаграма станів та список переходів кінцевого автомату для алгоритму 3; навести перелік помилок, які виявляє лексичний аналізатор; навести приклад роботи лексичного аналізатора при наявності помилок у тестовій програмі та при їх відсутності);
- 7) синтаксичний аналізатор (дати опис алгоритму, надати потрібні перетворення граматики залежно від алгоритму, а також додаткову інформацію, а саме: дерево процедур, що використовуються, для алгоритму 1; діаграми станів та списки переходів всіх підавтоматів для алгоритму 2; таблицю передувальних та візуалізацію процесу розбору для алгоритму 3; навести перелік помилок, які виявляє синтаксичний аналізатор; навести приклад роботи

синтаксичного аналізатора при наявності помилок у тестовій програмі та при їх відсутності);

8) проміжна форма подання програми - ПОЛІЗ (навести блок-схеми реалізації складних операторів, особливості трансляції службових слів та роботи стека, особливості реалізації операторів вводу-виводу. Навести приклад побудови ПОЛІЗ та пояснити отриманий результат);

9) виконання отриманої ПОЛІЗ-програми (навести особливості виконання команд ПОЛІЗ, таблицю позначок, результат виконання тестової програми; для контролю обчислити цю ж програму на будь-якій алгоритмічній мові та порівняти результати;

10) інструкція користувача по роботі з транслятором;

11) висновок;

12) список використаних джерел.

Записка повинна включати наступні додатки:

Додаток 1. Текст програми.

Додаток 2. Опис програми.

4 ВКАЗІВКИ ДО ВИКОНАННЯ РОЗДІЛІВ КУРСОВОЇ РОБОТИ

4.1 Основні блоки і структура транслятора

До складу будь-якого транслятора входять три основні компоненти:

- а) лексичний аналізатор,
- б) синтаксичний аналізатор,
- в) генератор кодів машинних команд.

На фазі лексичного аналізу (ЛА) вхідна програма, яка являє собою потік символів, розбивається на лексеми - слова у відповідності з визначеннями мови. Основним формалізмом, що лежить в основі реалізації лексичних аналізаторів, є кінцеві автомати і регулярні вирази. Лексичний аналізатор може працювати в двох основних режимах: або як підпрограма, що викликається синтаксичним аналізатором за черговою лексемою, або як повний прохід, результатом якого є файл лексем.

В процесі виділення лексем ЛА може як самостійно будувати таблиці імен і констант, так і видавати значення для кожної лексеми при черговому зверненні до нього. В цьому випадку таблиця імен будується в наступних фазах (наприклад, в процесі синтаксичного аналізу).

На етапі ЛА виявляються деякі (найпростіші) помилки аналізу (неприпустимі символи, невірний запис чисел, ідентифікаторів, необ'явлені ідентифікатори, відсутні мітки, на які є посилання, та ін.).

Основна задача синтаксичного аналізу - розбір структури програми. Як правило, під структурою розуміється дерево, відповідне розбору в контекстно-вільній граматиці мови. Результатом синтаксичного аналізу є синтаксичне дерево зі засланнями на таблицю імен. В процесі синтаксичного аналізу також виявляються помилки, пов'язані зі структурою програми.

Після цього програма може бути переведена у внутрішнє подання. Це робиться з метою оптимізації і/або зручності генерації коду.. Як внутрішнє подання може використовуватися префіксний або постфіксний запис, орієнтований граф, трійки, четвірки та інші.

Нарешті, генерація коду - остання фаза трансляції. Результатом її є або асемблерний модуль, або об'єктний (або завантажувальний) модуль. Для генерації коду розроблені різноманітні засоби, такі як таблиці рішень, зіставлення зразків, включаючи динамічне програмування, різноманітні синтаксичні засоби.

Звичайно, ті або інші фази транслятора можуть або бути відсутні зовсім, або об'єднуватися.

При розробці курсової роботи треба реалізувати наступні фази трансляції:

- лексичний аналіз;
- синтаксичний аналіз;
- побудова внутрішнього подання (постфіксний запис);
- виконання (інтерпретація) внутрішнього подання.

5 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ РОЗРАХУНКОВО- ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

При оформленні курсової роботи слід користуватися Методичними вказівками до оформлення курсових робіт (Методкабінет кафедри АПЕПС)

ВКАЗІВКИ ДО ПОРЯДКУ ЗАХИСТУ РОБОТИ

Захист курсової роботи включає наступні етапи:

- демонстрація на комп'ютері роботи розробленого транслятора при введенні на вхід програм на заданій мові;
- проходження нормоконтролю на відповідність правилам оформлення пояснювальної записки та додатків;
- пояснення та захист запропонованих рішень.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Медведєва, В. М. Транслятори: лексичний та синтаксичний аналізатори / В. М. Медведєва, В. А. Третяк. – К. : НТУУ "КПІ", 2012. – 148 с.
2. Медведєва, В. М. Транслятори: внутрішнє подання програм та інтерпретація / В. М. Медведєва, В. А. Третяк. – К. : НТУУ "КПІ", 2015. – 144 с.
3. Ахо Альфред, Лам Моника, Сети Рави, Ульман Джефри Д. Компиляторы: принципы, технологии и инструментарий, 2-е изд. : Пер. с англ. – М. : ООО „И.Д. Вильямс”, 2008. – 1184 с.
4. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов. : М. : Мир 1988. – 192 с.
5. Хантер Р. Проектирование и конструирование компиляторов. – Пер. с англ. - М. : Финансы и статистика 1984, 217 с.

Додаткова література

1. Рейерд-Смит В. Дж. Теория формальных языков. Вводный курс. Пер. с англ.-М.: Радио и связь, 1988.
2. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. Пер с англ. -М.: Мир, 1975.
3. Ахо А. Ульман Д. Теория синтаксического анализа, перевода и компиляции. -М.: Мир, 1978.