

**Факультет компьютерных наук
Кафедра вычислительных систем**

**КОНТРОЛЬНЫЕ ЗАДАНИЯ
по дисциплине «Системное программное обеспечение»**

Тема «LR(0)-распознаватели»

Цель работы: Закрепление навыков и знаний по организации LR(0)-распознавателей.

Задание: Построить граф взаимосвязи ситуаций, управляющую таблицу для языка, заданного LR(0)-грамматикой.

Указания к выполнению задания:

1. Для заданного варианта задания, представленного в виде правил грамматики, описать все четыре компонента грамматики $G(V_t, V_n, P, S)$, где V_t – множество терминальных символов грамматики, V_n – множество нетерминальных символов, P – множество правил, S – целевой символ грамматики.
2. Преобразовать грамматику к пополненному виду, получив грамматику G' , содержащую дополнительное правило $S' \rightarrow S$, если хоть одно правило исходной грамматики содержит в правой части целевой символ грамматики S .
3. Построить последовательность возможных ситуаций R_0 (начальная), R_1, \dots, R_m для полученной грамматики. В данной последовательности не должно быть ситуаций, полностью совпадающих по виду входящих в них правил (лишние ситуации исключить).
4. Начертить граф взаимосвязи ситуаций для построенной последовательности ситуаций.
5. Построенную грамматику и граф взаимосвязи ситуаций включить в отчет по работе. Пронумеровать правила грамматики.
6. По графу взаимосвязи ситуаций построить управляющую таблицу синтаксического разбора. Предусмотреть в ней колонки: *Номер состояния, Содержимое стека, Действия, Переходы (по числу символов грамматики)*.
7. Выбрать цепочку символов, относящуюся к заданной грамматике, и построить для нее таблицу разбора данной цепочки. Сложность цепочки должна быть таковой, чтобы таблица разбора содержала не менее 10-15 строк, а для вывода цепочки использовалось каждое правило грамматики не менее одного раза. Таблица должна содержать столбцы: *Номер n/n, Стек, Входная цепочка, Действие, Номер правила свертки, Вид правила*.
8. Управляющую таблицу синтаксического разбора, вид выбранной цепочки и таблицу разбора данной цепочки включить в отчет. В конце отчета на основе крайних правых столбцов таблицы разбора записать последовательность сентенциальных форм грамматики, позволивших свернуть исходную цепочку в целевой символ грамматики (обратное порождение цепочки). Отчет предоставляется на бумажном носителе информации.

Варианты заданий (20 вариантов):

1. $S \rightarrow a S b$ $S \rightarrow a S c$ $S \rightarrow a b$
2. $S \rightarrow F$ $F \rightarrow a$ $F \rightarrow (F + F)$
3. $S \rightarrow A$ $A \rightarrow a A A$ $A \rightarrow b$
4. $S \rightarrow a S b$ $S \rightarrow a S S$ $S \rightarrow c$
5. $S \rightarrow A$ $A \rightarrow (A + A)$ $A \rightarrow a$
6. $S \rightarrow a S S b$ $S \rightarrow a S S S$ $S \rightarrow c$
7. $S \rightarrow F F \rightarrow T$ $T \rightarrow a$ $T \rightarrow (F)$
8. $S \rightarrow x S y$ $S \rightarrow z$ $S \rightarrow x S S$
9. $S \rightarrow a$ $S \rightarrow (S R$ $R \rightarrow , S R$ $R \rightarrow)$
10. $S \rightarrow 0$ $S \rightarrow 2S 1$ $S \rightarrow 3S S 2$ $S \rightarrow 4S S 3$ $S \rightarrow 1S S S 4$
11. $S \rightarrow c A$ $S \rightarrow c c B$ $A \rightarrow c A$ $A \rightarrow a$ $B \rightarrow c c B$ $B \rightarrow b$
12. $S \rightarrow D D \rightarrow b$ $D \rightarrow a D D$

13. $S \rightarrow x S y \quad S \rightarrow z \quad S \rightarrow x S S$
14. $S \rightarrow E \quad E \rightarrow (E - E) \quad E \rightarrow a$
15. $S \rightarrow x S S y \quad S \rightarrow x S S S \quad S \rightarrow z$
16. $S \rightarrow L \quad L \rightarrow T \quad T \rightarrow b \quad T \rightarrow [L]$
17. $S \rightarrow 0 \quad S \rightarrow 2 S \quad S \rightarrow 4 S S \quad S \rightarrow 6 S S S$
18. $S \rightarrow k \quad S \rightarrow [SR \quad R \rightarrow , SR \quad R \rightarrow]$
19. $S \rightarrow 1 \quad S \rightarrow S 2 \quad S \rightarrow S S 3 \quad S \rightarrow S S 4 \quad S \rightarrow S S S 5$
20. $S \rightarrow c X \quad S \rightarrow c c Y \quad X \rightarrow a \quad X \rightarrow c X \quad Y \rightarrow b \quad Y \rightarrow c c Y$

**Пример построения распознавателя для языка
на основе LR(0)-грамматики**

Задание

Для грамматики $G(\{a,b\}, \{S\}, \{S \textcircled{R} aSS \mid b\}, S)$ выполнить следующее:

1. Преобразовать ее к пополненному виду, получив грамматику G' , содержащую дополнительное правило $S' \textcircled{R} S$, так как правила исходной грамматики содержат в правой части целевой символ грамматики S .
2. Построить последовательность возможных ситуаций R_0 (начальная), R_1, \dots, R_m для полученной грамматики. В данной последовательности не должно быть ситуаций, полностью совпадающих по виду входящих в них правил (лишние ситуации исключить).
3. Начертить граф взаимосвязи ситуаций для построенной последовательности ситуаций.
4. По графу взаимосвязи ситуаций построить управляющую таблицу синтаксического разбора. Предусмотреть в ней колонки: *Номер состояния, Содержимое стека, Действия, Переходы (по числу символов грамматики)*.
5. Выбрать цепочку символов, относящуюся к заданной грамматике, и построить для нее таблицу разбора данной цепочки. Сложность цепочки должна быть таковой, чтобы таблица разбора содержала не менее 10-15 строк. Таблица должна содержать столбцы: *Номер n/n, Стек, Входная цепочка, Действие, Номер правила свертки, Вид правила*.

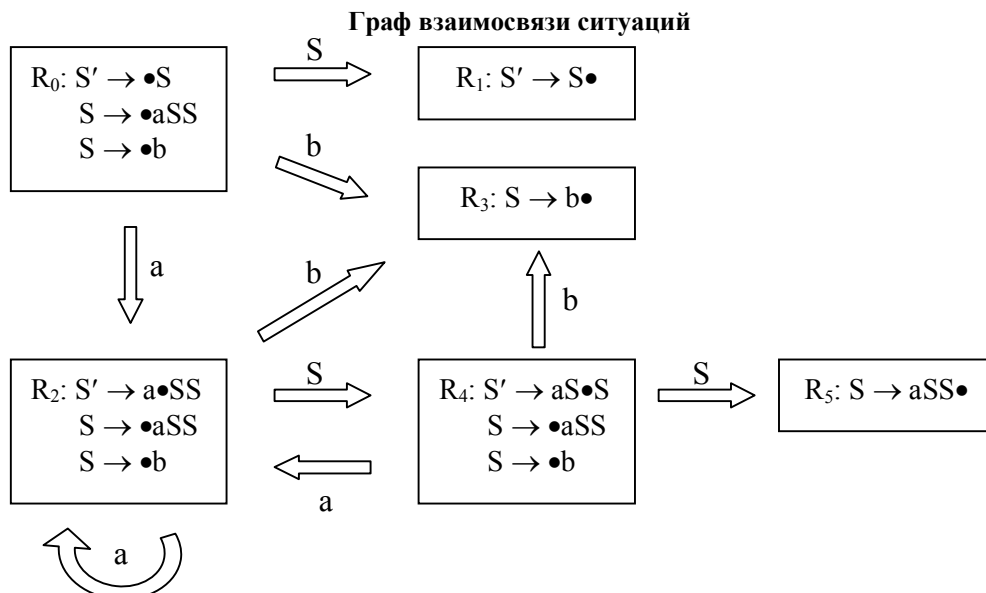
Грамматика:

$G(V_t, V_n, P, S)$,

где $V_t = \{a,b\}$ – множество терминальных символов,
 $V_n = \{S', S\}$ – множество нетерминальных символов,
 P – множество правил грамматики,

- 1) $S' \textcircled{R} S$
- 2) $S \textcircled{R} aSS$
- 3) $S \textcircled{R} b$

S – целевой символ грамматики



Управляющая таблица синтаксического разбора

№ состояния	Стек	Действия (правило свертки)	Переходы		
			S	a	b
0	⊥	Перенос	1	2	3
1	S	Успех, 1			
2	a	Перенос	4	2	3
3	b	Свертка, 3			
4	aS	Перенос	5	2	3
5	aSS	Свертка, 2			

Пример цепочки: abababb

Таблица разбора цепочки

№ п/п	Стек	Входная цепочка	Действие	№ правила	Вид правила
1	⊥0	abababb⊥	Перенос		
2	⊥0a2	bababb⊥	Перенос		
3	⊥0a2b3	ababb⊥	Свертка	3	S ® b
4	⊥0a2S4	ababb⊥	Перенос		
5	⊥0a2S4a2	babb⊥	Перенос		
6	⊥0a2S4a2b3	abb⊥	Свертка	3	S ® b
7	⊥0a2S4a2S4	abb⊥	Перенос		
8	⊥0a2S4a2S4a2	bb⊥	Перенос		
9	⊥0a2S4a2S4a2b3	b⊥	Свертка	3	S ® b
10	⊥0a2S4a2S4a2S4	b⊥	Перенос		
11	⊥0a2S4a2S4a2S4b3	⊥	Свертка	3	S ® b
12	⊥0a2S4a2S4a2S4S5	⊥	Свертка	2	S ® aSS
13	⊥0a2S4a2S4S5	⊥	Свертка	2	S ® aSS
14	⊥0a2S4S5	⊥	Свертка	2	S ® aSS
15	⊥0S1	⊥	Свертка	1	S' ® S
16	⊥0S'	⊥			

Вопросы к экзамену

1. Состав системного программного обеспечения. Операционные системы, системы программирования, утилиты. Состав системы программирования. Транслятор - основная часть системы программирования.
2. Возникновение систем программирования (СП). Этапы развития СП. Этапы обработки программ с помощью СП.
3. Понятие графического интерфейса пользователя. Ресурсы пользовательского интерфейса.
4. Структура современной системы программирования и этапы развития СП. Тенденция развития СП.
5. Принципы работы средств СП. Текстовые редакторы. Подсказки, гиперссылки, система справок.
6. Компоновщик и его функции. «Неразрешенные» ссылки. Возможные ошибки, находимые компоновщиком.
7. Загрузчики. Настраиваемый загрузчик. Динамический загрузчик.
8. Отладчики. Их возможности.
9. Библиотеки подпрограмм. Их основные компоненты. Статические библиотеки и их недостатки. Динамические библиотеки и варианты их загрузки и выгрузки. Преимущества и недостатки динамических библиотек.
10. Мобильность программного обеспечения. Основные факторы, влияющие на мобильность. Правила для обеспечения мобильности кода. Стандарт переносимости ПО. Достоинства и недостатки мобильности ПО.
11. Формальные языки и грамматики. Цепочки. Атрибуты цепочек. Операции над ними. Конкатенация, подстановка, обращение, итерация. Пустая цепочка. Звездочка Клини. Символ непустой итерации.

12. Понятие языка. Алфавит. Предложение языка. Лексемы. Способы определения языков. Синтаксис, семантика, лексика языка.
13. Грамматики и распознаватели. Понятие грамматики и ее продукции. Четверка, определяющая грамматику. Целевой символ грамматики. Терминальные и нетерминальные символы. Форма Бэкуса-Наура. Пример представления языка десятичных чисел со знаком. Рекурсия в правилах грамматики, Прямая и косвенная рекурсия. Использование метасимволов в правилах грамматики. Запись правил грамматики в виде направленного графа.
14. Понятие распознавателя. Операции, реализуемые распознавателями. Конфигурация распознавателя. Начальная и конечная конфигурация. Классификация распознавателей. Задача разбора.
15. Включающая классификация языков и грамматик (по Хомскому) и области их использования. Контекстно-зависимые и неукорачивающие грамматики. Контекстно-свободные грамматики. Регулярные грамматики.
16. Классификация языков.
17. Цепочки вывода. Непосредственно выводимые цепочки. Законченный вывод. Сентенциальная форма грамматики. Конечная сентенциальная форма. Синтаксическое дерево, дерево разбора.
18. Неоднозначные грамматики. Левосторонний и правосторонний вывод.
19. Трансляторы. Понятие транслятора, компилятора и интерпретатора. Кросс-система.
20. Этапы трансляции. Фазы трансляции. Анализ и синтез. Предварительная и заключительная стадии компиляции. Понятие оптимизации.
21. Однопроходные и многопроходные компиляторы. Особенности интерпретаторов.
22. Трансляторы с языка ассемблера. Макроязыки, макрокоманды и макроподстановки. Макропроцессор, макрогенератор.
23. Таблицы идентификаторов и их характеристики. Организация ТИ. Неупорядоченный массив. Упорядоченная ТИ. Бинарный поиск. Алгоритм бинарного поиска по дереву. Алгоритм построения дерева. Глубина дерева. Уравновешенные деревья. Сбалансированные деревья. Оптимальные деревья. Применение частот использования ключей.
24. Хеш-функции и хеш-адресация. Преимущества и недостатки их использования. Понятие коллизии. Понятие перехеширования (рехеширования). Степень заполненности таблицы идентификаторов.
25. Метод цепочек для разрешения коллизий. Использование списка с указателями и списка внутри массива.
26. Метод линейной открытой адресации для разрешения коллизий.
27. Открытая адресация с двойным хешированием. Улучшение Брента.
28. Лексический анализ. Понятие лексики, лексемы. Таблицы идентификаторов.
29. Способы организации взаимосвязи лексического и синтаксического анализаторов.
30. Способы построения лексических анализаторов.
31. Пример построения программы лексического анализатора вещественных чисел на основе регулярного выражения и на основе диаграммы переходов. Таблица переходов.
32. Лексические ошибки и способы восстановления после них. Стратегия единственного преобразования, стратегия минимального числа преобразований.
33. Синтаксический анализ. Задача синтаксического анализа. Понятие о нисходящем и восходящем анализе. Пример разбора цепочки при нисходящем анализе.
34. LL(k)-грамматики, LL(1)-грамматики. Стартовый символ нетерминала. Множество FIRST(A).
35. S-грамматики. q-грамматики.
36. Символ-последователь. Множество первых порождаемых символов нетерминала FOLLOW(A). Требования к нему у LL(1)-грамматик.
37. Синтаксический анализ методом рекурсивного спуска с использованием рекурсии и с использованием итерации. Пример.
38. Понятие о факторизации грамматик.
39. Достоинства и недостатки LL(1)-грамматик.
40. Восходящий синтаксический анализ. Пример разбора цепочки.
41. LR(k)-анализ. Модель LR-анализатора.
42. LR(0)-грамматики. Понятие ситуации и последовательности ситуаций. Пункт (элемент грамматики). Правила построения последовательности ситуаций.
43. Граф взаимосвязи ситуаций.
44. Управляющая таблица. Ее структура и правила построения. Таблица разбора.
45. Пример построения распознавателя для LR(0)-грамматики: последовательности ситуаций, графа их взаимосвязи, управляющей таблицы и таблицы разбора цепочки. Вид порождения.
46. Ограничения LR(0)-грамматик.
47. Синтаксический разбор для LR(1)-грамматик. Начальная ситуация. Правила построения последовательности ситуаций, графа их взаимосвязи. Правила заполнения управляющей таблицы. Разбор цепочки. Пример.
48. Особенности построения LR(1)-распознавателей.
49. Семантический анализ. Его фазы.

50. Возможности компиляторов по проверке смысловых норм языков программирования. Рекомендации по формированию полного имени лексемы.
51. Генерация кода. Генерация машинно-независимого кода. Метод СУ-(синтаксически-управляемого) перевода.
52. Формы внутреннего представления программ.
53. Синтаксические деревья. Преобразование дерева разбора в дерево операций. Тетрады. Триады. Обратная польская запись.
54. Преобразование типовых узлов дерева операций в последовательность триад. Пример.
55. Оптимизация кода. Основные критерии оптимизации. Два вида оптимизирующих преобразований.
56. Методы оптимизации. Оптимизация линейных участков программы. Виды оптимизирующих преобразований линейных участков.
57. Удаление бесполезных присваиваний. Устранение избыточных вычислений. Свертка объектного кода и перестановка операций. Арифметические преобразования.
58. Оптимизация вычисления логических выражений. Предопределенные операции и операнды. Инвариантные операнды.
59. Оптимизация передачи параметров в процедуры и функции. Использование регистровой памяти. Подстановка кода.
60. Оптимизация циклов. Вынесение инвариантных вычислений. Замена операций с индуктивными переменными. Слияние и развертывание циклов.
61. Генерация машинно-зависимого кода. СУ-перевод дерева операций на язык ассемблера.
62. Распределение памяти. Классификация областей памяти. Распределение памяти для переменных скалярных типов, для сложных структур. Выравнивание границ областей памяти.
63. Глобальная и локальная память. Статическая и динамическая память. Статическое и динамическое связывание. Менеджеры памяти. Преимущества и недостатки их использования.

ЛИТЕРАТУРА

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. - СПб.: Питер, 2010. - 396с.
2. Ефимов С.С. Лабораторный практикум по курсу «Системное программное обеспечение». – Омск: 2006. – 80 с.
3. Кнуг Д. Э. Искусство программирования. Т.1, Т.3.: Уч. пособие. – М.: Изд. дом «Вильямс», 2000.
4. Ахо А.В., Сети Р., Ульман Дж. Д. Теория синтаксического анализа, перевода и компиляции. - М.: 1978.