

Программа экзамена “Лингвистические основы информатики”,

7 семестр, КН, 2025-2026 уч.г.

Лектор: к.ф.-м.н. Нагребцкая Ю.В.

Преподаватели, ведущие практики: к.ф.-м.н. Нагребцкая Ю.В., к.ф.-м.н. Щербакова В.А.

Теоретическая часть (для всех утверждений предполагается доказательство)

1. Порождающая грамматика, вывод. Язык, порожденный грамматикой.
2. КС-грамматика. Дерево вывода, связь с выводом. Однозначность грамматики.
3. КС-языки и иерархия Хомского.
4. Приведенные КС-грамматики. Алгоритм получения приведенной грамматики.
5. Ациклические грамматики. Алгоритм удаления циклов (без доказательства). Лемма 4 об удалении циклов (с доказательством).
6. Определение аннулирующих нетерминалов. Алгоритм получения множества аннулирующих нетерминалов. Доказательство обоснования.
7. Определение ε -свободной грамматики. Алгоритм приведения грамматики к ε -свободной грамматике. Доказательство обоснования.
8. Грамматика в нормальной форме Хомского (НФХ). Формулировка теоремы о грамматике в НФХ и формулировки леммы 7. Доказательство леммы 7.
9. Алгоритм Кока-Янгера-Касами (СΥК-алгоритм). Доказательство обоснования.
10. Определение подстановки. Теорема о подстановке КС языков. Доказательство необходимости.
11. Теорема о накачке для КС языков (формулировка). Доказательство второго этапа, где уже доказано, что от корня до листа в дереве вывода с минимальной высотой данной цепочки есть путь, помеченный двумя одинаковыми нетерминалами.
12. Периодические множества. Теорема об унарных языках. Доказательство того, что периодическое множество является рациональным языком.
13. Определение подстановки. Теорема о подстановке КС языков. Доказательство необходимости.
14. Теорема о пересечении КС-языка и рационального языка (формулировка). Доказательство первого этапа: пересечение КС-языка и рационального языка включается в язык, порождаемый построенной КС-грамматикой.
15. Лексический анализатор. Основные понятия.
16. Определения автомата с магазинной памятью (МПА), конфигурации МПА, вывода. Язык, распознаваемый МПА. Язык, распознаваемый МПА при пустом стеке. Теорема об эквивалентности распознавания МПА и МПА при пустом стеке.
17. Левая факторизация. Алгоритм левой факторизации.
18. Непосредственная левая рекурсия. Алгоритм устранения непосредственной левой рекурсии.
19. Левая рекурсия. Алгоритм устранения левой рекурсии. Доказательство обоснования (только 1 утверждение).
20. Множества FIRST, FOLLOW, определения и свойства (без док-ва). Алгоритмы нахождения множеств FIRST и FOLLOW (без док-ва обоснов. корректности).
21. Множество SELECT, определения и свойства (без док-ва). Алгоритмы нахождения множества SELECT (без док-ва обоснов. корректности).
22. Разделенная (или LL(0)) грамматика. LL(1) – грамматика. Определения и свойства (без док-ва).

23. LL(k)-грамматики, LL-грамматики. Основные понятия и свойства. Утверждения о том, что $LL(k) \subset LL(k+1)$ и что класс LL не совпадает с множеством всех КС грамматик с доказательством.
24. Алгоритм построения нисходящего анализатора для LL(1) – грамматик, обоснование корректности.
25. Метод рекурсивного спуска для LL(1) – грамматик.
26. Определение отношений предшествования. Теорема об отношениях предшествования. Доказательство 1 и 2 утверждений.
27. LR(0)-автомат. LR(0)-грамматика. Анализатор на основе LR(0)-автомата.
28. Алгоритм построения SLR(1)-анализатора. Понятие SLR(1)-грамматики.
29. LR(1)-автомат, LR(1)-анализатор, LR(1)-грамматика.
30. «Гибридный» метод построения LR-автомата. LALR(1)-грамматика.
31. LR(k)-автомат, LR(k)-грамматика, LR(k)-анализатор, LR(k)-язык, LR-язык. Утверждение о том, что $LL(k) \subset LL(k+1)$ с доказательством.
32. Основная теорема LR анализа с доказательством. Лемма 1 и лемма 2 без доказательства.
33. Семантический анализ. Атрибутная грамматика. Граф зависимости в атрибутной грамматике. Синтаксическое дерево и даг в операторной грамматике.
34. Вычисление атрибутов, совмещенное с LR-анализом (восходящим анализ)
35. Алгоритм устранения непосредственной левой рекурсии в схеме
36. трансляции S-атрибутной грамматики. Доказательство корректности.
37. Основные понятия восходящего анализа с маркером. Алгоритм построения маркерной грамматики, его обоснование. Упрощение маркерной грамматики.

Практическая часть (задачи)

1. Проверить, принадлежат ли данные слова языку, порождаемому данной грамматике (можно без алгоритма Кока-Янгера-Касами).
2. Найти КС-грамматику, порождающую данный язык.
3. Проверить, является ли однозначной данная КС-грамматика.
4. Для данной грамматики найти эквивалентную ей приведенную грамматику.
5. В данной грамматике устранить циклы.
6. Для данной грамматики найти эквивалентную ей ϵ -свободную грамматику.
7. Привести данную грамматику в НФХ.
8. Найти КС-грамматику в ХНФ, порождающую данный язык.
9. Проанализировать цепочку данной длины, используя алгоритм Кока-Янгера-Касами (СУК).
10. Для данной цепочки, используя алгоритм Кока-Янгера-Касами (СУК), построить дерево вывода и левосторонний вывод.
11. Доказать, используя теорему о накачке, что данный язык не является КС.
12. Доказать, для данного унарного языка, используя теорему о накачке или теорему об унарных языках, что данный язык не является КС.
13. Построить в рамках лексического анализа ДКА для данного шаблона.
14. Методом рекурсивного спуска проверить, принадлежат ли данные цепочки языку, порождаемому данной $LL(1)$ грамматикой. Если да, то построить соответствующее дерево вывода.
15. В данной грамматике устранить левую рекурсию.
16. В данной грамматике выполнить левую факторизацию.
17. Вычислить отношения предшествования для грамматики. Является ли грамматика ПП-грамматикой, СП-грамматикой?
18. Построить таблицу приоритета операторов по грамматике.
19. Построить $LR(0)$ -автомат для данной расширенной грамматики. Будет ли данная грамматика $LR(0)$ -грамматикой, $SLR(1)$ -грамматикой?
20. Найти начальное состояние $LR(0)$ -автомата расширенной грамматики.
21. Построить $LR(1)$ -автомат для данной расширенной грамматики. Будет ли данная грамматика $LR(1)$ -грамматикой?
22. Построить $LR(1)$ -анализатор для расширенной грамматики. Показать обработчик ошибок в нескольких ячейках.
23. Является ли $LALR(1)$ -грамматикой грамматика.
24. Построить LR^* -анализатор для языка алгебраических выражений без скобок (например, для выражений вида $xx+xxx+x$) с учетом приоритета операций, построив соответствующую грамматику и $LALR(1)$ -автомат.
25. Построить атрибутивную грамматику, вычисляющую значение выражения.
26. Найти выражение, для которого синтаксическое дерево имеет данное количество уровней.
27. Построить синтаксическое дерево и даг выражения логики высказываний.