

1 - 45 вопросы надо выбрать правильный ответ и дать обоснование
ответа письменно

1. Синтаксис языков описывается:

способы описания синтаксиса языка можно представить в виде схемы:

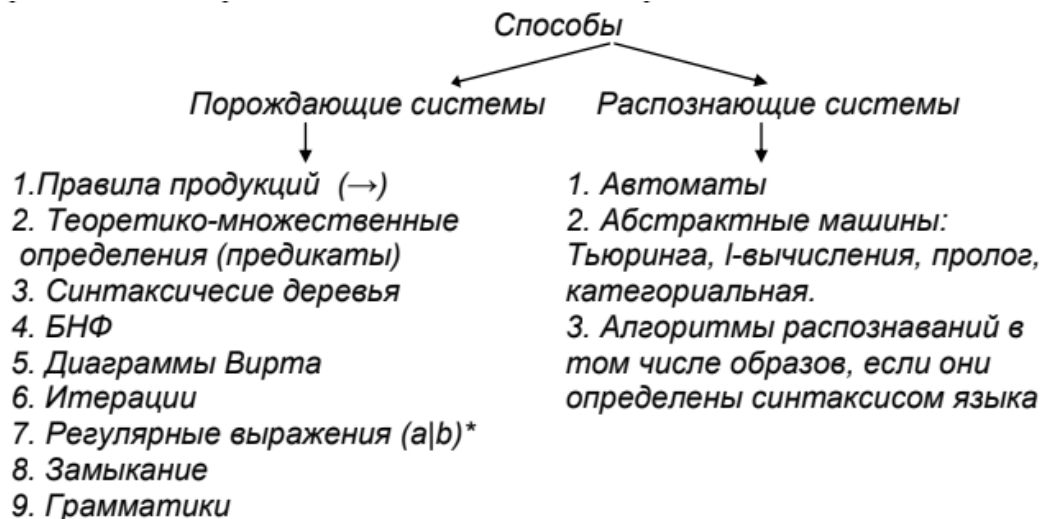


Рис. 1.1. Способы описания синтаксиса языков

2. Грамматика языка реализуется:

Можно представить её распознающим автоматом из порождающей системы

Программная реализация, в виде порождающей системы

3. $G = (T, V, P, S_0)$

Определение 1. Грамматика $G = (T, V, P, S_0)$,
где T - конечное множество терминальных символов (терминалов) алфавита;
 V - конечное множество нетерминальных символов алфавита, не пересекающихся с T , $T \cap V = \emptyset$,
 S_0 - начальный символ (или аксиома), $S_0 \in V$;
 P - конечное множество правил порождения (продукций), $P = (T \cup V)^+ \times (T \cup V)^*$. Элемент (α, β) множества P называется *правилом порождения* и записывается в виде $\alpha \rightarrow \beta$.

4. Языком, порождаемым грамматикой $G = (T, V, P, S_0)$, называется

Определение 2. Языком, порождаемым грамматикой $G = (T, V, P, S_0)$, называется множество $L(G) = \{\alpha \in T^* \mid S_0 \Rightarrow^* \alpha\}$.

Другими словами, $L(G)$ - это все цепочки в алфавите T , которые выводимы из S_0 с помощью P .

Цепочка $\alpha \in (T \cup V)^*$, для которой $S_0 \Rightarrow^* \alpha$ (то есть цепочка, которая может быть выведена из начального символа), называется *сентенциальной формой* в грамматике G . Язык, порождаемый грамматикой, можно определить как множество терминальных сентенциальных форм.

5. $\alpha = \text{defgabck} : |\alpha| = ?$

$|\alpha|$ - длина цепочки α . Для данной цепочки равно 8.

6. Грамматика: $S \rightarrow 0|0A \quad A \rightarrow 1B \quad B \rightarrow 0|0A$

По иерархии Хомского - регулярная грамматика (см. таблицу Хомского)

(P.S. по таблице Хомского могут быть устные вопросы)

7. Язык: $\{0(01)^n \mid n^3 0\}$

Доказательство регулярности языка с помощью леммы о накачке или построения КА.

8. Распознаватель: $\{0(01)^n \mid n^3 0\}$

КА (см. таблицу Хомского)

9. $КА = (Q, \delta, q_0, F)$

Определение 4. Конечный автомат (КА) – это пятерка объектов

$КА = (Q, \Sigma, \delta, q_0, F)$, где

Q - конечное множество состояний;

Σ - конечный алфавит входных символов;

δ - функция переходов, задаваемая отображением

$$\delta: Q \times \Sigma \rightarrow Q,$$

где Q - конечное множество подмножеств множества Q ;

q_0 - начальное состояние автомата, $q_0 \in Q$;

$F \subseteq Q$ - множество заключительных состояний.

δ - множество функций переходов

10. Лемма о накачке доказывает

Регулярность или нерегулярность языка

11. КА задается

Диаграммой, таблицей

12. Конфигурация КА

Определение 5. Конфигурация КА – это пара множества $(q, \omega) \in Q \times \Sigma^*$, где $q \in Q$, $\omega \in \Sigma^*$. Конфигурация (q_0, ω) называется *начальной*, а (q, ε) , где $q \in F$, – *заключительной*.

Язык порождаемый грамматикой G эквивалентен языку распознаваемому КА ($L(G) = L(KA)$).

13. $L(KA)$ распознаваемый КА

Языком $L(KA)$, распознаваемым КА называется множество входных цепочек,
 $L(KA) = \{\omega \in \Sigma^* \mid (q_0, \omega) \vdash^k (q, \varepsilon), \text{ где } q \in F\}$

14. Правила КС-грамматики имеют вид

$A \rightarrow \alpha$, где α – цепочка (последовательность) символов (терминальных и нетерминальных).

15. Беспольные символы КС-грамматики 1. устраняются 2. Недостижимы

Недостижимые, не порождающие, цепные

16. $G = (T, V, P, S)$, в нормальной форме Грейбах, если

Определение 7. КС грамматика $G = (T, V, P, S)$ называется грамматикой в нормальной форме Грейбах, если в ней нет ε -правил, т.е. правил вида $A \rightarrow \varepsilon$, и каждое правило из P отличное от $S \rightarrow \varepsilon$, имеет вид $A \rightarrow a\alpha$, где $a \in T$, $\alpha \in V^*$.

1. Отсутствуют ε -правила
2. Все правила начинаются с терминала: $A \rightarrow a\alpha$

17. Нормальная форма Хомского позволяет упростить рассмотрение свойств:

Не будет цепных правил, левой рекурсии, длинных правил, ε -правил.

18. $G = (T, V, P, S)$ в нормальной форме Хомского, если каждое правило из P имеет один из следующих видов:

Определение 8. КС грамматика $G = (T, V, P, S)$ называется грамматикой в нормальной форме Хомского, если каждое правило из P имеет один из следующих видов:

1. $A \rightarrow BC$, где $A, B, C \in V$;
2. $A \rightarrow a$, где $a \in T$;
3. $S \rightarrow \varepsilon$, если $\varepsilon \in L(G)$, причем S не встречается в правых частях правил.

19. Нетерминал КС-грамматики рекурсивен

Нетерминал КС-грамматики называется *рекурсивным*, если $A \Rightarrow^+ \alpha A \beta$, для некоторых α и β . Если $\alpha = \varepsilon$, то A называется *леворекурсивным*, если $\beta = \varepsilon$, то A называется *праворекурсивным*. Грамматика, имеющая хотя бы один леворекурсивный нетерминал, называется *леворекурсивной*. Грамматика, имеющая хотя бы один праворекурсивный, нетерминал называется *праворекурсивной*.

(устранение левой рекурсии: $A \rightarrow \beta \mid \beta A', A' \rightarrow \alpha A' \mid \alpha$)

20. МП = (Q, Σ , Γ , δ , q_0 , z_0 , F) (МП - магазинная память)

Определение 9. МП автомат – это семерка объектов
 $МП = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$
 Q – конечное множество состояний устройства управления;
 Σ – конечный алфавит входных символов;
 Γ – конечный алфавит магазинных символов;
 δ – функция переходов, отображает множества $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*$ в множество конечных подмножеств множества $Q \times \Gamma^*$;
 q_0 – начальное состояние, $q_0 \in Q$;
 z_0 – начальный символ магазина, $z_0 \in \Gamma$;
 F – множество заключительных состояний, $F \subseteq Q$.

δ - множество функций переходов, $\delta : Q \times (\Sigma \cup \varepsilon) \times \Gamma^* \rightarrow Q \times \Gamma^*$

(в отличие от КА добавляется магазинная память/ стек)

21. Конфигурация МП-автомата:

Определение 10. Конфигурацией МП-автомата называется тройка $(q, \omega, z) \in Q \times \Sigma^* \times \Gamma^*$, где
 q – текущее состояние управляющего устройства;
 ω – необработанная часть входной цепочки (первый символ цепочки ω находится под входной головкой; если $\omega = \varepsilon$, то считается, что вся входная цепочка прочитана);
 z – содержимое магазина (самый левый символ цепочки z считается верхним символом магазина; если $z = \varepsilon$, то магазин считается пустым).

22. Детерминированным МП-автомат

В правой части множество будет содержать один элемент (т.е. для каждой конфигурации можно определенно выбрать один переход или переход на другую конфигурацию определен однозначно)

Пример: $(q_i, a, s) = \{(q_j, s)\}, \forall q_i, q_j \in Q$

23. 1. FIRST(A) =

FIRST(A) – это множество первых терминальных символов, которыми начинаются цепочки, выводимые из нетерминала $A \in V$:

$$\text{FIRST}(A) = \{a \in T \mid A \Rightarrow^+ a\beta \cup \{\epsilon \text{ if } A \Rightarrow^* \epsilon\}, \text{ где } \beta \in (T \cup V)^*\}$$

24. Конфигурация “1-предсказывающего” алгоритма разбора имеет вид (x, Xa^\wedge, π) , где x – неиспользуемая часть входной цепочки, Xa – цепочка в магазине

??? (TODO: пересмотреть на MSTeams) (Пересмотрел, так и не понял в чем вопрос)

Конфигурацию “1-предсказывающего” алгоритма разбора будем представлять в виде (x, Xa^\perp, π) , где x – неиспользуемая часть входной цепочки, Xa – цепочка в магазине, а π – цепочка на выходной ленте, отражающая состояние алгоритма.

25. Управляющая таблица применяется для распознавания

... LL(1)/ LL(k) грамматик LR

23, 24 и 25 вопросы смешались

26. Для LR(k) языка можно построить анализатор

LR(k) проводит просмотр слева, разбор справа (т.е. разбор восходящий)

В конце разбора LR(k) на стеке должен остаться S_0 (начальный символ) грамматики

В отличие от управляющего устройства LL(k)-грамматики в управляющем устройстве LR(k)-грамматики добавляются пары функций (f, g) - (действия, перехода)

27. В LR(k) разборе применяются операции

Свертки и переноса

28. goto задается

Определение. Функция $GOTO(I, X)$ определяется как замыкание множества всех ситуаций $[A \rightarrow \alpha X \beta]$, таких что $[A \rightarrow \alpha \cdot X \beta] \in I$

29. Для грамматики предшествования строится

На основе таблицы предшествования?

Пропущено в базовом курсе. Самостоятельно разобрать хотя бы общее определение грамматики предшествования (Алгоритм “сдвиг-свертка”)

30. LR(k)-алгоритм разбора описывается

Двумя способами: способом активных префиксов и способом грамматических вхождений

31. $G = (T, V, P, S)$ - грамматика типа 0 если

Нет ограничений (см. таблицу Хомского)

32. Конфигурация машины Тьюринга

Машина Тьюринга - детерминированный преобразователь. Конфигурация машины - множество элементов следующего вида:

$\Sigma^* \times Q \times \Sigma^+$, то есть последовательности $w_1 q_1 a_j w_2$, где w_1 и $w_2 \in \Sigma^*$, $a_j \in \Sigma$, $q_i \in Q$.

Правила из δ задают последовательные переходы от одной конфигурации (до тех пор, пока не будет достигнута заключительная конфигурация).

Конфигурация $w_1 q_1 a_j w_2$ называется заключительной, если к ней неприменимо ни одно из правил δ (т.е. правил с заголовком $q_1 a_j$).

33. $G = (T, V, P, S)$ – неукорачивающаяся грамматика, если каждое правило из P имеет вид $\alpha \rightarrow \beta$, где

Определение 17. Грамматика $G = (T, V, P, S)$ называется *неукорачивающей грамматикой*, если каждое правило из P имеет вид $\alpha \rightarrow \beta$, где $\alpha \subset (T \cup V)^+$, $\beta \subset (T \cup V)^+$ и $|\alpha| \leq |\beta|$.

34. $G = (T, V, P, S)$ - контекстно-зависимая (КЗ), если каждое правило из P имеет вид $\alpha \rightarrow \beta$, где

Определение 18. Грамматика $G = (T, V, P, S)$ называется *контекстно-зависимой (КЗ)*, если каждое правило из P имеет вид $\alpha \rightarrow \beta$, где $\alpha = \xi_1 A \xi_2$; $\beta = \xi_1 \gamma \xi_2$; $A \subset V$; $\gamma \subset (T \cup V)^+$; $\xi_1, \xi_2 \subset (T \cup V)^*$.

35. Грамматику типа 1 можно определить как

Неукорачивающую контекстно-зависимую (КЗ) грамматику.

$\alpha \rightarrow \beta, |\alpha| \leq |\beta|$

36. Любая регулярная грамматика является

ответ на вопрос 36-40

Соотношения между типами грамматик:

- 1) любая регулярная грамматика является КС-грамматикой;
- 2) любая регулярная грамматика является укорачивающей КС-грамматикой (УКС). Отметим, что УКС-грамматика, содержащая правила вида $A \rightarrow \varepsilon$, не является КЗ-грамматикой и не является неукорачивающей грамматикой;
- 3) любая (приведенная) КС-грамматика является КЗ-грамматикой;
- 4) любая (приведенная) КС-грамматика является неукорачивающей грамматикой;
- 5) любая КЗ-грамматика является грамматикой типа 0.
- 6) любая неукорачивающая грамматика является грамматикой типа 0.

37. Любая КС-грамматика является грамматикой типа

38. Любая КС-грамматика является грамматикой типа

39. Любая КЗ-грамматика является грамматикой типа

40. Любая неукорачивающая грамматика является грамматикой типа

41. Каждый регулярный язык является

ответ на вопрос 41-43

Определение 19. Язык $L(G)$ является языком типа k , если его можно описать грамматикой типа k .

Соотношения между типами языков:

- 1) каждый регулярный язык является КС-языком, но существуют КС-языки, которые не являются регулярными (например, $L = \{a^n b^n \mid n > 0\}$);
- 2) каждый КС-язык является КЗ-языком, но существуют КЗ-языки, которые не являются КС-языками (например, $L = \{a^n b^n c^n \mid n > 0\}$);
- 3) каждый КЗ-язык является языком типа 0. УКС-язык, содержащий пустую цепочку, не является КЗ-языком. Если язык задан грамматикой типа k , то это не значит, что не существует грамматики типа k' ($k' > k$), описывающей тот же язык. Поэтому, когда говорят о языке типа k , обычно имеют в виду максимально возможный номер k .

42. Каждый КС-язык является

43. Каждый КЗ-язык является языком типа

44- 83 дать определение или привести пример письменно

44. Определите понятие семантики языков программирования

(В лекциях - 5 и 6 главы)

Семантика - вычисление алгоритмов: сложение, вычитание, “другие там операции”...

45. Простейшие переводы описываются:

Существуют два подхода к описанию перевода:

1. использование системы, аналогичной формальной грамматике или являющейся её расширением, которая порождает перевод (точнее пару цепочек, принадлежавших переводу);
2. применение автомата, распознающего входную цепочку и выдающего на выходе перевод этой цепочки.

46. Определите синтаксически управляемый перевод и входную и выходную цепочки СУ-схемы

Определение. Пусть Σ — входной алфавит и Δ — выходной алфавит. Переводом с языка $L_1 \subseteq \Sigma^*$ на язык $L_2 \subseteq \Delta^*$ называется отношение τ из $\Sigma^* \times \Delta^*$, для которого L_1 — область определения, а L_2 — множество значений.

Если $(x, y) \in \tau$, то Цепочка y называется выходом для цепочки x . В общем случае в переводе τ для одной входной цепочки может быть задано более одной выходной цепочки.

Для языков программирования перевод должен быть функцией, т. е. для каждой входной программы должно быть не более одной выходной программы.

47. Определите L-атрибутную и S-атрибутную транслирующие грамматики

Сравнить с определением грамматики

Вводится словарь операционных символов (операция), объединение двух словарей

Транслирующая грамматика – это КС-грамматика, множество терминалов которой разбито на два множества: множество входных и множество операционных символов.

Дадим формальное определение транслирующей грамматики.

Определение: транслирующей грамматикой называется пятерка объектов $G^T = (N, \Sigma_i, \Sigma_a, P, S)$, где Σ_i — словарь входных символов, Σ_a — словарь операционных символов, N — нетерминальный словарь, $S \in N$ — начальный символ транслирующей грамматики, P — конечное множество правил вывода вида $A \rightarrow \alpha$, в которых $A \in N$, а $\alpha \in (\Sigma_i \cup \Sigma_a \cup N)^*$.

Σ_i - словарь входных символов

Определение: АТ-грамматика называется L-атрибутной тогда и только тогда, когда выполняются следующие условия:

1. Аргументами правила вычисления значения унаследованного атрибута символа из правой части правила вывода могут быть только унаследованные атрибуты символа из левой части и произвольные атрибуты символов из правой части, расположенные левее рассматриваемого символа.
2. Аргументами правила вычисления значения синтезированного атрибута символа из левой части правила вывода являются унаследованные атрибуты этого символа или произвольные атрибуты символов из правой части.
3. Аргументами правила вычисления значения синтезированного атрибута операционного символа могут быть только унаследованные атрибуты этого символа.

Строится дерево зависимостей для вычисления значений атрибутов (в процессе нисходящего анализа входной цепочки для L- атрибутной грамматики)

Определение: АТ-грамматика называется S-атрибутной тогда и только тогда, когда она является L-атрибутной и все атрибуты нетерминалов синтезированные.

48. Определите копирующее правило вычисления атрибутов

Форма простого присваивания

Определение: правило вычисления атрибутов называется копирующим правилом тогда и только тогда, когда левая часть правила — это атрибут или список атрибутов, а правая часть – константа или атрибут. Правая часть называется источником копирующего правила, а каждый атрибут из левой части – приемником копирующего правила.

49. В каком случае АТ-грамматика имеет форму простого присваивания?

Определение: атрибутная транслирующая грамматика имеет форму простого присваивания тогда и только тогда, когда:
атрибутов операционных символов.
копирующих правил независимо.

Более понятное определение из другой книги:

Атрибутная транслирующая грамматика имеет форму простого присваивания тогда и только тогда, когда:



1. Некопирующими являются только правила вычисления синтезированных атрибутов операционных символов.
 2. Для каждого правила вывода грамматики соответствующее множество копирующих правил независимо.
-

50. Расширить восходящий анализатор действиями по выполнению перевода.

51. Представление в магазине и вычисление унаследованных и синтезированных атрибутов символов грамматики в S-Атрибутом ДМП-процессоре.

(Стр. 142)

Для любого восходящего синтаксического анализатора, рассматриваемого в данном учебнике, последовательность операций переноса и свертки, выполняемых при обработке допустимых входных цепочек, можно описать с по-

мощью транслирующей грамматики. Входной для этой транслирующей грамматики является грамматика, на основе которой построен анализатор. Для получения транслирующей грамматики в самую крайнюю правую позицию каждого i -го правила входной грамматики вставляется операционный символ {СВЕРТКА, i }. Например, транслирующая грамматика, построенная по

Обратить внимание на **представление в магазине**

52. Случай АТ-грамматика с формой простого присваивания.

(Повторяет 49)

53. Приведите процедуру преобразования произвольной АТ-грамматики в форму простого присваивания.

(стр. 126)

54. Преобразование деревьев под управлением СУ-схем

(Алгоритм 5.1., страница 94)

Своими словами; обратить внимание: что подается на вход, что получаем на выходе

55. Приведите алгоритм построения управляющей таблицы для транслирующей грамматики цепочного перевода, входной грамматикой которого является LL(1)-грамматика.

56. Определите транслирующую грамматику.

см. вопрос 47

57. Опишите, каким образом представляются в магазине и вычисляются унаследованные и синтезированные атрибуты символов грамматики в L-атрибутом ДМП-процессоре.

Нисходящий детерминированный процессор с магазинной памятью (нисходящий ДМП-процессор).

58. Какие СУ-схемы называются простыми.

Примечание: СУ-схема $T = (N, \Sigma, \Delta, R, S)$ называется простой, если для каждого правила $A \rightarrow \alpha, \beta \in R$ соответствующие друг другу вхождения не терминалов встречаются в α и β в одном и том же порядке.

(+ простой пример, использующий, например, два из шести доступных правила грамматики)

59. Приведите процедуру преобразования ДМП-процессора в L-атрибутный ДМП-процессор.

(Стр. 132)

60. Определите активную цепочку, ее входную и операционную части.

Рассматривая процесс перевода инфиксных арифметических выражений в польскую инверсную запись, попытаемся построить некоторый процессор, выполняющий этот перевод.

Если на входной ленте процессора находится цепочка $a + b * c$, то процессор должен работать следующим образом:

- ☐ прочитайте входной символ, a ;
- ☐ выдать символ, a на выходную ленту;
- ☐ почитать входной символ '+';
- ☐ почитать входной символ b ;
- ☐ выдать символ b на выходную ленту;
- ☐ почитать входной символ '*';
- ☐ почитать входной символ c ;
- ☐ выдать символ c на выходную ленту;
- ☐ выдать символ '*' на выходную ленту;
- ☐ выдать символ '+' на выходную ленту.

Этот план работы определяется тем, что порядок следования операндов в инфиксной записи и ПОЛИЗ совпадает, а операции в ПОЛИЗ должны следовать сразу за своими операндами. Если операцию Чтения символа из входной ленты мы будем обозначать Читаемыми символами, а операцию записи — символами, помещаемыми на выходную ленту, заключёнными в фигурные скобки (операционными символами), то работа по переводу может быть записана следующим образом:

$a \{a\} + b \{b\} * c \{c\} \{*\} \{+\}$.

Транслирующая грамматика — это КС-грамматика, множество

61. Как программируются процедуры в методе рекурсивного спуска, реализующего перевод, описываемый L-атрибутной транслирующей грамматикой.

62. Определите входную и выходную грамматики транслирующей грамматики

63. Как программируются процедуры в методе рекурсивного спуска, реализующие перевод, описываемый транслирующей грамматикой.

(Повтор 61)

64. Как граф зависимостей используется при вычислении атрибутов

65. С какой целью осуществляется переименование имен атрибутов нетерминальных символов из левых частей правил вывода L-атрибутной транслирующей грамматики при реализации вывода L-атрибутного перевода методом рекурсивного спуска.

Для осуществления вычислений при помощи присваивания

(Вопрос будет пересмотрен)

66. Как связаны вывод цепочки и дерево вывода активной цепочки

67. С какой целью осуществляется переименование имен атрибутов нетерминальных символов из левых частей правил вывода L-атрибутной транслирующей грамматики при реализации вывода L-атрибутного перевода методом рекурсивного спуска

(Повторяет 65, удален)

68. Определите атрибутную транслирующую грамматику.

Атрибутная транслирующая грамматика - это транслирующая грамматика, обладающая следующими дополнительными свойствами:

1. Каждый символ грамматики (входной, нетерминальный и операционный) имеет конечное множество атрибутов, и каждый атрибут имеет (возможно, бесконечное) множество допустимых значений.

2. Все атрибуты нетерминальных и операционных символов делятся на синтезированные и унаследованные.

3. Унаследованные атрибуты подчиняются следующим правилам:

- каждому вхождению унаследованного атрибута в правую часть правила вывода сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов символов, входящих в левую или правую часть данного правила;

- для каждого унаследованного атрибута начального символа грамматики задаётся начальное значение.

4. Правила вычисления значений синтезированных атрибутов определяются следующим образом:

- каждому вхождению синтезированного атрибута нетерминального символа в левую часть правила вывода сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов символов, входящих в левую или правую часть данного правила;

- каждому синтезированному атрибуту операционного символа сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов данного символа.

- каждому синтезированному атрибуту операционного символа сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов данного символа.

69. Опишите, каким образом представляются в магазине и вычисляются унаследованные и синтезированные атрибуты символов грамматики в L-атрибутом ДМП-процессоре

70. Какие атрибуты называются унаследованными, синтезированными.

Атрибуты, значения которых вычисляются при движении по дереву вывода снизу вверх, традиционно называют синтезированными. Термин "синтезированный" подчёркивает, что значение атрибута синтезируется из значений атрибутов потомков.

Термин "унаследованный" означает то, что значение атрибута зависит от значений атрибутов предка символа или атрибутов его соседей в дереве вывода.

71. С какой целью осуществляется переименование имен атрибутов нетерминальных символов из левых частей правил вывода L-атрибутной транслирующей грамматики при реализации вывода L-атрибутного перевода методом рекурсивного спуска.

(Удален)

72. Определите различия между унаследованными и синтезированными атрибутами

Возможный ответ (От Максимки):

Синтезированные атрибуты получают значения при движении по дереву вывода снизу вверх(от листьев), а унаследованные наоборот: либо при движении сверху вниз (от корня), либо от соседей на одной вершине дерева, но в тоже время "сосед" может быть синтезированным атрибутом.

73. Определите представление в магазине и вычисление унаследованных и синтезированных атрибутов символов грамматики в S-Атрибутом ДМП-процессоре.

74. Определите правило вычисления атрибутов.

(Повторяющийся вопрос)

75. Приведите алгоритм построения управляющей таблицы для транслирующей грамматики цепочного перевода, входной грамматикой которого является LL(1)-грамматика

76. Как строится граф зависимостей.

На основе атрибутов (синтезируемые и унаследованные)

77. Случай АТ-грамматика с формой простого присваивания.

78. Типы ошибок при тестировании АТ-грамматики.

Неправильно заданы атрибуты или правила присваивания атрибутов

Тестирование АТ-Грамматики позволяет определять два вида ошибок:

- ошибки в структуре выходной цепочки (неверный порядок следования символов в выходной цепочке);
- ошибки в значениях символов выходной цепочки, что связано с ошибками в передаче значений атрибутов.

79. Приведите процедуру преобразования ДМП-процессора в L-атрибутный ДМП-процессор.

Добавление в LL-управляющую таблицу операционных символов?

80. Этапы построения транслирующей грамматики

81. Определите представление в магазине и вычисление унаследованных и синтезированных атрибутов символов грамматики в S-Атрибутом ДМП-процессоре.

см. вопрос 73

82. Виды объектов включаемые в выходную цепочку при переводе.

2.Выходная цепочка (результат перевода) включает в себя объекты (сущности) трех видов:

- объекты, передаваемые в выходную цепочку из входной цепочки;
- объекты, не изменяющиеся в процессе перевода (терминалы выходного языка);
- объекты, генерируемые во время перевода.

83. Определите L-атрибутную и S-атрибутную транслирующие грамматики.

см. вопрос 47.

To Do List

- построить правильную структуру файла.
 - пояснение: справа в браузерной версии есть структура файл, в которой отображаются все заголовки и названия. Нужно сделать так, чтобы каждый вопрос был заголовком для быстрого перемещения по файлу. Если у кого есть ещё идеи, то добавляйте сюда, если есть желание помочь, то делайте то, что считаете нужным из списка.
- дополнить отсутствующие(неотвеченные) вопросы.
- радоваться жизни и делать лабы и курсач. :)