Лингвистические основы информатики
Лекция 17, часть 1
Восходящий анализ. Грамматики простого предшествования

Ю. В. Нагребецкая

Уральский федеральный университет
Институт естественных наук и математики
Департамент математики, механики и компьютерных наук

Направления: Математика и компьютерные науки
Компьютерная безопасность
(6 семестр)

Восходящий анализ. Основные понятия

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС, приведенная, ε-свободная, однозначная грамматика.

Восходящий анализ. Основные понятия

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС, приведенная, ε-свободная, однозначная грамматика.
- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод в грамматике G. Цепочка α_i называется r-формой.

Восходящий анализ. Основные понятия

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС, приведенная, ε-свободная, однозначная грамматика.
- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод в грамматике G. Цепочка α_i называется r-формой.
- Напомним, что основой r-формы назвается подслово этой r-формы, являющееся правой частью некоторого правила грамматики, а именно: $\alpha_i = \alpha A \beta$, $\alpha_{i+1} = \alpha\gamma\beta$, $(A \to \gamma) \in P$. Ввиду однозначности грамматики основа r-формы определяется единственным образом (увидим ниже).

Восходящий анализ. Основные понятия

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС, приведенная, ε-свободная, однозначная грамматика.
- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод в грамматике G. Цепочка α_i называется r-формой.
- Напомним, что основой r-формы назвается подслово этой r-формы, являющееся правой частью некоторого правила грамматики, а именно: $\alpha_i = \alpha A \beta$, $\alpha_{i+1} = \alpha\gamma\beta$, $(A \to \gamma) \in P$. Ввиду однозначности грамматики основа r-формы определяется единственным образом (увидим ниже).
- Пусть T — соответствующее дерево вывода.

Восходящий анализ. Основные понятия

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС, приведенная, ε-свободная, однозначная грамматика.
- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод в грамматике G. Цепочка α_i называется r-формой.
- Напомним, что основой r-формы назвается подслово этой r-формы, являющееся правой частью некоторого правила грамматики, а именно: $\alpha_i = \alpha A \beta$, $\alpha_{i+1} = \alpha\gamma\beta$, $(A \to \gamma) \in P$. Ввиду однозначности грамматики основа r-формы определяется единственным образом (увидим ниже).
- Пусть T — соответствующее дерево вывода.
- Поддерево K дерева T называется кустом, если оно вершинно порождено (что это означает?) отцом f некотрого листа s, у которого все братья являются листьями, и всеми этими братьями вершины s.

Восходящий анализ. Основные понятия

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС, приведенная, ε-свободная, однозначная грамматика.
- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод в грамматике G. Цепочка α_i называется r-формой.
- Напомним, что основой r-формы назвается подслово этой r-формы, являющееся правой частью некоторого правила грамматики, а именно: $\alpha_i = \alpha A \beta$, $\alpha_{i+1} = \alpha\gamma\beta$, $(A \to \gamma) \in P$. Ввиду однозначности грамматики основа r-формы определяется единственным образом (увидим ниже).
- Пусть T — соответствующее дерево вывода.
- Поддерево K дерева T называется кустом, если оно вершинно порождено (что это означает?) отцом f некотрого листа s, у которого все братья являются листьями, и всеми этими братьями вершины s.
- Упорядочим все узлы дерева T при помощи поиска в глубину с выбором самого левого сына. Этот порядок $\preceq$ на узлах породит соответственно порядок $\preceq$ на корнях кустов, т.е. на кустах (см. рис.1).

Куст. Иллюстрация

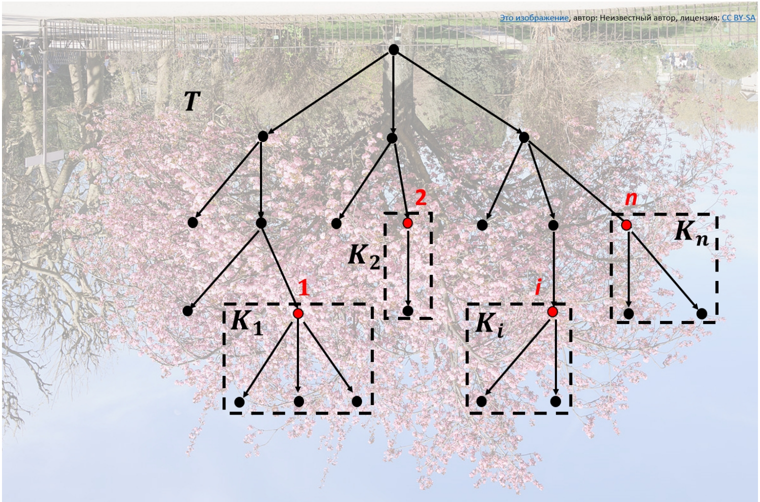


Рис. 1

Восходящий анализ. Основные понятия

- Пусть поддерево T_i дерева T вершинно порождено всеми символами формы α_i. Дерево T_i назовем стандартным.

Восходящий анализ. Основные понятия

- Пусть поддерево T_i дерева T вершинно порождено всеми символами формы α_i. Дерево T_i назовем стандартным.
- Тогда дерево T_i получается из T_{i+1} удалением всех листьев самого левого куста (почему?).

Восходящий анализ. Основные понятия

- Пусть поддерево T_i дерева T вершинно порождено всеми символами формы α_i. Дерево T_i назовем стандартным.
- Тогда дерево T_i получается из T_{i+1} удалением всех листьев самого левого куста (почему?).
- Это значит, что на листьях самого левого куста стандартного поддерева слева направо написана основа r-формы, представленной этим поддеревом (почему?) (см. рис. 11).

Получение дерево T_i получается из T_{i+1}. Иллюстрация

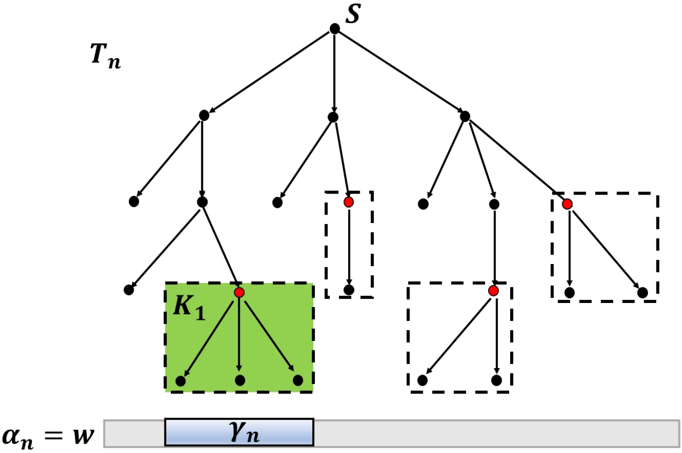


Рис. 2

Получение дерево T_i получается из T_{i+1}. Иллюстрация

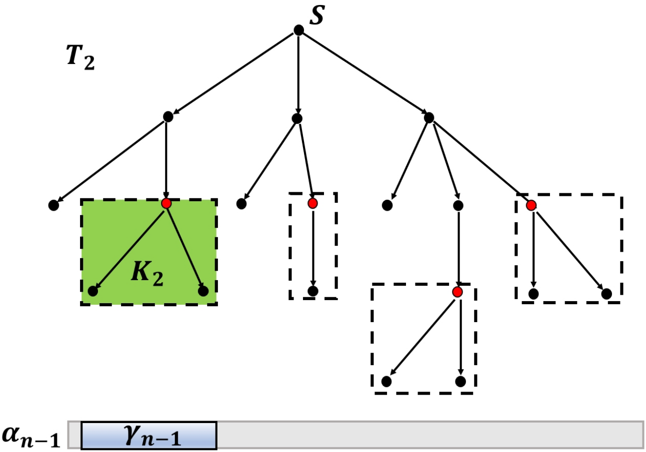


Рис. 3

Получение дерево T_i получается из T_{i+1}. Иллюстрация

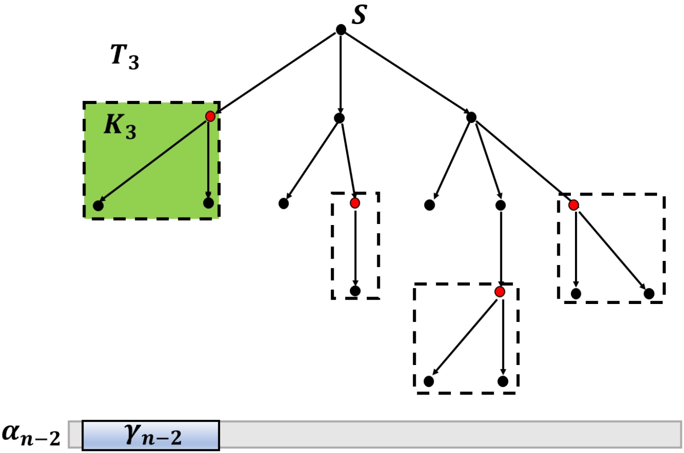


Рис. 4

Получение дерево T_i получается из T_{i+1}. Иллюстрация

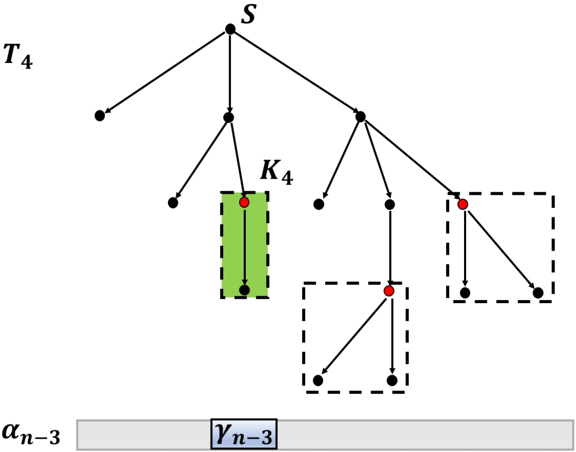


Рис. 5

Получение дерево T_i получается из T_{i+1}. Иллюстрация

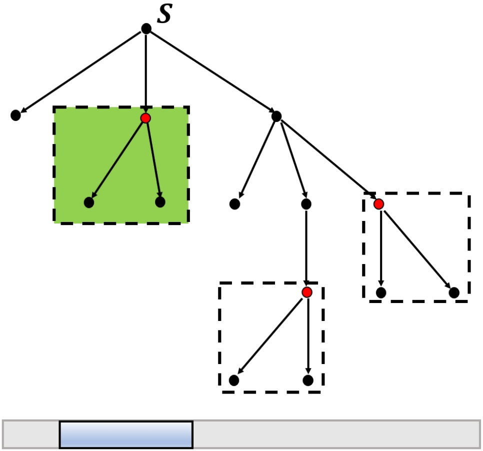


Рис. 6

Получение дерево T_i получается из T_{i+1}. Иллюстрация

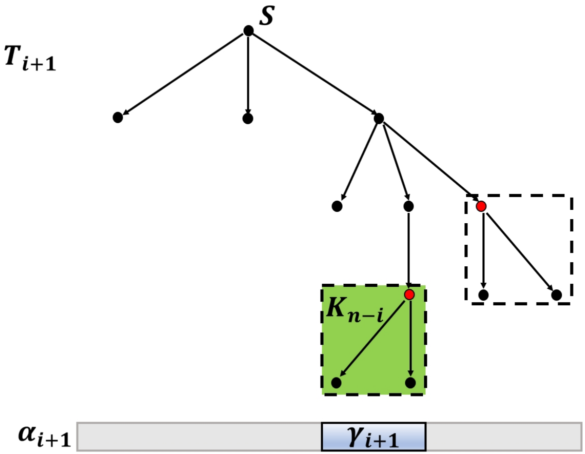


Рис. 7

Получение дерево T_i получается из T_{i+1}. Иллюстрация

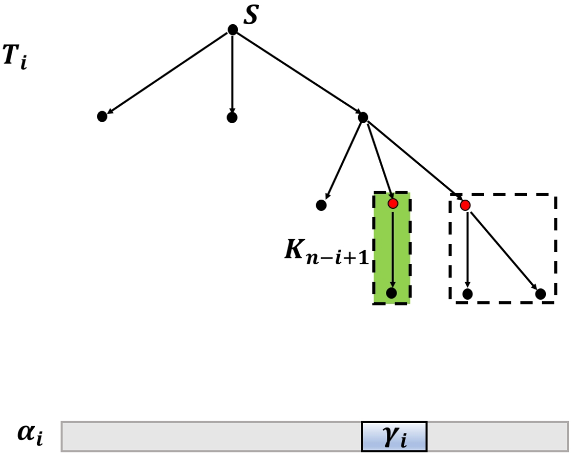


Рис. 8

Получение дерево T_i получается из T_{i+1}. Иллюстрация

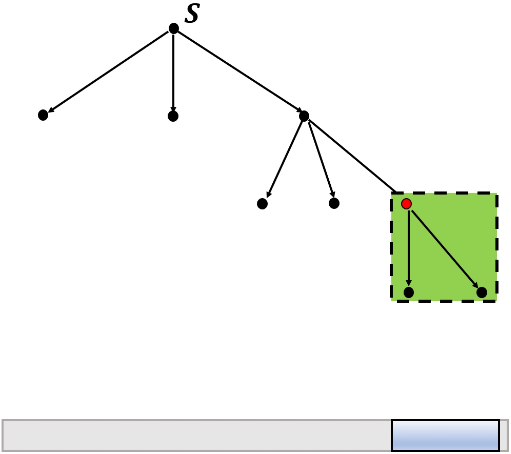


Рис. 9

Получение дерево T_i получается из T_{i+1}. Иллюстрация

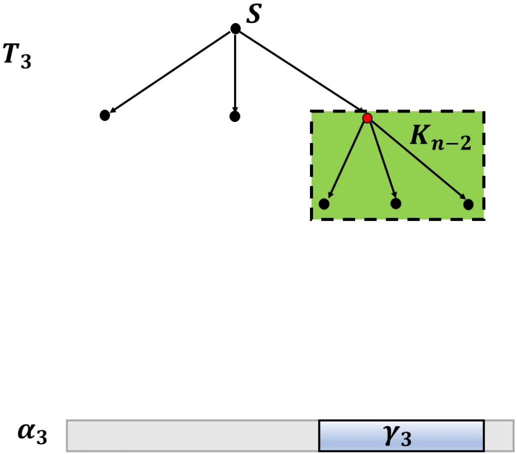


Рис. 10

Получение дерево T_i получается из T_{i+1}. Иллюстрация

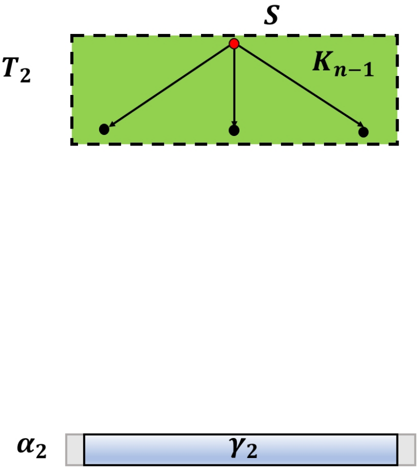


Рис. 11

Получение дерево T_i получается из T_{i+1}. Иллюстрация

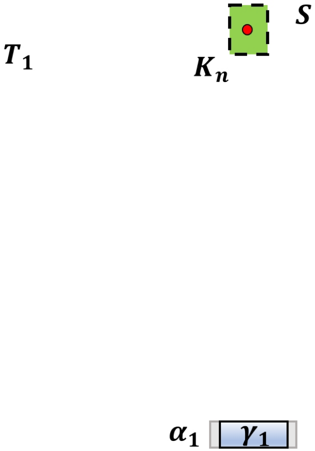


Рис. 12

Восходящий анализ. Основные понятия. Пример 1

- **Пример 1** Дана приведенная, ε-свободная, однозначная грамматика.

$$\begin{aligned} S &\rightarrow aFSd \mid c \\ F &\rightarrow Fb \mid b \end{aligned}$$

Восходящий анализ. Основные понятия. Пример 1

- **Пример 1** Дана приведенная, ε-свободная, однозначная грамматика.

$$S \to aFSd \mid c$$
$$F \to Fb \mid b$$

- Рассмотрим вывод
  $S \Rightarrow \underline{aFSd} \Rightarrow aF\underline{c}d \Rightarrow a\underline{Fb}cd \Rightarrow a\underline{Fb}bcd \Rightarrow a\underline{b}bbcd$

Восходящий анализ. Основные понятия. Пример 1

- **Пример 1** Дана приведенная, ε-свободная, однозначная грамматика.

$$\begin{aligned} S &\to aFSd \mid c \\ F &\to Fb \mid b \end{aligned}$$

- Рассмотрим вывод
$S \Rightarrow \underline{aFSd} \Rightarrow aF\underline{c}d \Rightarrow a\underline{Fb}cd \Rightarrow a\underline{Fb}bcd \Rightarrow a\underline{b}bbcd$

- $\alpha_1 = S,$
$\alpha_2 = \boxed{aFSd}, \ \gamma_2 = \boxed{aFSd}$ — основа
$\alpha_3 = aF\boxed{c}d, \ \gamma_3 = \boxed{c}$ — основа
$\alpha_4 = a\boxed{Fb}cd, \ \gamma_4 = \boxed{Fb}$ — основа
$\alpha_5 = a\boxed{Fb}bcd \ \gamma_5 = \boxed{Fb}$ — основа
$\alpha_6 = a\boxed{b}bbcd \ \gamma_6 = \boxed{b}$ — основа

Восходящий анализ. Основные понятия. Пример 1

- Пример 1 Дана приведенная, ε-свободная, однозначная грамматика.

$$S \to aFSd \mid c$$
$$F \to Fb \mid b$$

- Рассмотрим вывод
  $S \Rightarrow \underline{aFSd} \Rightarrow aF\underline{c}d \Rightarrow a\underline{Fb}cd \Rightarrow a\underline{Fb}bcd \Rightarrow a\underline{b}bbcd$

- $\alpha_1 = S$,
  $\alpha_2 = \boxed{aFSd}$, $\gamma_2 = \boxed{aFSd}$ — основа
  $\alpha_3 = aF\boxed{c}d$, $\gamma_3 = \boxed{c}$ — основа
  $\alpha_4 = a\boxed{Fb}cd$, $\gamma_4 = \boxed{Fb}$ — основа
  $\alpha_5 = a\boxed{Fb}bcd$ $\gamma_5 = \boxed{Fb}$ — основа
  $\alpha_6 = a\boxed{b}bbcd$ $\gamma_6 = \boxed{b}$ — основа

- Дерево вывода цепочки w приведено на рис.13–19. Процесс "сворачивания" цепочки в аксиому соответствует последовательной "обрезке" кустов $K_1 - K_5$ (поддерево K_2 становится кустом после "обрезки" K_1 и т. д.).

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

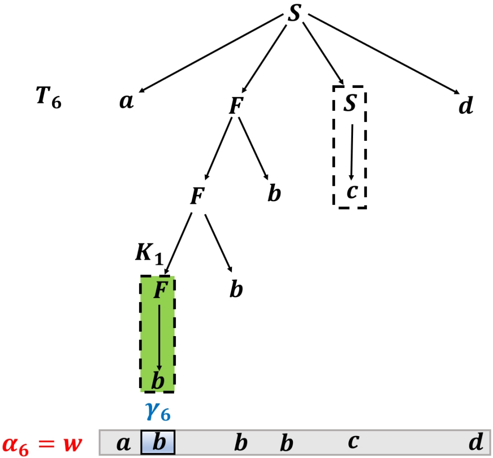


Рис. 13

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

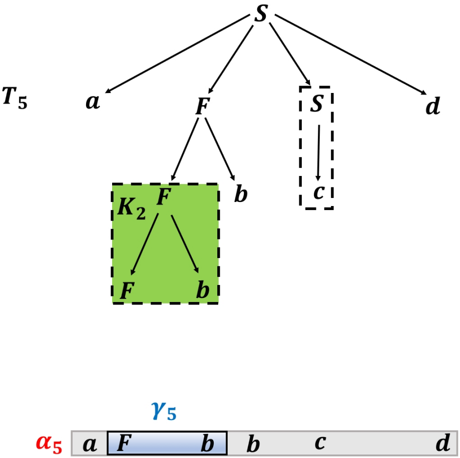


Рис. 14

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

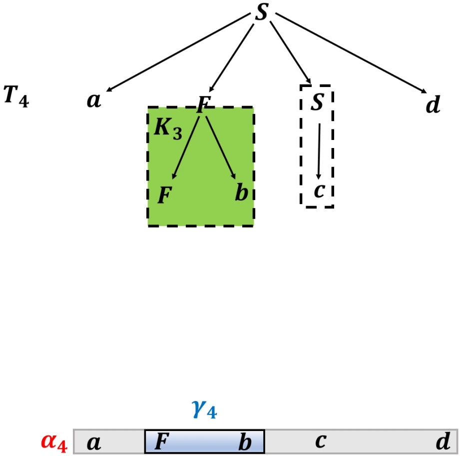


Рис. 15

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

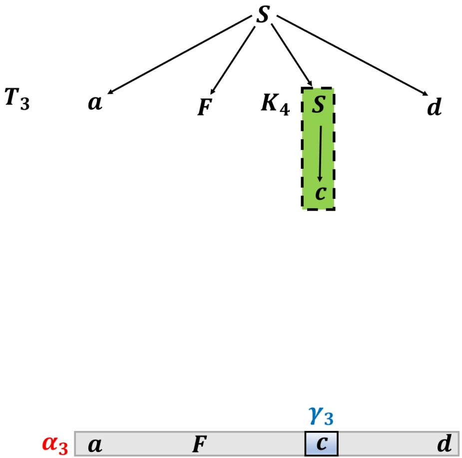


Рис. 16

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

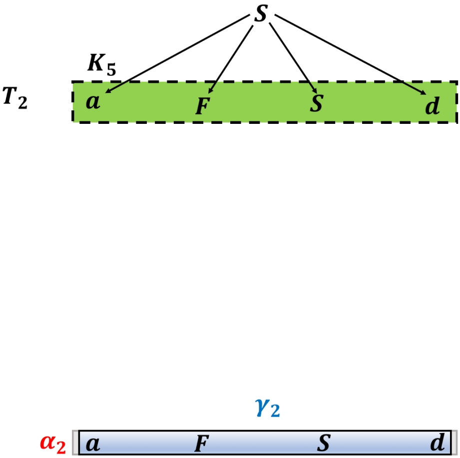


Рис. 17

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

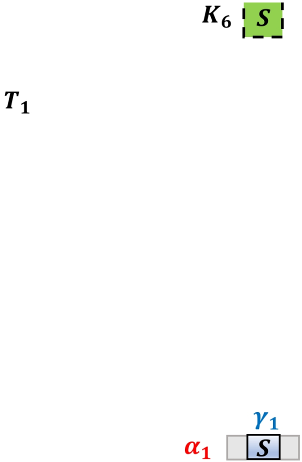


Рис. 18

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.
- Цепочка, записанная на входной ленте, посимвольно переносится в стек до тех пор, пока наверху стека не окажется основа цепочки.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.
- Цепочка, записанная на входной ленте, посимвольно переносится в стек до тех пор, пока наверху стека не окажется основа цепочки.
- Тогда делаем свертку по соответствующему правилу.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.
- Цепочка, записанная на входной ленте, посимвольно переносится в стек до тех пор, пока наверху стека не окажется основа цепочки.
- Тогда делаем свертку по соответствующему правилу.
- Когда делаем свертку, по ленте не сдвигаемся.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.
- Цепочка, записанная на входной ленте, посимвольно переносится в стек до тех пор, пока наверху стека не окажется основа цепочки.
- Тогда делаем свертку по соответствующему правилу.
- Когда делаем свертку, по ленте не сдвигаемся.
- Процедура повторяется, пока не просмотрена вся цепочка.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.
- Цепочка, записанная на входной ленте, посимвольно переносится в стек до тех пор, пока наверху стека не окажется основа цепочки.
- Тогда делаем свертку по соответствующему правилу.
- Когда делаем свертку, по ленте не сдвигаемся.
- Процедура повторяется, пока не просмотрена вся цепочка.
- Все эти действия можно организовать при помощи обычного МП-автомата (как?). В течение этой и следующей глав при записи содержимого стека мы пишем символ дна слева, а верхушку – справа, считая ее последним символом стековой цепочки, а не первым, как до сих пор.

Общая идея восходящего анализа

Восходящие методы реализуются при помощи стека.

- Вначале стек пуст.
- Цепочка, записанная на входной ленте, посимвольно переносится в стек до тех пор, пока наверху стека не окажется основа цепочки.
- Тогда делаем свертку по соответствующему правилу.
- Когда делаем свертку, по ленте не сдвигаемся.
- Процедура повторяется, пока не просмотрена вся цепочка.
- Все эти действия можно организовать при помощи обычного МП-автомата (как?). В течение этой и следующей глав при записи содержимого стека мы пишем символ дна слева, а верхушку – справа, считая ее последним символом стековой цепочки, а не первым, как до сих пор.
- Произведение содержимого стека ζ_i на необработанную часть v_i цепочки w есть r-форма α_i.

Общая идея восходящего анализа. Обоснование

1. Действительно, если наверху стека лежит основа γ_i r-формы α_i, то она соответствует самому левому кусту K_{n-i+1} дерева T_i.

Общая идея восходящего анализа. Обоснование

1. Действительно, если наверху стека лежит основа γ_i r-формы α_i, то она соответствует самому левому кусту K_{n-i+1} дерева T_i.
2. Тогда на листьях поддерева T_i' дерева T_i, вершинно порожденного всеми вершинами, меньшими самой большой вершины куста K_{n-i+1}, написано содержимое ζ_i стека слева-направо. То есть поддерево T_i' дерева T_i является деревом вывода цепочки ζ_i (см. рис. 19).

Общая идея восходящего анализа. Обоснование

1. Действительно, если наверху стека лежит основа γ_i r-формы α_i, то она соответствует самому левому кусту K_{n-i+1} дерева T_i.
2. Тогда на листьях поддерева T_i' дерева T_i, вершинно порожденного всеми вершинами, меньшими самой большой вершины куста K_{n-i+1}, написано содержимое ζ_i стека слева-направо. То есть поддерево T_i' дерева T_i является деревом вывода цепочки ζ_i (см. рис. 19).
3. А на всех остальных листьях дерева T_i написана необработанная часть v_i цепочки w (см. рис. 20).

Общая идея восходящего анализа. Обоснование

1. Действительно, если наверху стека лежит основа γ_i r-формы α_i, то она соответствует самому левому кусту K_{n-i+1} дерева T_i.
2. Тогда на листьях поддерева T_i' дерева T_i, вершинно порожденного всеми вершинами, меньшими самой большой вершины куста K_{n-i+1}, написано содержимое ζ_i стека слева-направо. То есть поддерево T_i' дерева T_i является деревом вывода цепочки ζ_i (см. рис. 19).
3. А на всех остальных листьях дерева T_i написана необработанная часть v_i цепочки w (см. рис. 20).
4. Если на вершине стека лежит только часть γ_i' основы γ_i, т.е. $\zeta_i = \zeta_i'\gamma_i'$, (см. рис. 21), то оставшаяся часть γ_i'' основы является префиксом необработанной части v_i цепочки, т.е. $v_i = \gamma_i''v_i'$, $\gamma_i'' \neq \varepsilon$. Тогда через несколько шагов γ_i'' перенесется в стек, и в стеке будет слово $\zeta_i'\gamma_i$, а на ленте будет написана цепочка v_i'. Тогда по (3) цепочка $\zeta_i'\gamma_i v_i' = \zeta_i'\gamma_i'\gamma_i''v_i' = \zeta_i v_i$ будет r-формой, и таким образом все доказано.

Общая идея восходящего анализа. Обоснование. Иллюстрация

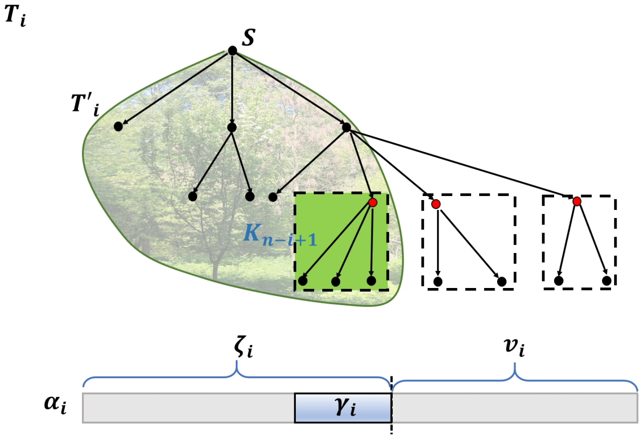


Рис. 19

Общая идея восходящего анализа. Обоснование. Иллюстрация

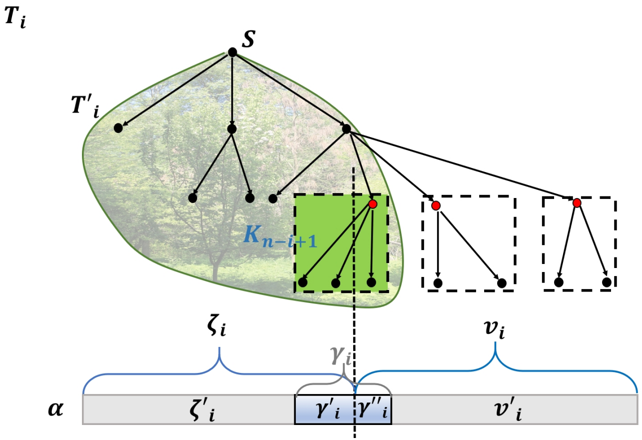


Рис. 20

Общая идея восходящего анализа. Обоснование. Иллюстрация

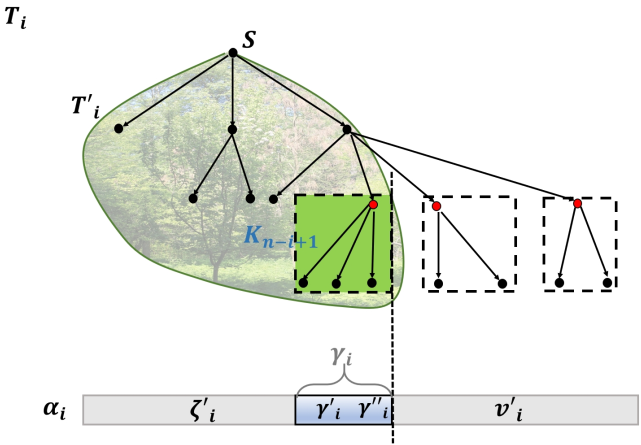


Рис. 21

Протокол для обработки цепочки в примере 1

Рассмотрим протокол обработки цепочки $w = abbbcd$ в примере 1.

№ такта	содержимое стека	позиция указателя
1	∇	$\diamond abbbcd\dashv$
2	∇a	$a\diamond bbbcd\dashv$
3	∇ab	$ab\diamond bbcd\dashv$
4	∇aF	$ab\diamond bbcd\dashv$
5	∇aFb	$abb\diamond bcd\dashv$
6	∇aF	$abb\diamond bcd\dashv$
7	∇aFb	$abbb\diamond cd\dashv$
8	∇aF	$abbb\diamond cd\dashv$
9	∇aFc	$abbbc\diamond d\dashv$
10	∇aFS	$abbbc\diamond d\dashv$
11	$\nabla aFSd$	$abbbcd\diamond\dashv$
12	∇S	$abbbcd\diamond\dashv$

Восстанавливаем вывод снизу вверх

Протокол для обработки цепочки в примере 1

Рассмотрим протокол обработки цепочки $w = abbbcd$ в примере 1.

№ такта	содержимое стека	позиция указателя
1	∇	⋄abbbcd⊣
2	∇a	a⋄bbbcd⊣
3	∇ab	ab⋄bbcd⊣
4	∇aF	ab⋄bbcd⊣
5	∇aFb	abb⋄bcd⊣
6	∇aF	abb⋄bcd⊣
7	∇aFb	abbb⋄cd⊣
8	∇aF	abbb⋄cd⊣
9	∇aFc	abbbc⋄d⊣
10	∇aFS	abbbc⋄d⊣
11	∇aFSd	abbbcd⋄⊣
12	∇S	abbbcd⋄⊣

Восстанавливаем вывод снизу вверх

$S \Rightarrow aFSd \Rightarrow aFSd \Rightarrow aFcd \Rightarrow aFbcd \Rightarrow aFbcd \Rightarrow aFbbcd \Rightarrow abbbcd \Rightarrow abbbcd$

Протокол для обработки цепочки в примере 1

Рассмотрим протокол обработки цепочки $w = abbbcd$ в примере 1.

№ такта	содержимое стека	позиция указателя
1	∇	◇abbbcd⊣
2	∇a	a◇bbbcd⊣
3	∇ab	ab◇bbcd⊣
4	∇aF	ab◇bbcd⊣
5	∇aFb	abb◇bcd⊣
6	∇aF	abb◇bcd⊣
7	∇aFb	abbb◇cd⊣
8	∇aF	abbb◇cd⊣
9	∇aFc	abbbc◇d⊣
10	∇aFS	abbbc◇d⊣
11	∇aFSd	abbbcd◇⊣
12	∇S	abbbcd◇⊣

Восстанавливаем вывод снизу вверх

$S \Rightarrow aFSd \Rightarrow aFSd \Rightarrow aFcd \Rightarrow aFbcd \Rightarrow aFbcd \Rightarrow aFbbcd \Rightarrow abbbcd \Rightarrow abbbcd$

После вычеркивания повторяющихся цепочек, получаем правосторонний вывод: $S \Rightarrow aFSd \Rightarrow aFcd \Rightarrow aFbcd \Rightarrow aFbbcd \Rightarrow abbbcd$

Активный префикс

- **Активным** префиксом r-формы называется ее префикс, который не выходит за правую границу основы.

Активный префикс

- **Активным** префиксом r-формы называется ее префикс, который не выходит за правую границу основы.
- Напрмер, для r-формы $\alpha_4 = a\underline{Fb}cd$ активным префиксом являются следующие слова ε, a, aF, aFb, $aFbc$, $aFbcd$.

Активный префикс

- Активным префиксом r-формы называется ее префикс, который не выходит за правую границу основы.
- Напрмер, для r-формы $\alpha_4 = a\underline{Fb}cd$ активным префиксом являются следующие слова $\varepsilon, a, aF, aFb, aFbc, aFbcd$.

Из обоснования восходящего анализа (см. выше) следует

Активный префикс

- **Активным** префиксом r-формы называется ее префикс, который не выходит за правую границу основы.
- Напрмер, для r-формы $\alpha_4 = a\underline{Fb}cd$ активным префиксом являются следующие слова ε, a, aF, aFb, $aFbc$, $aFbcd$.

Из обоснования восходящего анализа (см. выше) следует

Лемма об активном префиксе

При восходящем анализе к при помощи стека в стеке всегда лежит только активный префикс.

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\cdot$, $\cdot>$ (см.рис.22):

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\cdot$, $\cdot>$ (см.рис.22):
    1. $X \doteq Y$ т. и т.т.к. если XY содержится в основе некоторой r-формы;

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\!\cdot$, $\cdot\!>$ (см.рис.22):
  1. $X \doteq Y$ т. и т.т.к. если XY содержится в основе некоторой r-формы;
  2. $X <\!\cdot Y$ т. и т.т.к. если основа некоторой r-формы начинается с символа Y, перед которым стоит символ X;

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\!\cdot$, $\cdot\!>$ (см.рис.22):
  1. $X \doteq Y$ т. и т.т.к. если XY содержится в основе некоторой r-формы;
  2. $X <\!\cdot Y$ т. и т.т.к. если основа некоторой r-формы начинается с символа Y, перед которым стоит символ X;
  3. $X \cdot\!> Y$ т. и т.т.к. основа некоторой r-формы заканчивается символом X, после которого стоит символ Y.

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\cdot$, $\cdot>$ (см.рис.22):
    1. $X \doteq Y$ т. и т.т.к. если XY содержится в основе некоторой r-формы;
    2. $X<\cdot Y$ т. и т.т.к. если основа некоторой r-формы начинается с символа Y, перед которым стоит символ X;
    3. $X\cdot>Y$ т. и т.т.к. основа некоторой r-формы заканчивается символом X, после которого стоит символ Y.

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\cdot$, $\cdot>$ (см.рис.22):
  1. $X \doteq Y$ т. и т.т.к. если XY содержится в основе некоторой r-формы;
  2. $X <\cdot Y$ т. и т.т.к. если основа некоторой r-формы начинается с символа Y, перед которым стоит символ X;
  3. $X \cdot> Y$ т. и т.т.к. основа некоторой r-формы заканчивается символом X, после которого стоит символ Y.
  4. $X \cdot> \dashv$ т. и т.т.к. на X заканчивается какая-либо r-форма

Отношения предшествования. Замечание 1

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика и $M = \Sigma \cup \Gamma \cup \{\dashv\} \cup \{\vdash\}$.
- Зададим на множестве M бинарные отношениями простого предшествования $\doteq$, $<\!\cdot$, $\cdot\!>$ (см.рис.22):
    1. $X \doteq Y$ т. и т.т.к. если XY содержится в основе некоторой r-формы;
    2. $X <\!\cdot Y$ т. и т.т.к. если основа некоторой r-формы начинается с символа Y, перед которым стоит символ X;
    3. $X \cdot\!> Y$ т. и т.т.к. основа некоторой r-формы заканчивается символом X, после которого стоит символ Y.
    4. $X \cdot\!> \dashv$ т. и т.т.к. на X заканчивается какая-либо r-форма

Замечание 1

Поскольку мы рассматриваем только r-формы, в случае отношения $\cdot\!>$ можно считать, что Y – терминал.

Общая идея восходящего анализа. Обоснование. Иллюстрация

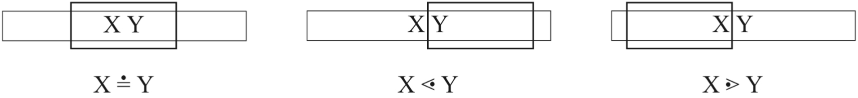


Рис. 22

Теорема об отношениях предшествования

Теорема об отношениях предшествования

Для любых $X, Y \in \Sigma \cup \Gamma$ (см.рис.24).

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;

Теорема об отношениях предшествования

Теорема об отношениях предшествования

Для любых $X, Y \in \Sigma \cup \Gamma$ (см.рис.24).

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X <\!\cdot\, Y$ т. и т.т.к. $(A \to \alpha XZ\beta) \in P$ и $Z \Rightarrow^{+} Y\gamma$;

Замечание 2 (упр.)

Отношения $\cdot\!>$, $<\!\cdot$ необязательно антисимметричны и транзитивны, а отношение $\doteq$ необязательно эквивалентность.

Теорема об отношениях предшествования

Теорема об отношениях предшествования

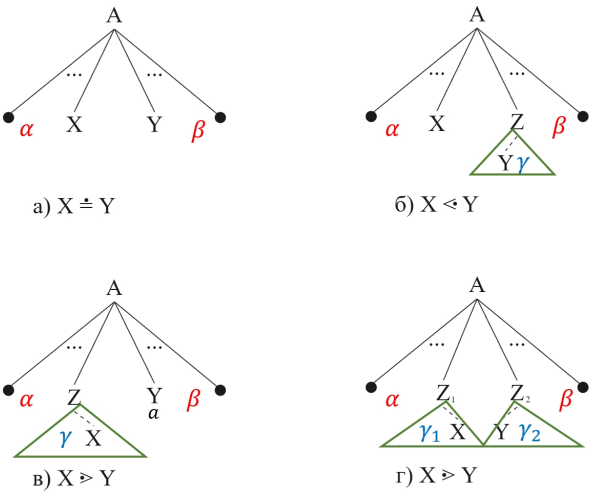
Для любых $X, Y \in \Sigma \cup \Gamma$ (см.рис.24).

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X \lessdot Y$ т. и т.т.к. $(A \to \alpha XZ\beta) \in P$ и $Z \Rightarrow^{+} Y\gamma$;
3. $X \gtrdot Y$ т. и т.т.к. либо $(A \to \alpha ZY\beta) \in P$, $Z \Rightarrow^{+} \gamma X$, $Y \in \Sigma$, либо $(A \to \alpha Z_1 Z_2 \beta) \in P$ и $Z_1 \Rightarrow^{+} \gamma_1 X$, $Z_2 \Rightarrow^{*} Y\gamma_2$.

Замечание 2 (упр.)

Отношения $\gtrdot$, $\lessdot$ необязательно антисимметричны и транзитивны, а отношение $\doteq$ необязательно эквивалентность.

Теорема об отношениях предшествования. Иллюстрация



а) $X \doteq Y$

б) $X \lessdot Y$

в) $X \gtrdot Y$

г) $X \gtrdot Y$

Рис. 23

Теорема об отношениях предшествования

Доказательство.

Теорема об отношениях предшествования

Доказательство.

- (1) следует из определения отношения $X \doteq Y$ (см. рис.23а).

Теорема об отношениях предшествования

Доказательство.

- (1) следует из определения отношения $X \doteq Y$ (см. рис.23а).
- (2) Пусть $X \lessdot Y$, A — ближайший предок X и Y в дереве вывода T. Тогда X — обязательно сын A (см. рис.23б). Если то не так, существовал бы куст K', у которого X — самый правый сын, и тот куст был бы левее куста K'', у которого Y — самый левый сын. Поскольку при восходящем анализе "обрывается" самый левый куст, то X оказался бы самым правым символом основы некоторой r-формы, которая бы свернулась раньше, чем основа, в которой находится Y, т.е. оказалось бы, что $X \gtrdot Y$, что противоречиво.

Теорема об отношениях предшествования

Доказательство.

- (1) следует из определения отношения $X \doteq Y$ (см. рис.23а).
- (2) Пусть $X \lessdot Y$, A — ближайший предок X и Y в дереве вывода T. Тогда X — обязательно сын A (см. рис.23б). Если то не так, существовал бы куст K', у которого X — самый правый сын, и тот куст был бы левее куста K'', у которого Y — самый левый сын. Поскольку при восходящем анализе "обрывается" самый левый куст, то X оказался бы самым правым символом основы некоторой r-формы, которая бы свернулась раньше, чем основа, в которой находится Y, т.е. оказалось бы, что $X \gtrdot Y$, что противоречиво.
- Итак, X — обязательно сын A, причем у X еще есть некоторый правый брат Z, т.е. $(A \to \alpha XZ\beta) \in P$. Так как не выполняется $X \doteq Y$, то $Z \neq Y$, но $Z \Rightarrow^+ Y\gamma$, поскольку $X \lessdot Y$.

Теорема 1 об отношениях предшествования

- Покажем (3). Пусть $X \cdot\!> Y$. Узлы X и Y не могут быть братьями, иначе $X \doteq Y$ (см. рис.), X не может быть сыном общего предка A символов X и Y (см. рис. 23), иначе по доказанному $X \doteq Y$. Следовательно, возможны только два случая:

Теорема 1 об отношениях предшествования

- Покажем (3). Пусть $X \cdot\!> Y$. Узлы X и Y не могут быть братьями, иначе $X \doteq Y$ (см. рис.), X не может быть сыном общего предка A символов X и Y (см. рис. 23), иначе по доказанному $X \doteq Y$. Следовательно, возможны только два случая:
    1. Общий предок X и Y — отец Y. Тогда $(A \to \alpha Z Y \beta) \in P$ и $Z \Rightarrow^{+} \gamma X$ (см. рис.23в)

Теорема 1 об отношениях предшествования

- Покажем (3). Пусть $X \lessdot\!\!> Y$. Узлы X и Y не могут быть братьями, иначе $X \doteq Y$ (см. рис.), X не может быть сыном общего предка A символов X и Y (см. рис. 23), иначе по доказанному $X \doteq Y$. Следовательно, возможны только два случая:
  1. Общий предок X и Y — отец Y. Тогда $(A \to \alpha ZY\beta) \in P$ и $Z \Rightarrow^{+} \gamma X$ (см. рис.23в)
  2. Общий предок X и Y — не отец ни X, ни Y. Тогда $(A \to \alpha Z_1 Z_2 \beta) \in P$ и $Z_1 \Rightarrow^{+} \gamma_1 X$, $Z_2 \Rightarrow^{*} Y\gamma_2$ (см. рис.23г)

Множества $FIRST'$, $LAST'$

Определим множества $FIRST'$ и $LAST'$.

Определение

Пусть $Y \in (\Sigma \cup \Gamma)^*$.

- $FIRST'(Y) = \{X \in \Sigma \cup \Gamma \mid Y \Rightarrow^+ X\gamma, \gamma \in (\Sigma \cup \Gamma)^*\}$ (см.рис.24);

Множества $FIRST'$, $LAST'$

Определим множества $FIRST'$ и $LAST'$.

Определение

Пусть $Y \in (\Sigma \cup \Gamma)^*$.

- $FIRST'(Y) = \{X \in \Sigma \cup \Gamma \mid Y \Rightarrow^+ X\gamma, \gamma \in (\Sigma \cup \Gamma)^*\}$ (см.рис.24);
- $LAST'(Y) = \{X \in \Sigma \cup \Gamma \mid Y \Rightarrow^+ \gamma X, \gamma \in (\Sigma \cup \Gamma)^*\}$ (см.рис.25).

Множества $FIRST'$, $LAST'$

По теореме об отношениях предшествования, а конкретно из рис.26-27 следует

Теорема 2 об отношениях предшествования через $FIRST'$ и $LAST'$

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;

Множества $FIRST'$, $LAST'$

По теореме об отношениях предшествования, а конкретно из рис.26-27 следует

Теорема 2 об отношениях предшествования через $FIRST'$ и $LAST'$

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X \lessdot Y$ т. и т.т.к. существует $Z \in \Gamma$ т.ч. $X \doteq Z$ и $Y \in FIRST'(Z)$ для некоторого $Z \in \Gamma$;

Множества $FIRST'$, $LAST'$

По теореме об отношениях предшествования, а конкретно из рис.26-27 следует

Теорема 2 об отношениях предшествования через $FIRST'$ и $LAST'$

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X \lessdot Y$ т. и т.т.к. существует $Z \in \Gamma$ т.ч. $X \doteq Z$ и $Y \in FIRST'(Z)$ для некоторого $Z \in \Gamma$;
3. $X \gtrdot Y$ т. и т.т.к. либо $Z \doteq Y$ и $X \in LAST'(Z)$ для некоторых $Z \in \Gamma$ и $Y \in \Sigma$, либо $Z_1 \doteq Z_2$ и $X \in LAST'(Z_1)$, $Y \in FIRST'(Z_2)$ для некоторых $Z_1, Z_2 \in \Gamma$.

Множества $FIRST'$, $LAST'$

По теореме об отношениях предшествования, а конкретно из рис.26-27 следует

Теорема 2 об отношениях предшествования через $FIRST'$ и $LAST'$

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X \lessdot Y$ т. и т.т.к. существует $Z \in \Gamma$ т.ч. $X \doteq Z$ и $Y \in FIRST'(Z)$ для некоторого $Z \in \Gamma$;
3. $X \gtrdot Y$ т. и т.т.к. либо $Z \doteq Y$ и $X \in LAST'(Z)$ для некоторых $Z \in \Gamma$ и $Y \in \Sigma$, либо $Z_1 \doteq Z_2$ и $X \in LAST'(Z_1)$, $Y \in FIRST'(Z_2)$ для некоторых $Z_1, Z_2 \in \Gamma$.
4. $\vdash \lessdot X$ т. и т.т.к. $X \in FIRST'(S) \cup \{S\}$.

Множества $FIRST'$, $LAST'$

По теореме об отношениях предшествования, а конкретно из рис.26-27 следует

Теорема 2 об отношениях предшествования через $FIRST'$ и $LAST'$

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X \lessdot Y$ т. и т.т.к. существует $Z \in \Gamma$ т.ч. $X \doteq Z$ и $Y \in FIRST'(Z)$ для некоторого $Z \in \Gamma$;
3. $X \gtrdot Y$ т. и т.т.к. либо $Z \doteq Y$ и $X \in LAST'(Z)$ для некоторых $Z \in \Gamma$ и $Y \in \Sigma$, либо $Z_1 \doteq Z_2$ и $X \in LAST'(Z_1)$, $Y \in FIRST'(Z_2)$ для некоторых $Z_1, Z_2 \in \Gamma$.
4. $\vdash \lessdot X$ т. и т.т.к. $X \in FIRST'(S) \cup \{S\}$.
5. $X \gtrdot \dashv$ т. и т.т.к. $X \in LAST'(S) \cup \{S\}$.

Определение множества FIRST'. Иллюстрация

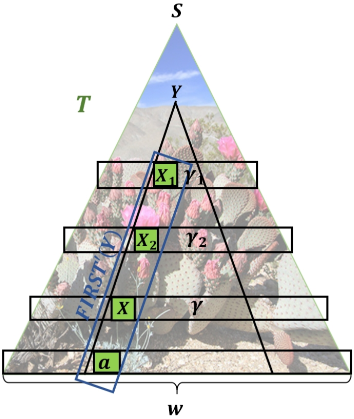


Рис. 24

Определение множества LAST'. Иллюстрация

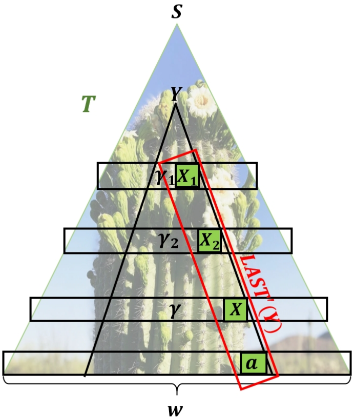
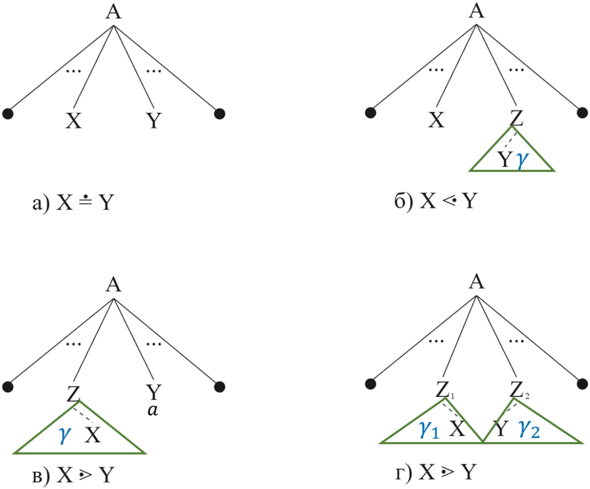


Рис. 25

Теорема 2 об отношениях предшествования через *FIRST′* и *LAST′* Иллюстрация



а) $X \doteq Y$

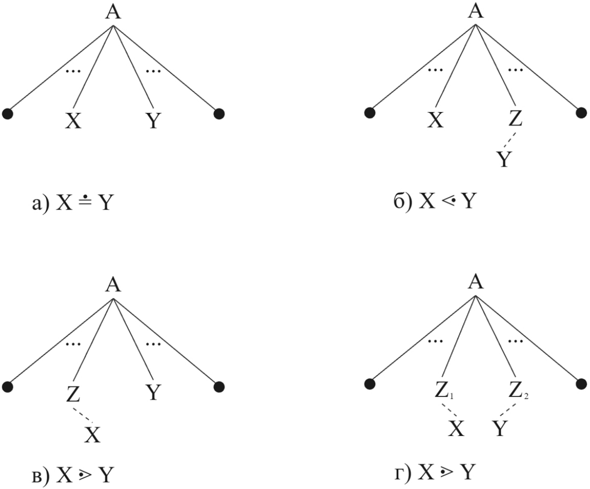
б) $X \lessdot Y$

в) $X \gtrdot Y$

г) $X \gtrdot Y$

Рис. 24

Теорема 2 об отношениях предшествования через *FIRST′* и *LAST′* Иллюстрация



а) $X \doteq Y$

б) $X \lessdot Y$

в) $X \gtrdot Y$

г) $X \gtrdot Y$

Рис. 25

Множества $FIRST'$, $LAST'$ (продолжение)

- В чем отличие от $FIRST'$ от $FIRST$ и как они между собой связаны.

Множества $FIRST'$, $LAST'$ (продолжение)

- В чем отличие от $FIRST'$ от $FIRST$ и как они между собой связаны.
- Как связаны между собой $FIRST'$ и $LAST'$.

Множества $FIRST'$, $LAST'$ (продолжение)

- В чем отличие от $FIRST'$ от $FIRST$ и как они между собой связаны.
- Как связаны между собой $FIRST'$ и $LAST'$.
- Очевидно, $FIRST'(c) = \{c\}$, $LAST'(c) = \{c\}$ для всех $c \in \Sigma$.

Множества $FIRST'$, $LAST'$ (продолжение)

- В чем отличие от $FIRST'$ от $FIRST$ и как они между собой связаны.
- Как связаны между собой $FIRST'$ и $LAST'$.
- Очевидно, $FIRST'(c) = \{c\}$, $LAST'(c) = \{c\}$ для всех $c \in \Sigma$.
- Для нахождения $FIRST'$ для нетерминалов и произвольных цепочек можно модифицировать алгоритм нахождения $FIRST$ для нетерминалов и цепочек, учитывая, что мы рассматриваем только ε-свободные грамматики.

Множества $FIRST'$, $LAST'$ (продолжение)

- В чем отличие от $FIRST'$ от $FIRST$ и как они между собой связаны.
- Как связаны между собой $FIRST'$ и $LAST'$.
- Очевидно, $FIRST'(c) = \{c\}$, $LAST'(c) = \{c\}$ для всех $c \in \Sigma$.
- Для нахождения $FIRST'$ для нетерминалов и произвольных цепочек можно модифицировать алгоритм нахождения $FIRST$ для нетерминалов и цепочек, учитывая, что мы рассматриваем только ε-свободные грамматики.
- Кроме того, очевидно, что нашем случае ε-свободной грамматики $FIRST'(\beta) = FIRST'(X)$, где X — первый символ цепочки $\beta \in (\Sigma \cup \Gamma)^+$ (см. рис. 24).

Множества $FIRST'$, $LAST'$ (продолжение)

- В чем отличие от $FIRST'$ от $FIRST$ и как они между собой связаны.
- Как связаны между собой $FIRST'$ и $LAST'$.
- Очевидно, $FIRST'(c) = \{c\}$, $LAST'(c) = \{c\}$ для всех $c \in \Sigma$.
- Для нахождения $FIRST'$ для нетерминалов и произвольных цепочек можно модифицировать алгоритм нахождения $FIRST$ для нетерминалов и цепочек, учитывая, что мы рассматриваем только ε-свободные грамматики.
- Кроме того, очевидно, что нашем случае ε-свободной грамматики $FIRST'(\beta) = FIRST'(X)$, где X — первый символ цепочки $\beta \in (\Sigma \cup \Gamma)^+$ (см. рис. 24).
- Для нахождения $LAST'$ для нетерминалов и произвольных цепочек можно использовать алгоритм, что $LAST'(\beta) = FIRST'(revers(\beta))$ (что такое реверс слова?)(см. рис. 25).

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $FIRST'(a) = \{a\}$

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $FIRST'(a) = \{a\}$
- for $A \in \Gamma$:

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $FIRST'(a) = \{a\}$
- for $A \in \Gamma$:
  $FIRST'(A) = \emptyset$

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $FIRST'(a) = \{a\}$
- for $A \in \Gamma$:
  $FIRST'(A) = \emptyset$
- while (все множества $FIRST'(A)$ для всех $A \in \Gamma$ не стабилизировались):

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $FIRST'(a) = \{a\}$
- for $A \in \Gamma$:
  $FIRST'(A) = \emptyset$
- while (все множества $FIRST'(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \ldots X_n) \in P$:

Алгоритм построения множеств FIRST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST'(Y)$ для всех терминалов и нетерминалов Y

```
for $a \in \Sigma$:
    $FIRST'(a) = \{a\}$
for $A \in \Gamma$:
    $FIRST'(A) = \emptyset$
while (все множества $FIRST'(A)$ для всех $A \in \Gamma$ не стабилизировались):
    for $(A \to X_1 X_2 \dots X_n) \in P$:
        $FIRST'(A) = FIRST'(A) \cup FIRST'(X_1)$
```

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $LAST'(a) = \{a\}$

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $LAST'(a) = \{a\}$
- for $A \in \Gamma$:

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $LAST'(a) = \{a\}$
- for $A \in \Gamma$:
  $LAST'(A) = \emptyset$

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

- for $a \in \Sigma$:
  $LAST'(a) = \{a\}$
- for $A \in \Gamma$:
  $LAST'(A) = \emptyset$
- while (все множества $LAST'(A)$ для всех $A \in \Gamma$ не стабилизировались):

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

```
for $a \in \Sigma$:
    $LAST'(a) = \{a\}$
for $A \in \Gamma$:
    $LAST'(A) = \emptyset$
while (все множества $LAST'(A)$ для всех $A \in \Gamma$ не стабилизировались):
    for $(A \to X_1X_2 \ldots X_n) \in P$:
```

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $LAST'(Y)$ для всех терминалов и нетерминалов Y

```
for $a \in \Sigma$:
    $LAST'(a) = \{a\}$
for $A \in \Gamma$:
    $LAST'(A) = \emptyset$
while (все множества $LAST'(A)$ для всех $A \in \Gamma$ не стабилизировались):
    for $(A \to X_1 X_2 \dots X_n) \in P$:
        $LAST'(A) = LAST'(A) \cup LAST'(X_n)$
```

Восходящий анализ. Основные понятия. Пример 2

Пример 2

- Дана грамматика $S \to aSSb \mid c$

Восходящий анализ. Основные понятия. Пример 2

Пример 2

- Дана грамматика $S \to aSSb \mid c$
- Она, очевидно, приведенная, ε-свободная, однозначная.

Восходящий анализ. Основные понятия. Пример 2

Пример 2

- Дана грамматика $S \to aSSb \mid c$
- Она, очевидно, приведенная, ε-свободная, однозначная.
- Вычислим множества $FIRST'$, $LAST'$ и определим отношения $\doteq$, $<\cdot$, $\cdot>$ на множестве $\Sigma \cup \{\dashv\} \cup \{\vdash\}$ определению.

Восходящий анализ. Основные понятия. Пример 2

Пример 2

- Дана грамматика $S \rightarrow aSSb \mid c$
- Она, очевидно, приведенная, ε-свободная, однозначная.
- Вычислим множества $FIRST'$, $LAST'$ и определим отношения $\doteq$, $<\cdot$, $\cdot>$ на множестве $\Sigma \cup \{\dashv\} \cup \{\vdash\}$ определению.

	$FIRST'$	$LAST'$
S	a, c	b, c

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a$, $Z = S$, $Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a$, $a \lessdot c$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a,\ a \lessdot c$
- $\boxed{aSSb}$: $S \doteq S$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a,\ a \lessdot c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S \lessdot a,\ S \lessdot c$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a$, $Z = S$, $Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a$, $a \lessdot c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S$, $Z = S$, $Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S \lessdot a$, $S \lessdot c$
- $Z_1 = S$, $Z_2 = S$, $X \in \{b, c\} \subseteq LAST'(Z_1)$, $Y \in \{a, c\} \subseteq FIRST(Z_1) \Rightarrow b \gtrdot a$, $b \gtrdot c$, $c \gtrdot a$, $c \gtrdot c$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a,\ a \lessdot c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S \lessdot a,\ S \lessdot c$
- $Z_1 = S,\ Z_2 = S,\ X \in \{b, c\} \subseteq LAST'(Z_1),\ Y \in \{a, c\} \subseteq FIRST(Z_1) \Rightarrow b \gtrdot a$, $b \gtrdot c,\ c \gtrdot a,\ c \gtrdot c$
- $Z = S,\ Y = S$, но $Y = S$ — нетерминал.

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a{<\!\cdot}a,\ a{<\!\cdot}c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S{<\!\cdot}a,\ S{<\!\cdot}c$
- $Z_1 = S,\ Z_2 = S,\ X \in \{b, c\} \subseteq LAST'(Z_1),\ Y \in \{a, c\} \subseteq FIRST(Z_1) \Rightarrow b{\cdot\!>}a,\ b{\cdot\!>}c,\ c{\cdot\!>}a,\ c{\cdot\!>}c$
- $Z = S,\ Y = S$, но $Y = S$ — нетерминал.
- $\boxed{aSSb}$: $S \doteq b$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a,\ a \lessdot c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S \lessdot a,\ S \lessdot c$
- $Z_1 = S,\ Z_2 = S,\ X \in \{b, c\} \subseteq LAST'(Z_1),\ Y \in \{a, c\} \subseteq FIRST(Z_1) \Rightarrow b \gtrdot a,\ b \gtrdot c,\ c \gtrdot a,\ c \gtrdot c$
- $Z = S,\ Y = S$, но $Y = S$ — нетерминал.
- $\boxed{aSSb}$: $S \doteq b$
- $Z = S,\ Y = b,\ X \in \{b, c\} \subseteq LAST'(Z) \Rightarrow b \gtrdot b,\ c \gtrdot b$

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a \lessdot a,\ a \lessdot c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S \lessdot a,\ S \lessdot c$
- $Z_1 = S,\ Z_2 = S,\ X \in \{b, c\} \subseteq LAST'(Z_1),\ Y \in \{a, c\} \subseteq FIRST(Z_1) \Rightarrow b \gtrdot a,\ b \gtrdot c,\ c \gtrdot a,\ c \gtrdot c$
- $Z = S,\ Y = S$, но $Y = S$ — нетерминал.
- $\boxed{aSSb}$: $S \doteq b$
- $Z = S,\ Y = b,\ X \in \{b, c\} \subseteq LAST'(Z) \Rightarrow b \gtrdot b,\ c \gtrdot b$
- $\vdash \lessdot S,\ \vdash \lessdot a,\ \vdash \lessdot c$ т. и т.т.к. $FIRST'(S) \cup \{S\} = \{S, a, c\}$.

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

- $\boxed{aSSb}$: $a \doteq S$
- $X = a,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow a <\cdot a,\ a <\cdot c$
- $\boxed{aSSb}$: $S \doteq S$
- $X = S,\ Z = S,\ Y \in \{a, c\} \subseteq FIRST'(Z) \Rightarrow S <\cdot a,\ S <\cdot c$
- $Z_1 = S,\ Z_2 = S,\ X \in \{b, c\} \subseteq LAST'(Z_1),\ Y \in \{a, c\} \subseteq FIRST(Z_1) \Rightarrow b \cdot> a,$ $b \cdot> c,\ c \cdot> a,\ c \cdot> c$
- $Z = S,\ Y = S$, но $Y = S$ — нетерминал.
- $\boxed{aSSb}$: $S \doteq b$
- $Z = S,\ Y = b,\ X \in \{b, c\} \subseteq LAST'(Z) \Rightarrow b \cdot> b,\ c \cdot> b$
- $\vdash <\cdot S,\ \vdash <\cdot a,\ \vdash <\cdot c$ т. и т.т.к. $FIRST'(S) \cup \{S\} = \{S, a, c\}$.
- $S \cdot> \dashv,\ b \cdot> \dashv,\ c \cdot> \dashv,$ т. и т.т.к. $LAST'(S) \cup \{S\} = \{S, b, c\}$.

Восходящий анализ. Основные понятия. Пример 2

Получим таблицу:

	S	a	b	c	$\dashv$
S	$\doteq$	$<\cdot$	$\doteq$	$<\cdot$	$\cdot>$
a	$\doteq$	$<\cdot$		$<\cdot$	
b		$\cdot>$	$\cdot>$	$\cdot>$	$\cdot>$
c		$\cdot>$	$\cdot>$	$\cdot>$	$\cdot>$
$\vdash$	$<\cdot$	$<\cdot$		$<\cdot$	

Грамматика простого предшествования

Замечание 2

Любые символы $X, Y \in \Sigma \cup \Gamma$ находятся в некоторой r-форме т.и.т.к. между ними стоит какой-нибудь знак.

Доказательство следует может быть только ситуация, изображенная рис.24-25, а ситуаций как на рис.26 быть не может.

Грамматика простого предшествования

Замечание 2

Любые символы $X, Y \in \Sigma \cup \Gamma$ находятся в некоторой r-форме т.и.т.к. между ними стоит какой-нибудь знак.

Доказательство следует может быть только ситуация, изображенная рис.24-25, а ситуаций как на рис.26 быть не может.

Определение

Грамматика называется грамматикой простого предшествования (ПП), если между любыми двумя символами не более одного отношения.

Грамматика простого предшествования

Замечание 2

Любые символы $X, Y \in \Sigma \cup \Gamma$ находятся в некоторой r-форме т.и.т.к. между ними стоит какой-нибудь знак.

Доказательство следует может быть только ситуация, изображенная рис.24-25, а ситуаций как на рис.26 быть не может.

Определение

Грамматика называется грамматикой простого предшествования (ПП), если между любыми двумя символами не более одного отношения.

Пример 1 Грамматика в примере 2 является грамматикой простого предшествования.

Доказательство замечания 2. Иллюстрация

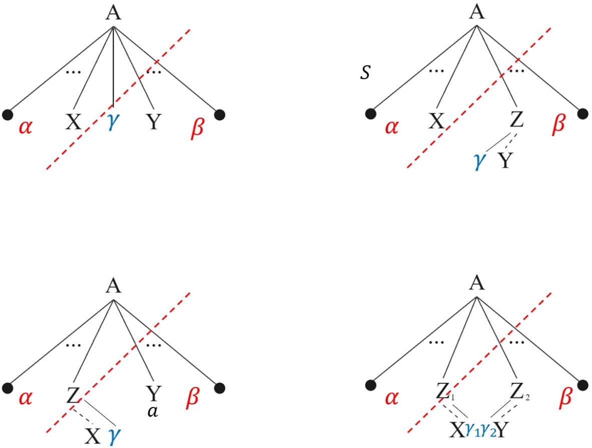


Рис. 26

Грамматика простого предшествования. Теорема 3

Теорема 3

Пусть G — ПП-грамматика, γ – ее r-форма и $\vdash \gamma \dashv = X_0X_1...X_nX_{n+1}$. Тогда основой формы γ является цепочка $X_kX_{k+1}\dots X_{l-1}X_l$ такая, что l — минимальный номер, для которого выполнено $X_l \cdot\!> X_{l+1}$, $1 \le k \le l \le n$, $X_{k-1} <\!\cdot X_k \doteq X_{k+1} \doteq \dots \doteq X_{l-1} \doteq X_l \cdot\!> X_{l+1}$.

Грамматика простого предшествования. Теорема 3

Теорема 3

Пусть G — ПП-грамматика, γ – ее r-форма и $\vdash \gamma \dashv = X_0 X_1 ... X_n X_{n+1}$. Тогда основой формы γ является цепочка $X_k X_{k+1} \dots X_{l-1} X_l$ такая, что l — минимальный номер, для которого выполнено $X_l \cdot\!\!> X_{l+1}$, $1 \le k \le l \le n$, $X_{k-1} <\!\!\cdot X_k \doteq X_{k+1} \doteq \dots \doteq X_{l-1} \doteq X_l \cdot\!\!> X_{l+1}$.

- **Доказательство от противного.** Пусть $X_r X_{r+1} \dots X_{s-1} X_s$ — реальная основа данной r-формы, $1 \le r \le s \le n$ и $X_r X_{r+1} \dots X_{s-1} X_s \neq X_k X_{k+1} \dots X_{l-1} X_l$.

Грамматика простого предшествования. Теорема 3

Теорема 3

Пусть G — ПП-грамматика, γ – ее r-форма и $\vdash \gamma \dashv = X_0X_1 ... X_nX_{n+1}$. Тогда основой формы γ является цепочка $X_kX_{k+1}\dots X_{l-1}X_l$ такая, что l — минимальный номер, для которого выполнено $X_l \cdot\!> X_{l+1}$, $1 \leq k \leq l \leq n$, $X_{k-1} <\!\cdot X_k \doteq X_{k+1} \doteq \dots \doteq X_{l-1} \doteq X_l \cdot\!> X_{l+1}$.

- **Доказательство от противного.** Пусть $X_rX_{r+1}\dots X_{s-1}X_s$ — реальная основа данной r-формы, $1 \leq r \leq s \leq n$ и $X_rX_{r+1}\dots X_{s-1}X_s \neq X_kX_{k+1}\dots X_{l-1}X_l$.
- Тогда по определению отношений предшествования $X_{r-1} <\!\cdot X_r \doteq X_{r+1} \doteq \dots = X_{s-1} \doteq X_s \cdot\!> X_{s+1}$.

Грамматика простого предшествования. Теорема 3

Теорема 3

Пусть G — ПП-грамматика, γ – ее r-форма и $\vdash \gamma \dashv = X_0X_1...X_nX_{n+1}$. Тогда основой формы γ является цепочка $X_kX_{k+1}\dots X_{l-1}X_l$ такая, что l — минимальный номер, для которого выполнено $X_l \cdot\!> X_{l+1}$, $1 \le k \le l \le n$, $X_{k-1} <\!\cdot X_k \doteq X_{k+1} \doteq \dots \doteq X_{l-1} \doteq X_l \cdot\!> X_{l+1}$.

- **Доказательство от противного.** Пусть $X_rX_{r+1}\dots X_{s-1}X_s$ — реальная основа данной r-формы, $1 \le r \le s \le n$ и $X_rX_{r+1}\dots X_{s-1}X_s \ne X_kX_{k+1}\dots X_{l-1}X_l$.
- Тогда по определению отношений предшествования $X_{r-1} <\!\cdot X_r \doteq X_{r+1} \doteq \dots = X_{s-1} \doteq X_s \cdot\!> X_{s+1}$.
- Поскольку мы имеем дело с грамматикой ПП, основы $X_rX_{r+1}\dots X_{s-1}X_s$ и $X_kX_{k+1}\dots X_{l-1}X_l$ не перекрываются, т.е. либо $s < k$, либо $r > l$.

Грамматика простого предшествования. Теорема 3

- Например, не может быть, ни такой ситуации
  $X_{k-1} \lessdot X_k \doteq X_{k+1} \doteq \ldots \doteq X_{l-1} \doteq X_l \gtrdot X_{l+1}$
  $X_{r-1} \lessdot X_r \doteq X_{r+1} \doteq \ldots \doteq X_{s-1} \doteq X_s \gtrdot X_{s+1}$

Грамматика простого предшествования. Теорема 3

- Например, не может быть, ни такой ситуации
  $X_{k-1} \lessdot X_k \doteq X_{k+1} \doteq \ldots \doteq X_{l-1} \doteq X_l \gtrdot X_{l+1}$
  $X_{r-1} \lessdot X_r \doteq X_{r+1} \doteq \ldots \doteq X_{s-1} \doteq X_s \gtrdot X_{s+1}$
- ни такой —
  $X_{r-1} \lessdot X_r \doteq X_{r+1} \doteq \ldots \doteq X_{s-1} \doteq X_s \gtrdot X_{s+1}$
  $X_{k-1} \lessdot X_k \doteq X_{k+1} \doteq \ldots \doteq X_{l-1} \doteq X_l \gtrdot X_{l+1}$

Грамматика простого предшествования. Теорема 3

- Например, не может быть, ни такой ситуации
  $X_{k-1} \lessdot X_k \doteq X_{k+1} \doteq \ldots \doteq X_{l-1} \doteq X_l \gtrdot X_{l+1}$
  $X_{r-1} \lessdot X_r \doteq X_{r+1} \doteq \ldots \doteq X_{s-1} \doteq X_s \gtrdot X_{s+1}$
- ни такой —
  $X_{r-1} \lessdot X_r \doteq X_{r+1} \doteq \ldots \doteq X_{s-1} \doteq X_s \gtrdot X_{s+1}$
  $X_{k-1} \lessdot X_k \doteq X_{k+1} \doteq \ldots \doteq X_{l-1} \doteq X_l \gtrdot X_{l+1}$
- Таким образом, $s < k$ либо $r > l$. Но по условию ввиду того, что l выбиралось минимальным, случая $s < k$ не может быть. Значит, $r > l$, и основа $X_r X_{r+1} \ldots X_{s-1} X_s$ находится строго правее основы $X_k X_{k+1} \ldots X_{l-1} X_l$. Но это противоречит тому, что при восходящем анализе "обрывается" самый левый куст, и основой r-формы γ была бы основа $X_k X_{k+1} \ldots X_{l-1} X_l \neq X_r X_{r+1} \ldots X_{s-1} X_s$.

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma =\vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
    $n$ =длина $\gamma$
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0 \quad k = 0$
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$    $k = 0$    $l = 0$
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \dots X_l$)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \ldots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \ldots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \ldots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i \lessdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \ldots X_l$ )
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \ldots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i \lessdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \ldots X_l$ )
      $k = i + 1$
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i <\cdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \dots X_l$ )
      $k = i + 1$
    case ($X_i \cdot> X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \dots X_l$)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i \lessdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \dots X_l$ )
      $k = i + 1$
    case ($X_i \gtrdot X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \dots X_l$)
      $l = i$
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \ldots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i \lessdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \ldots X_l$ )
      $k = i + 1$
    case ($X_i \gtrdot X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \ldots X_l$)
      $l = i$
    (else: $X_i \doteq X_{i+1}$)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i \lessdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \dots X_l$ )
      $k = i + 1$
    case ($X_i \gtrdot X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \dots X_l$)
      $l = i$
    (else: $X_i \doteq X_{i+1}$)
    $i = +1$ (двигаемся дальше, пока не дойдем до конца основы)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq\vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \ldots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i <\cdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \ldots X_l$ )
      $k = i + 1$
    case ($X_i \cdot> X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \ldots X_l$)
      $l = i$
    (else: $X_i \doteq X_{i+1}$)
    $i = +1$ (двигаемся дальше, пока не дойдем до конца основы)
  if ($l = 0$) or ($\nexists A$: $A \to X_k \ldots X_l$): (не найден правый конец основы или не существует
соответствующего правила для основы)
    print('$w \notin L(G)$')   exit (выход из программы)
```

Восходящий анализ для ПП-грамматики

Вход: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i <\cdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \dots X_l$ )
      $k = i + 1$
    case ($X_i \cdot> X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \dots X_l$)
      $l = i$
    (else: $X_i \doteq X_{i+1}$)
    $i = +1$ (двигаемся дальше, пока не дойдем до конца основы)
  if ($l = 0$) or ($\nexists A$: $A \to X_k \dots X_l$): (не найден правый конец основы или не существует соответствующего правила для основы)
    print('$w \notin L(G)$')   exit (выход из программы)
  else: (когда нашли основу, произвели свертку по соответствующему правилу $A \to X_k \dots X_l$)
```

Восходящий анализ для ПП-грамматики

<u>Вход</u>: ПП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
<u>Выход</u>: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$ (заметим, что либо $X_0 \lessdot X_1$ либо $X_0 \circ X_1$)
($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print($\gamma$)
($\gamma$ — текущая $r$-форма) (печать текущей $r$-формы)
  $i = 0$   $k = 0$   $l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены левый $X_k$ и правый $X_l$ конец основы $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit (выход из программы)
    case ($X_i \lessdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \dots X_l$ )
      $k = i + 1$
    case ($X_i \gtrdot X_{i+1}$): (найдем правый конец $X_l = X_i$ основы $X_k \dots X_l$)
      $l = i$
    (else: $X_i \doteq X_{i+1}$)
    $i = +1$ (двигаемся дальше, пока не дойдем до конца основы)
  if ($l = 0$) or ($\nexists A$: $A \to X_k \dots X_l$): (не найден правый конец основы или не существует соответствующего правила для основы)
    print('$w \notin L(G)$')   exit (выход из программы)
  else: (когда нашли основу, произвели свертку по соответствующему правилу $A \to X_k \dots X_l$)
    $\gamma = X_0 X_1 \dots X_{k-1} A X_{l+1} \dots X_n X_{n+1}$
```

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$
$\vdash \lessdot a \doteq S \lessdot a \lessdot \boxed{c} \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$
$\vdash \lessdot a \doteq S \lessdot a \lessdot \boxed{c} \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$
$\vdash \lessdot a \doteq S \lessdot a \doteq S \lessdot \boxed{c} \gtrdot b \gtrdot b \gtrdot \dashv$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot a \lessdot \boxed{c} \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot a \doteq S \lessdot \boxed{c} \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot b \gtrdot \dashv$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$$

$$\vdash \lessdot a \doteq S \lessdot a \lessdot \boxed{c} \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$$

$$\vdash \lessdot a \doteq S \lessdot a \doteq S \lessdot \boxed{c} \gtrdot b \gtrdot b \gtrdot \dashv$$

$$\vdash \lessdot a \doteq S \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot b \gtrdot \dashv$$

$$\vdash \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot \dashv$$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot a \lessdot \boxed{c} \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot a \doteq S \lessdot \boxed{c} \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot b \gtrdot \dashv$

$\vdash \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot \dashv$

$\vdash \lessdot S \gtrdot \dashv$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$\vdash <\cdot a <\cdot \boxed{c} \cdot> a <\cdot c \cdot> c \cdot> b \cdot> b \cdot> \dashv$

$\vdash <\cdot a \doteq S <\cdot a <\cdot \boxed{c} \cdot> c \cdot> b \cdot> b \cdot> \dashv$

$\vdash <\cdot a \doteq S <\cdot a \doteq S <\cdot \boxed{c} \cdot> b \cdot> b \cdot> \dashv$

$\vdash <\cdot a \doteq S <\cdot \boxed{a \doteq S \doteq S \doteq b} \cdot> b \cdot> \dashv$

$\vdash <\cdot \boxed{a \doteq S \doteq S \doteq b} \cdot> \dashv$

$\vdash <\cdot S \cdot> \dashv$

И восстановим соответствующий правосторонний вывод для грамматики

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$$\vdash <\cdot a <\cdot \boxed{c} \cdot> a <\cdot c \cdot> c \cdot> b \cdot> b \cdot> \dashv$$
$$\vdash <\cdot a \doteq S <\cdot a <\cdot \boxed{c} \cdot> c \cdot> b \cdot> b \cdot> \dashv$$
$$\vdash <\cdot a \doteq S <\cdot a \doteq S <\cdot \boxed{c} \cdot> b \cdot> b \cdot> \dashv$$
$$\vdash <\cdot a \doteq S <\cdot \boxed{a \doteq S \doteq S \doteq b} \cdot> b \cdot> \dashv$$
$$\vdash <\cdot \boxed{a \doteq S \doteq S \doteq b} \cdot> \dashv$$
$$\vdash <\cdot S \cdot> \dashv$$

И восстановим соответствующий правосторонний вывод для грамматики

$$S \Rightarrow \underline{aSSb} \Rightarrow aS\underline{aSSb}b \Rightarrow aSaS\underline{c}bb \Rightarrow aSa\underline{c}cbb \Rightarrow aca\underline{c}cbb = w.$$

Пример 3 восходящего анализа для ПП грамматики

Пример 3. Рассмотрим восходящий анализ для слова $w = acaccbb$ для ПП грамматики из примера 2, опираясь на теорему 3, помня, что мы выбираем самую левую основу, основываясь на алгоритме восходящего анализа для ПП-грамматик.

$\vdash \lessdot a \lessdot \boxed{c} \gtrdot a \lessdot c \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot a \lessdot \boxed{c} \gtrdot c \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot a \doteq S \lessdot \boxed{c} \gtrdot b \gtrdot b \gtrdot \dashv$

$\vdash \lessdot a \doteq S \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot b \gtrdot \dashv$

$\vdash \lessdot \boxed{a \doteq S \doteq S \doteq b} \gtrdot \dashv$

$\vdash \lessdot S \gtrdot \dashv$

И восстановим соответствующий правосторонний вывод для грамматики

$S \Rightarrow \underline{aSSb} \Rightarrow aS\underline{aSSb}b \Rightarrow aSaS\underline{c}bb \Rightarrow aSa\underline{c}cbb \Rightarrow aca\underline{c}cbb = w.$

Пример 4 восходящего анализа для ПП грамматики

Пример 4. Теперь рассмотрим восходящий анализ для слова $u = acb$ для той же грамматики из примера 2.

Пример 4 восходящего анализа для ПП грамматики

Пример 4. Теперь рассмотрим восходящий анализ для слова $u = acb$ для той же грамматики из примера 2.

$\vdash < \cdot a < \cdot \boxed{c} \cdot > b \cdot > \dashv$

Пример 4 восходящего анализа для ПП грамматики

Пример 4. Теперь рассмотрим восходящий анализ для слова $u = acb$ для той же грамматики из примера 2.

$$\vdash \lessdot a \lessdot \boxed{c} \gtrdot b \gtrdot \dashv$$

$$\vdash \lessdot \underline{a \doteq S} \gtrdot b \gtrdot \dashv$$

Пример 4 восходящего анализа для ПП грамматики

Пример 4. Теперь рассмотрим восходящий анализ для слова $u = acb$ для той же грамматики из примера 2.

$$\vdash \lessdot a \lessdot \boxed{c} \gtrdot b \gtrdot \dashv$$

$$\vdash \lessdot \underline{a \doteq S} \gtrdot b \gtrdot \dashv$$

Подслово aS не является основой, следовательно, слово $u = acb$ не принадлежит языку, порождаемому грамматикой.