Лингвистические основы информатики
Лекция 17
Восходящий анализ. Отношения предшествования. Операторные грамматики

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

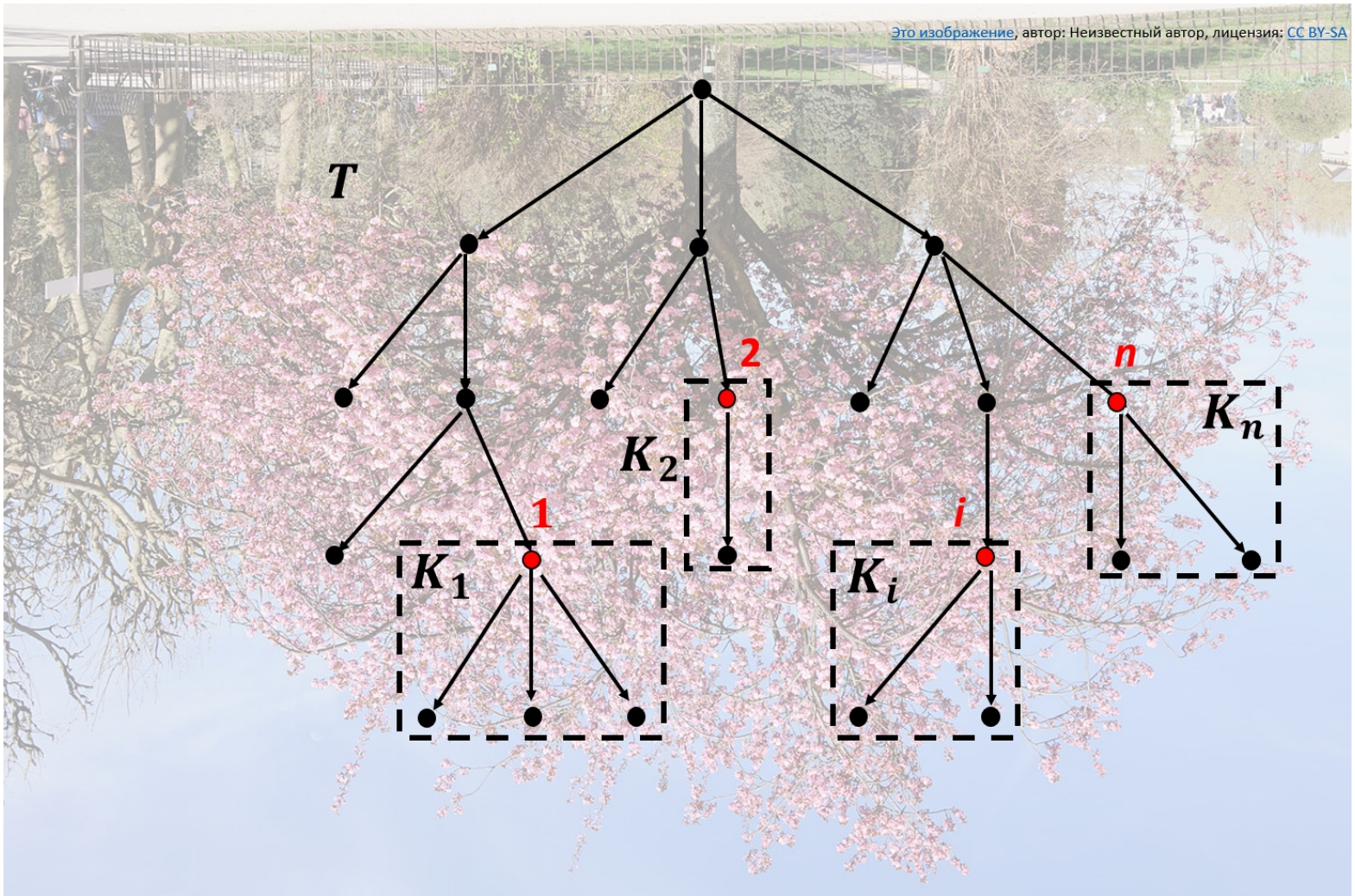


Рис. 1

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

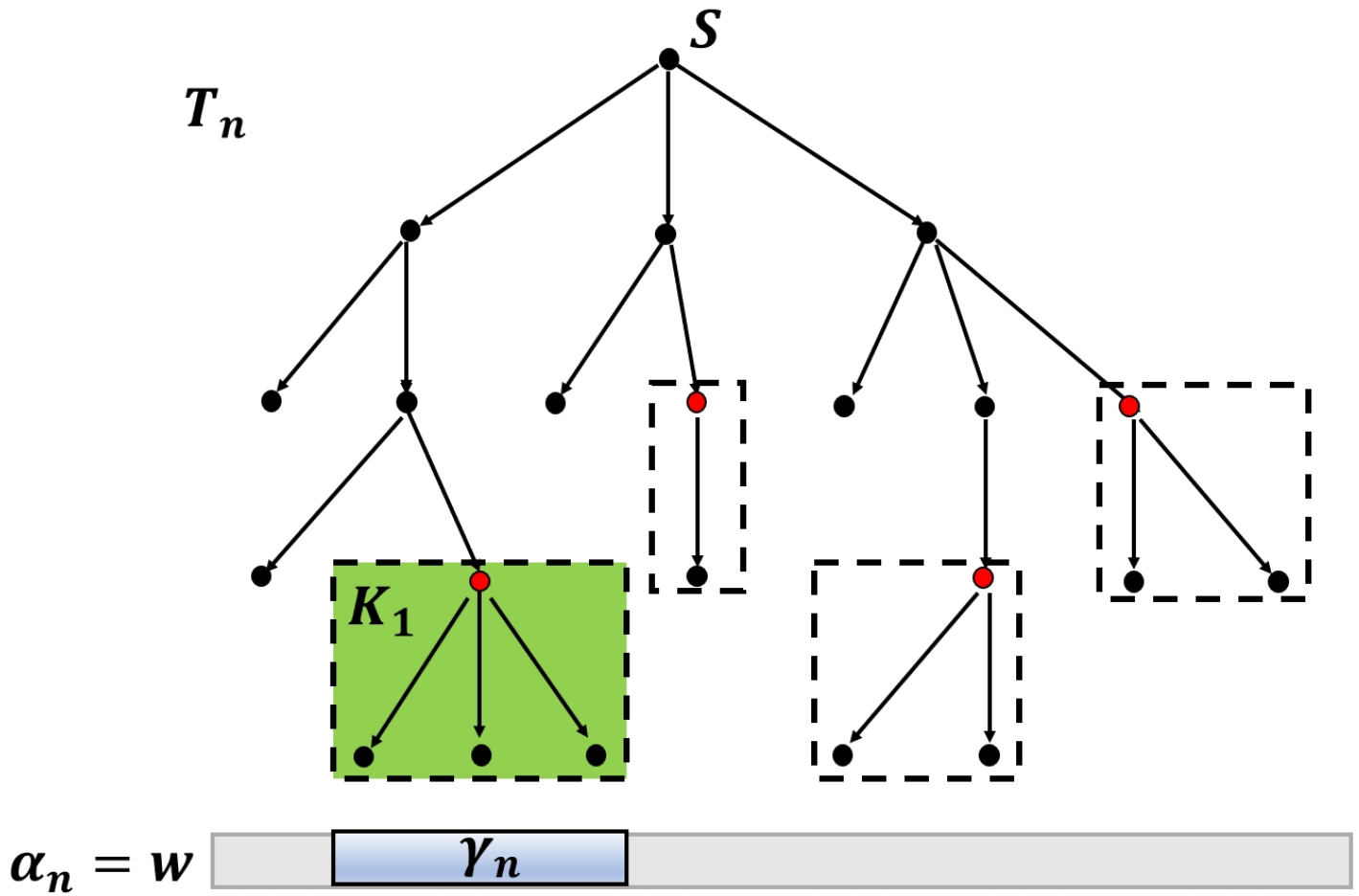


Рис. 2

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

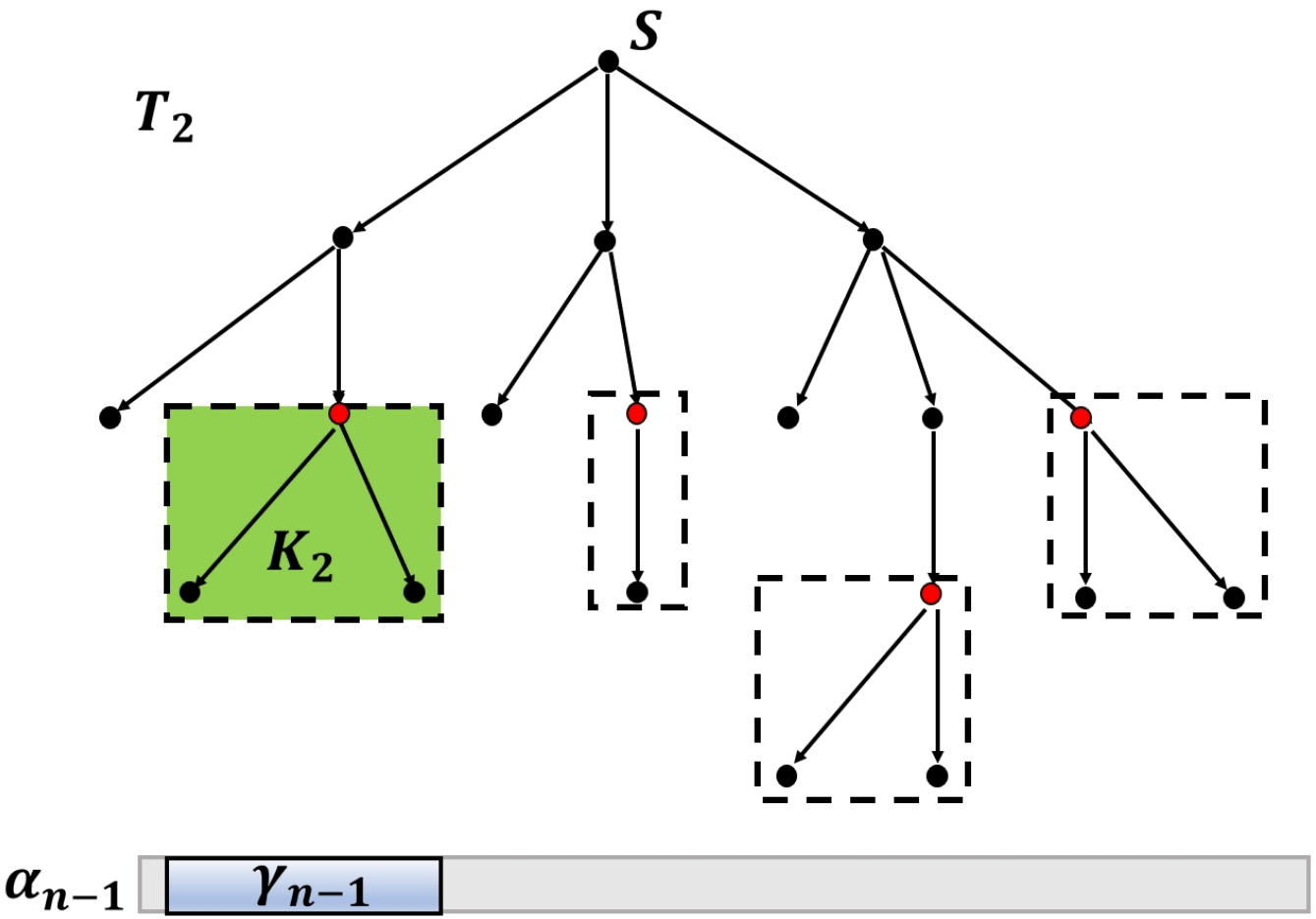


Рис. 3

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

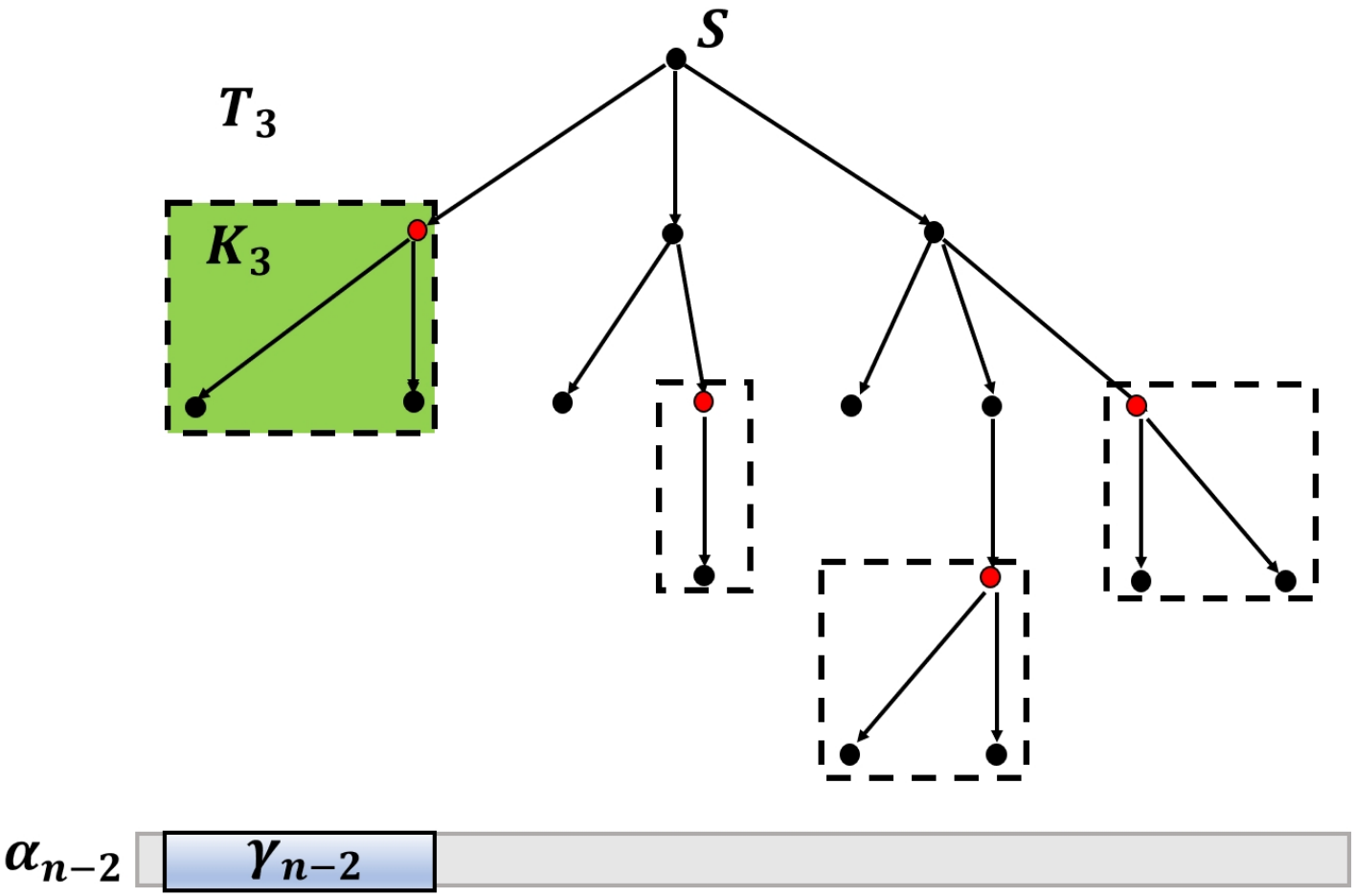


Рис. 4

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

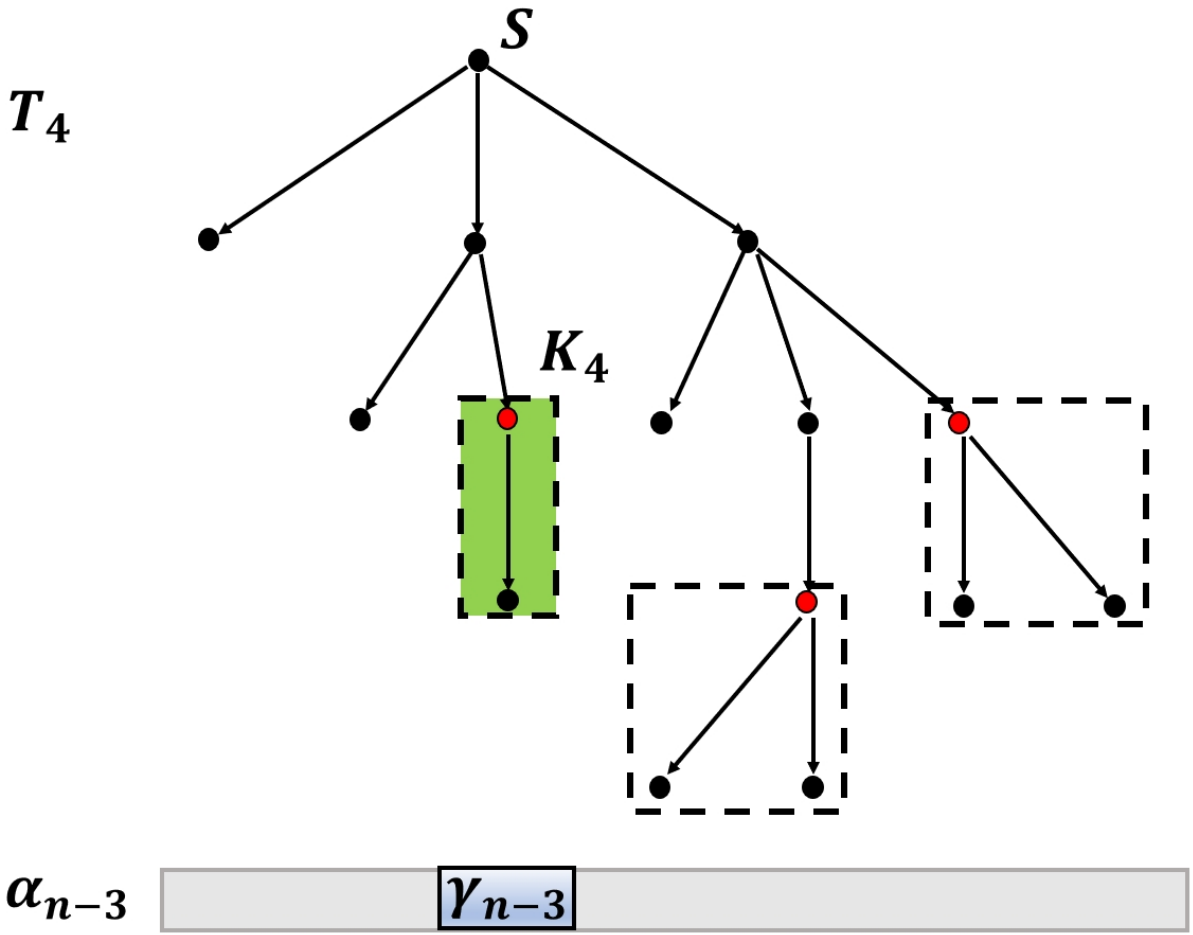


Рис. 5

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

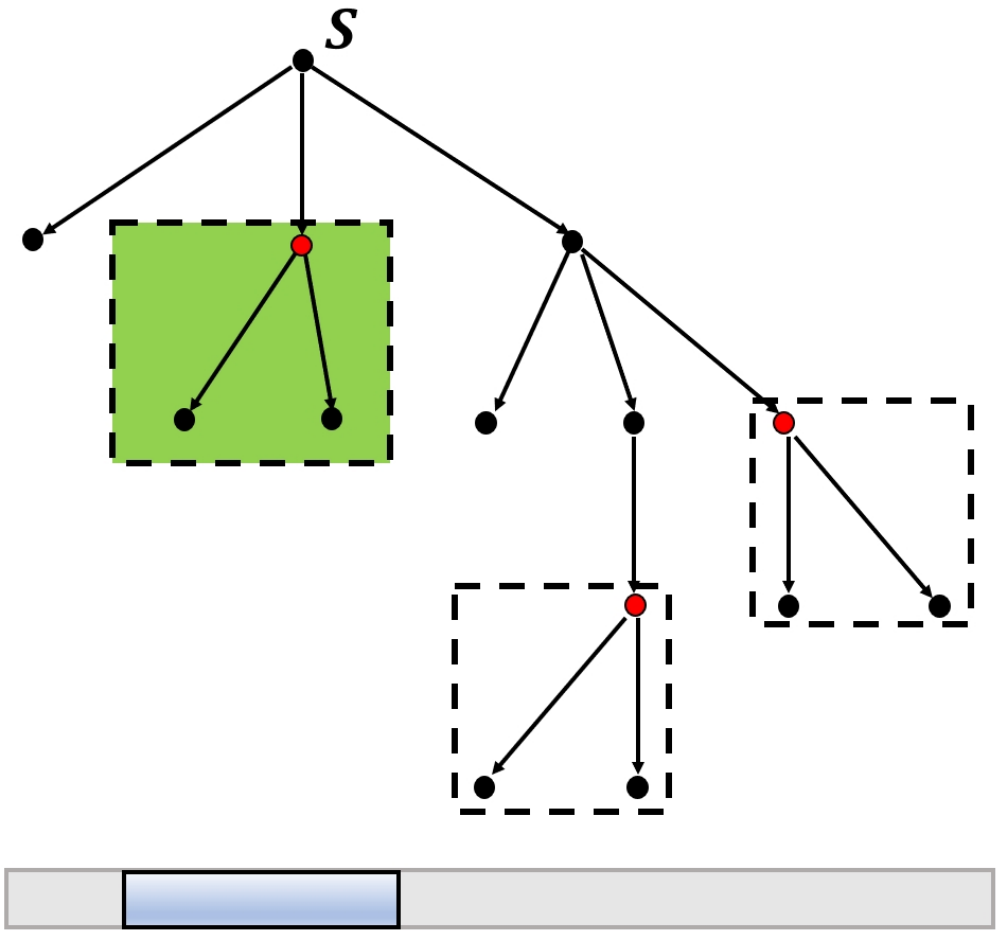


Рис. 6

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

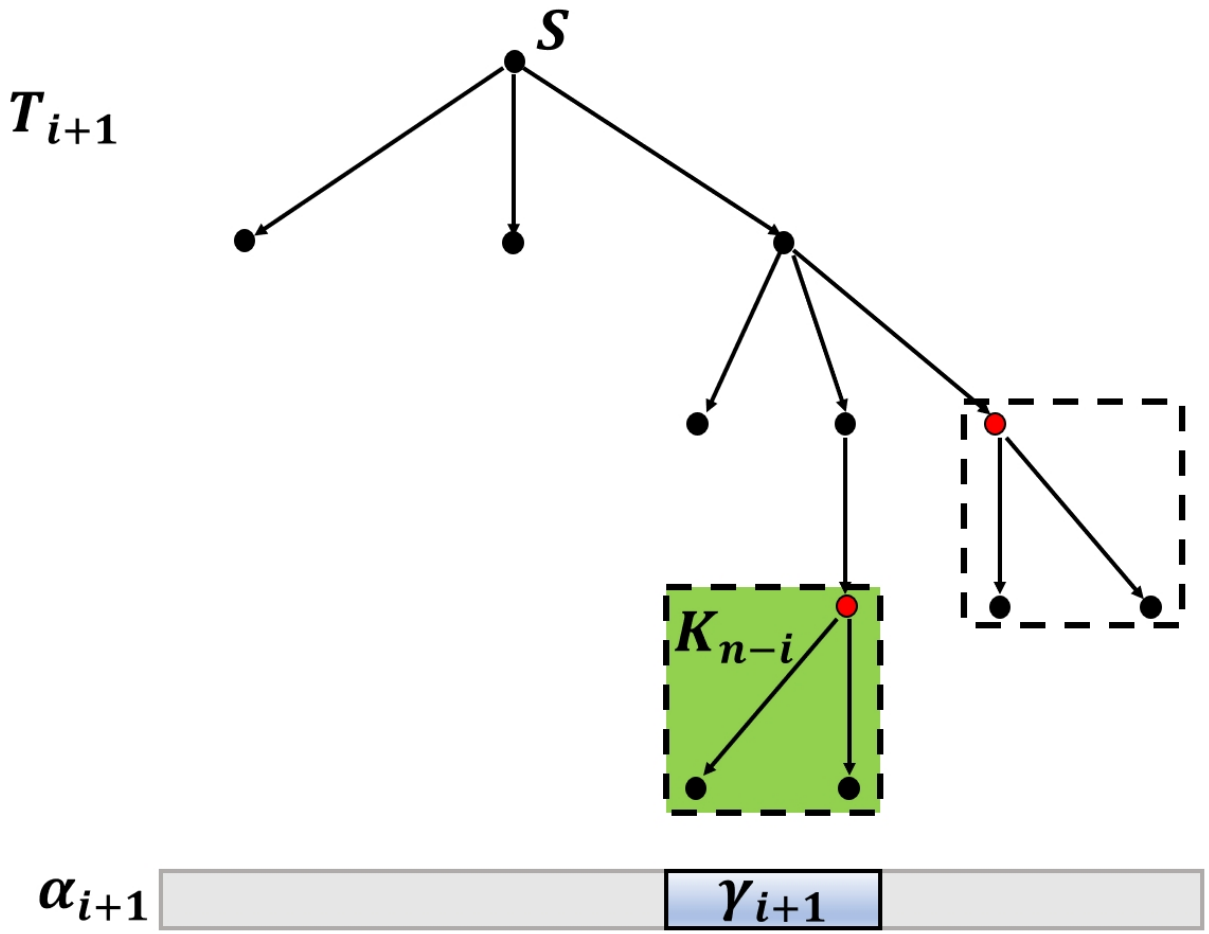


Рис. 7

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

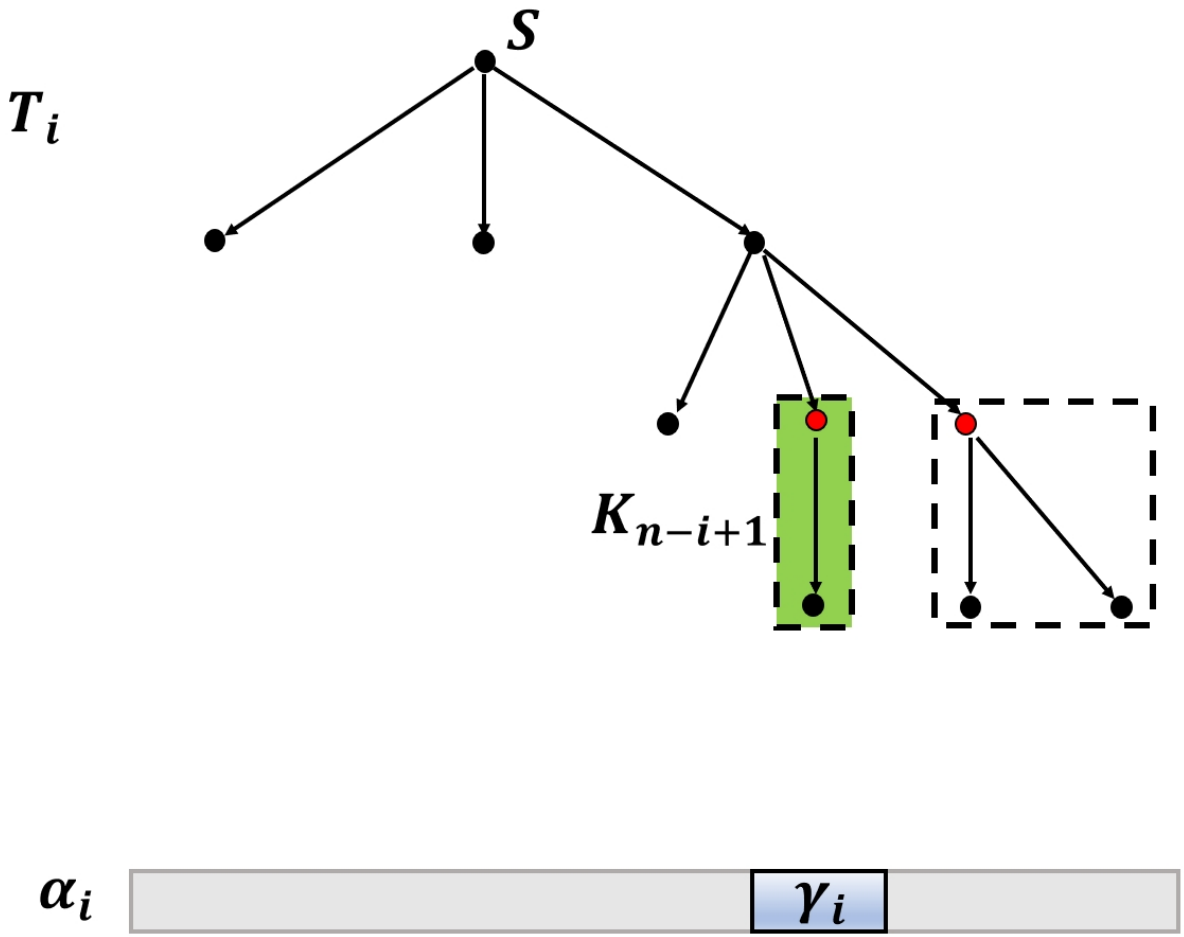


Рис. 8

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

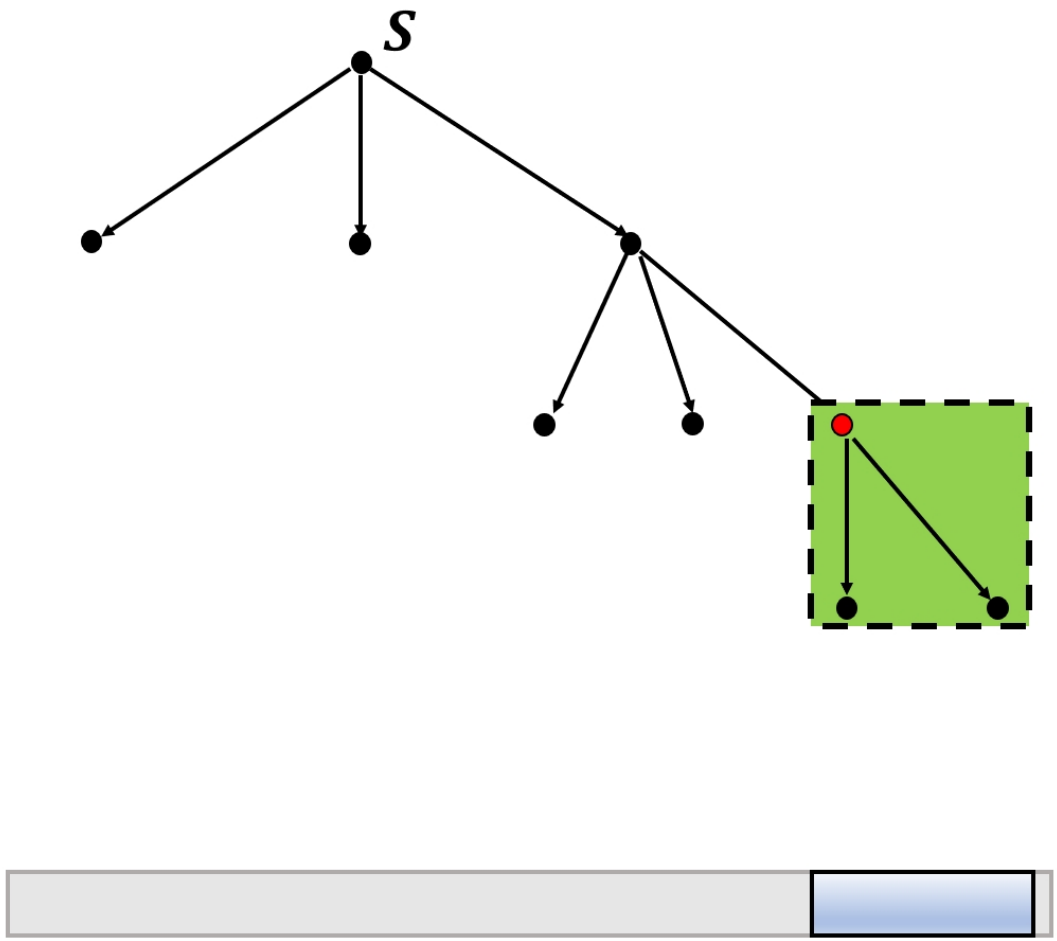


Рис. 9

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

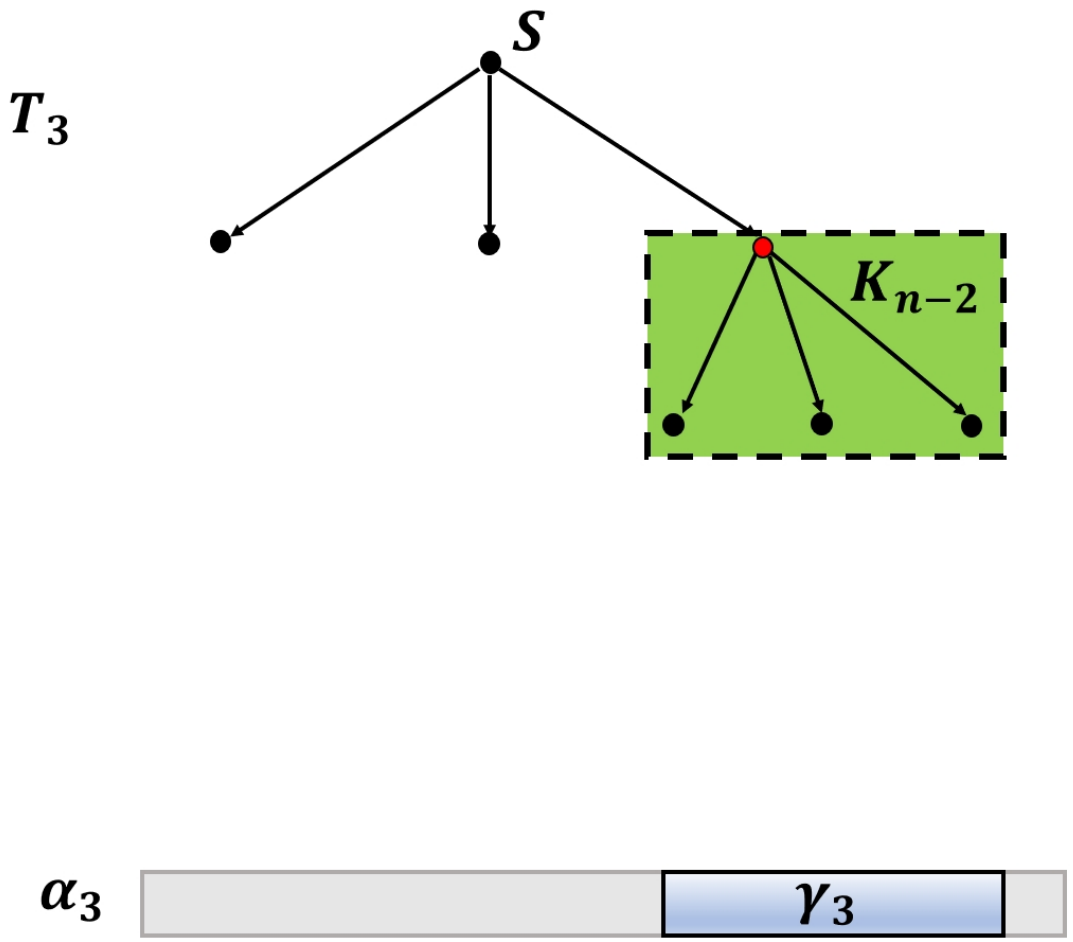


Рис. 10

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

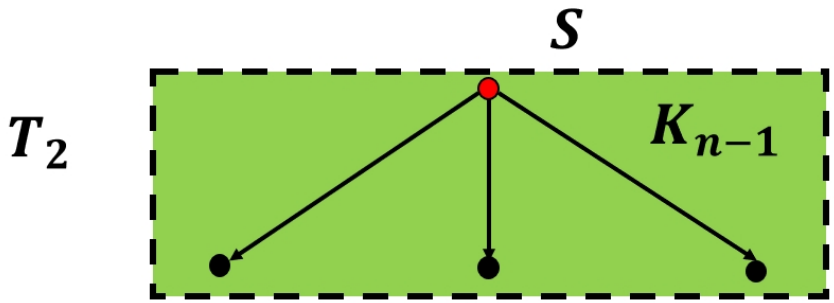


Рис. 11

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.

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

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
  $S \Rightarrow \underline{aFSd} \Rightarrow aF\underline{cd} \Rightarrow a\underline{Fb}cd \Rightarrow a\underline{Fb}bcd \Rightarrow a\underline{b}bbcd$

Восходящий анализ. Основные понятия. Пример 1

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

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

- 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

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

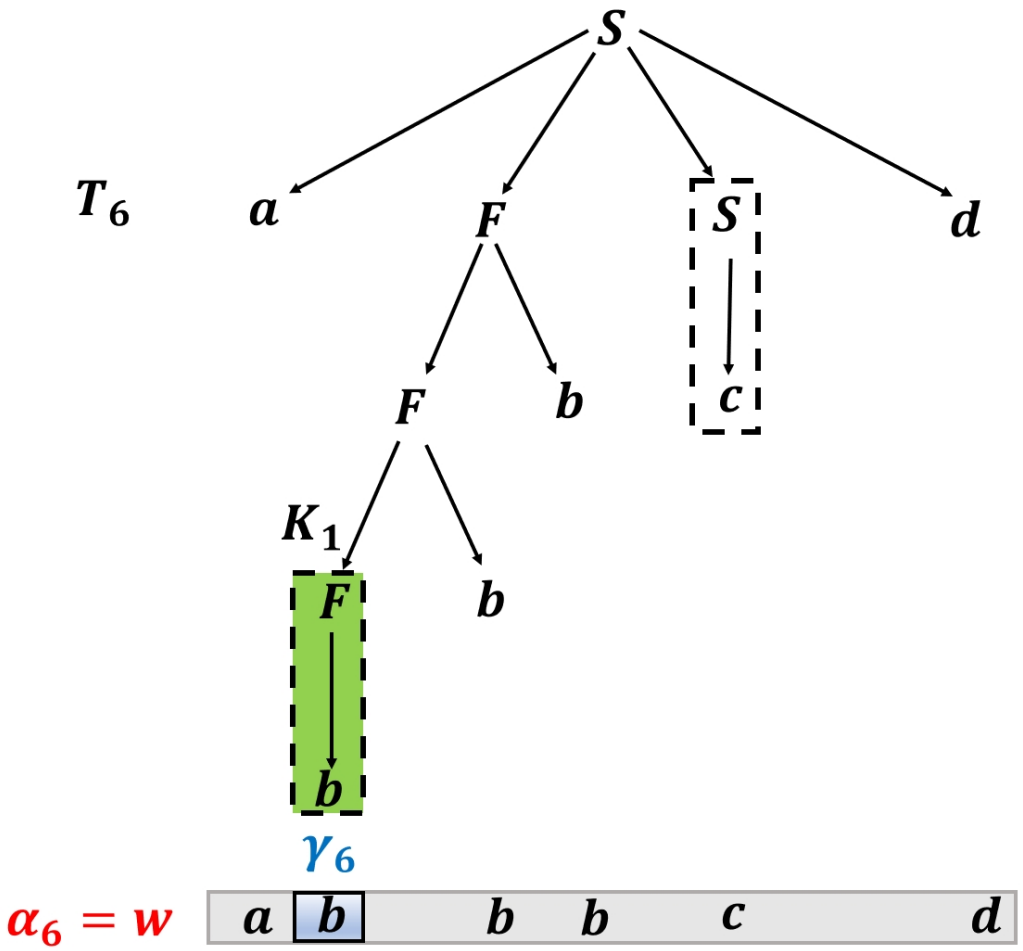
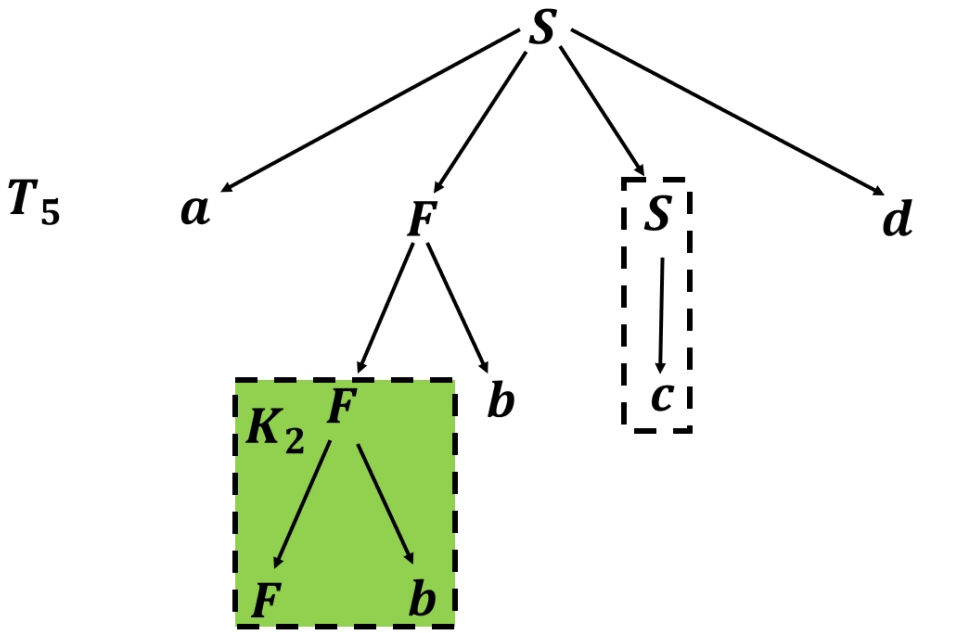


Рис. 13

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация



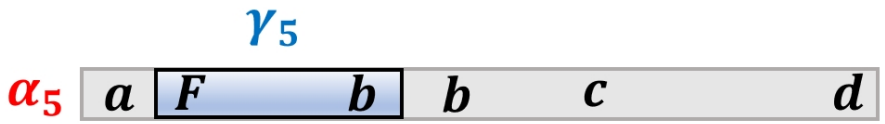
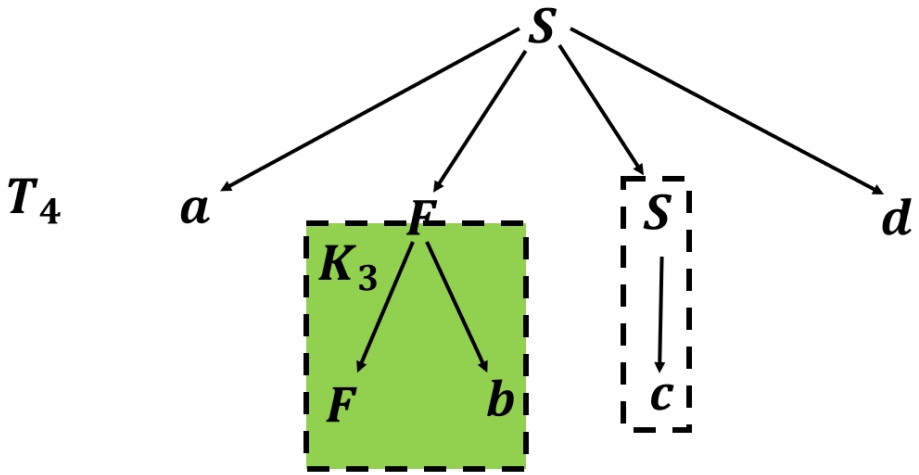


Рис. 14

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация



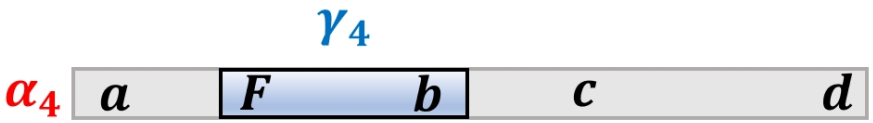


Рис. 15

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

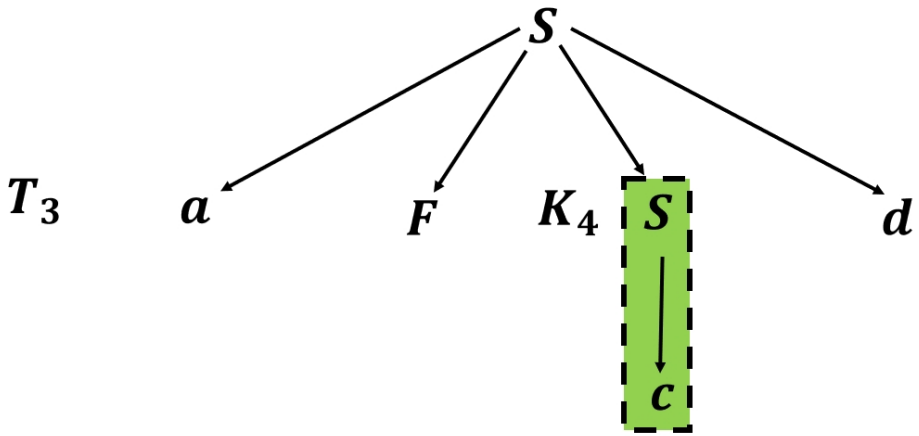


Рис. 16

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

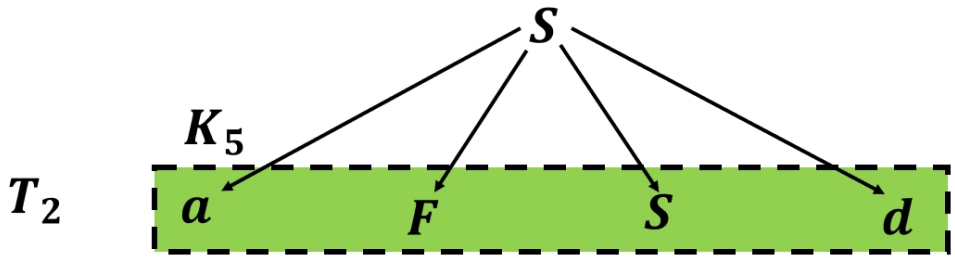


Рис. 17

Восходящий анализ. Основные понятия. Пример 1. Иллюстрация

T_1

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-j+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-j+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-j+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-j+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-j+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-j+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-j+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

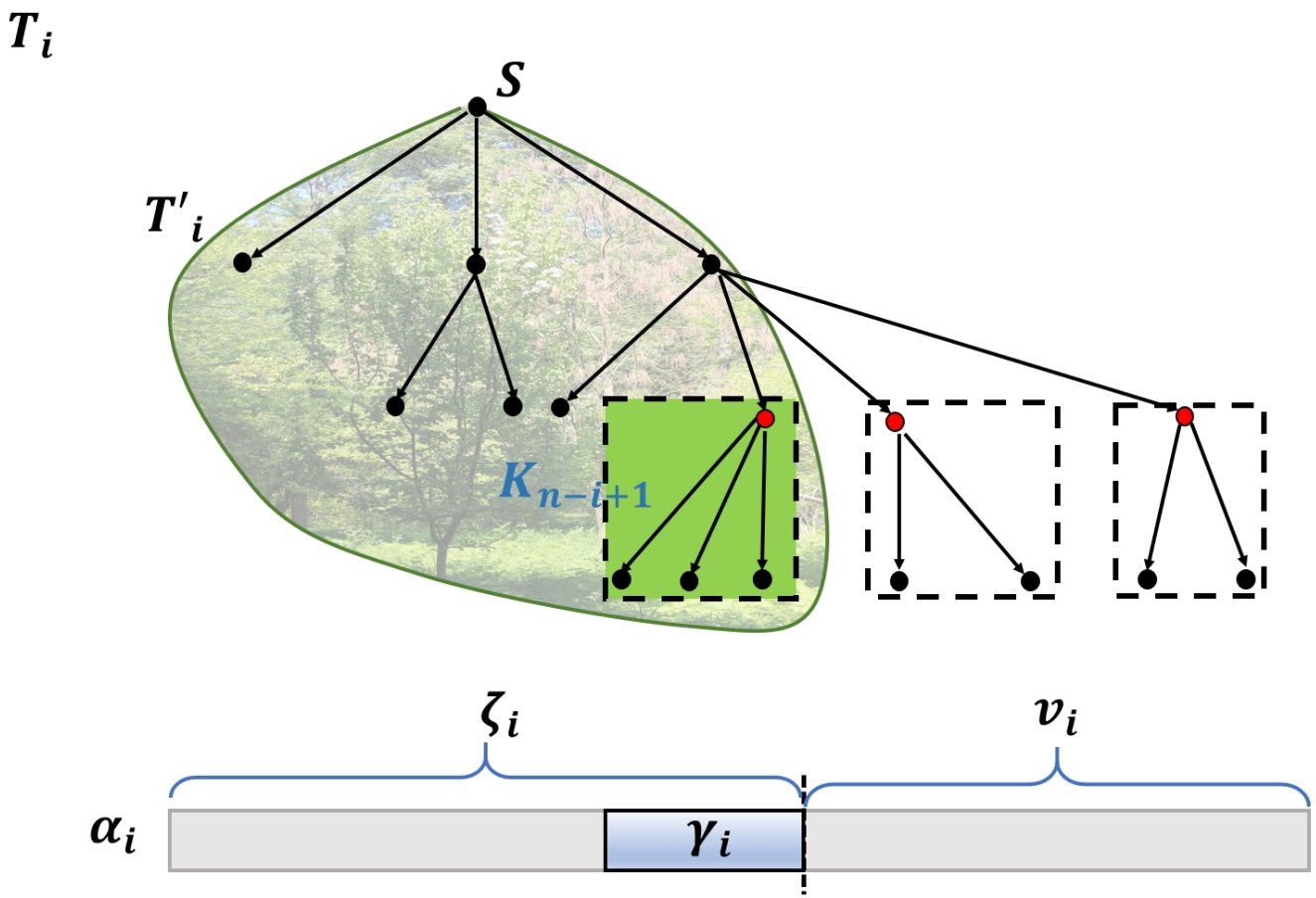


Рис. 19

Общая идея восходящего анализа. Обоснование. Иллюстрация

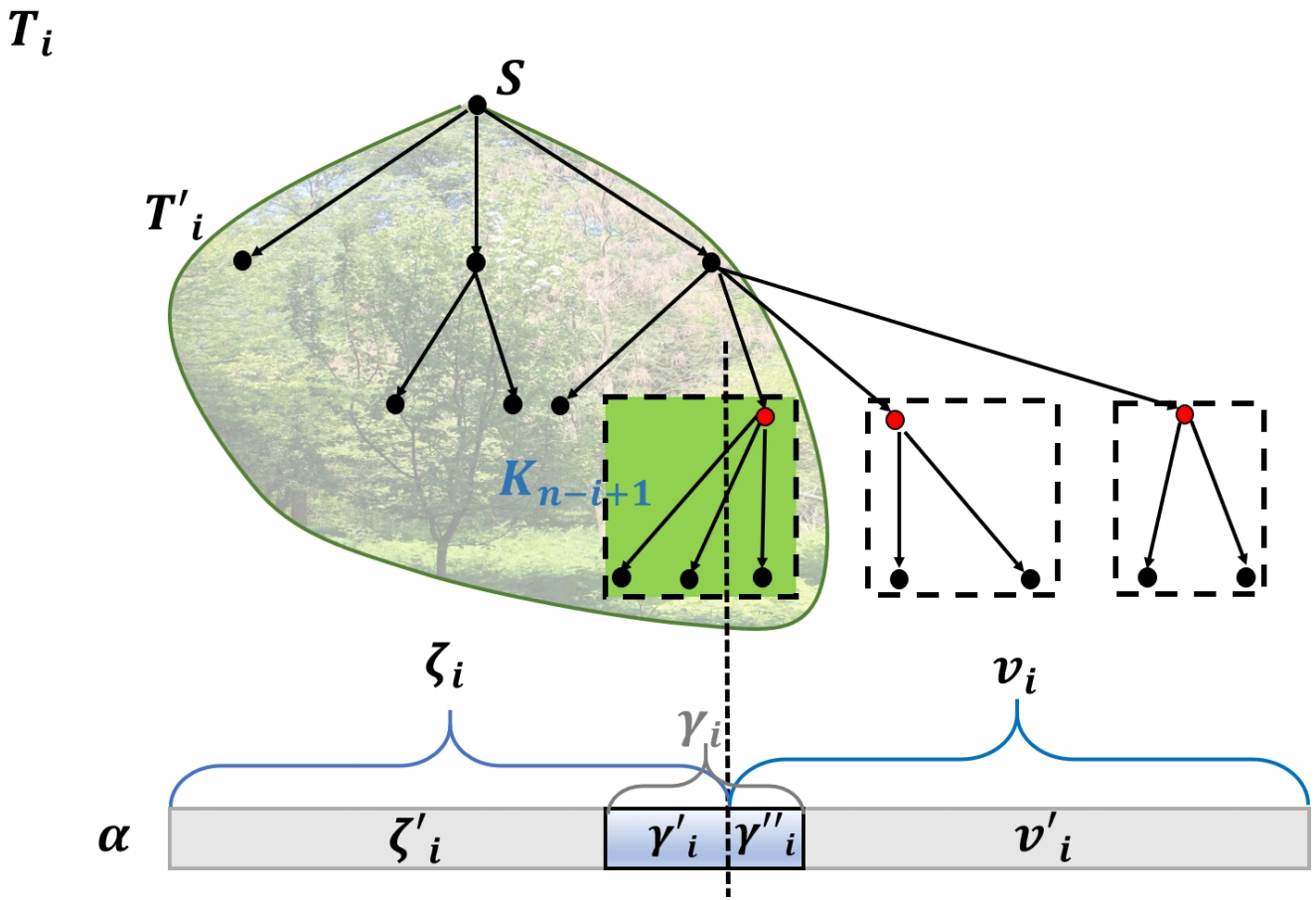


Рис. 20

Общая идея восходящего анализа. Обоснование. Иллюстрация

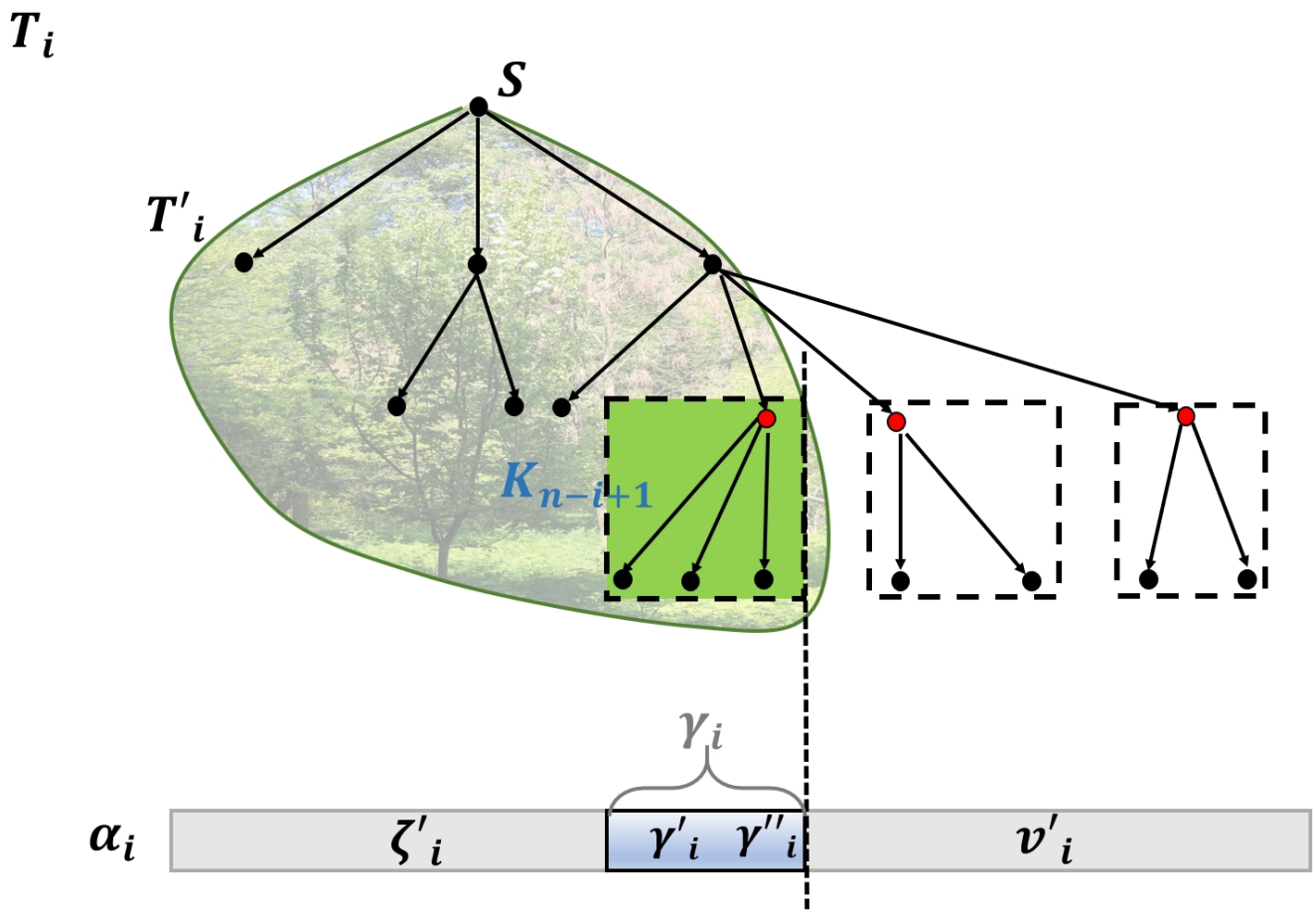


Рис. 21

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	∇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	∇	$\diamond$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	∇aFSd	abbbcd$\diamond\dashv$
12	∇S	abbbcd$\diamond\dashv$

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	∇	$\diamond$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	∇aFSd	abbbcd$\diamond\dashv$
12	∇S	abbbcd$\diamond\dashv$

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 $\alpha_4 = aF\underline{bc}d$ активным префиксом являются следующие слова $\varepsilon, a, aF, aFb, aFbc, aFbcd$.

Активный префикс

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

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

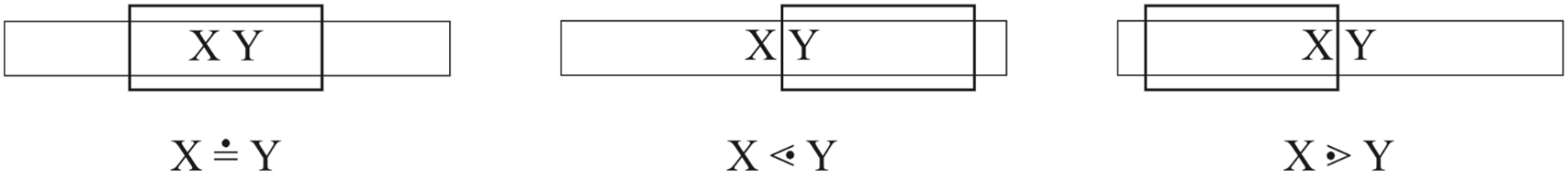


Рис. 22

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

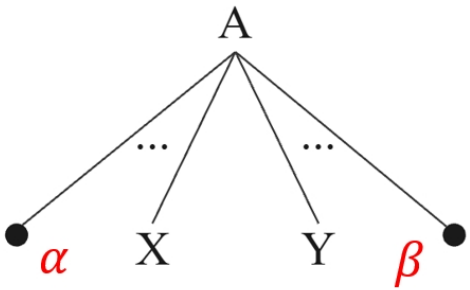
Для любых $X, Y \in \Sigma \cup \Gamma$ (см.рис.24).

1. $X \doteq Y$ т. и т.т.к. $(A \to \alpha XY\beta) \in P$;
2. $X <\cdot Y$ т. и т.т.к. $(A \to \alpha XZ\beta) \in P$ и $Z \Rightarrow^+ Y\gamma$;
3. $X \cdot > Y$ 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 $Z_1 \Rightarrow^+ \gamma_1 X$, $Z_2 \Rightarrow^* Y\gamma_2$.

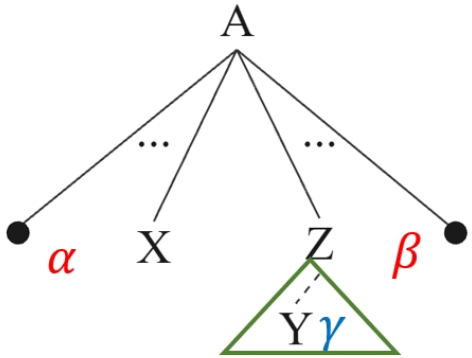
Замечание 2 (упр.)

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

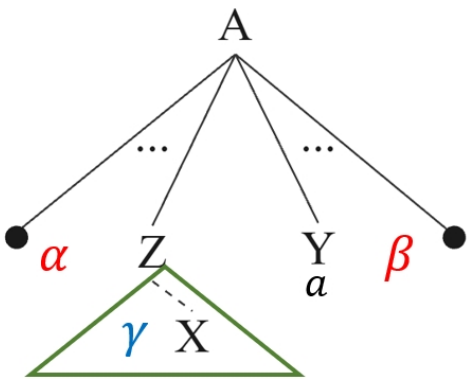
Теорема об отношениях предшествования. Иллюстрация



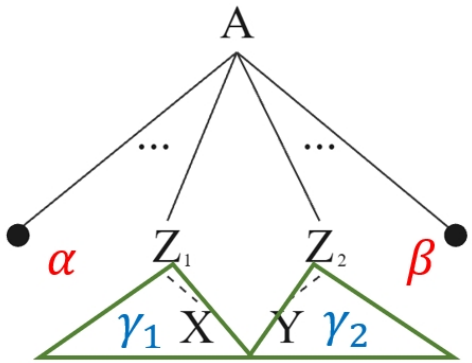
а) $X \doteq Y$



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 $X <\cdot Y$, 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 $X \cdot > Y$, что противоречиво.

Теорема об отношениях предшествования

Доказательство.

- (1) следует из определения отношения $X \doteq Y$ (см. рис.23а).
- (2) Пусть $X <\cdot Y$, 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 $X \cdot > Y$, что противоречиво.
- Итак, X — обязательно сын A, причем у X еще есть некоторый правый брат Z, т.е. $(A \to \alpha X Z \beta) \in P$. Так как не выполняется $X \doteq Y$, то $Z \neq Y$, но $Z \Rightarrow^{+} Y\gamma$, поскольку $X <\cdot Y$.

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 $X \cdot > Y$. 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)
  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. $X <\cdot Y$ 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. $X <\cdot Y$ 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. $X \cdot > Y$ 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. $X <\cdot Y$ 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. $X \cdot > Y$ 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. $\vdash <\cdot X$ 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. $X <\cdot Y$ 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. $X \cdot > Y$ 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. $\vdash <\cdot X$ т. и т.т.к. $X \in FIRST'(S) \cup \{S\}$.
5. $X \cdot > \dashv$ т. и т.т.к. $X \in LAST'(S) \cup \{S\}$.

Определение множества FIRST'. Иллюстрация

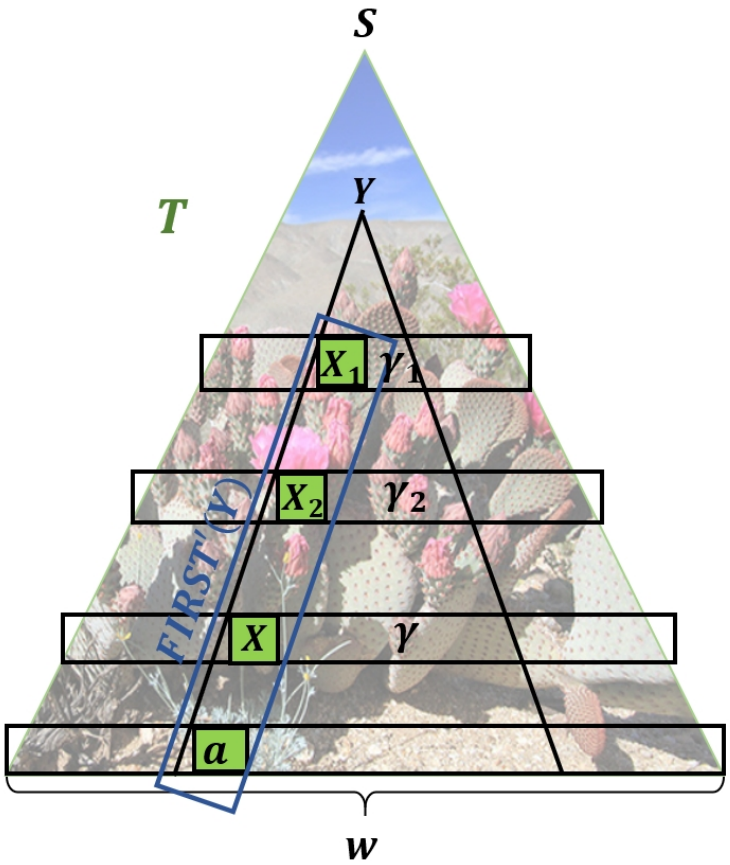


Рис. 24

Определение множества LAST'. Иллюстрация

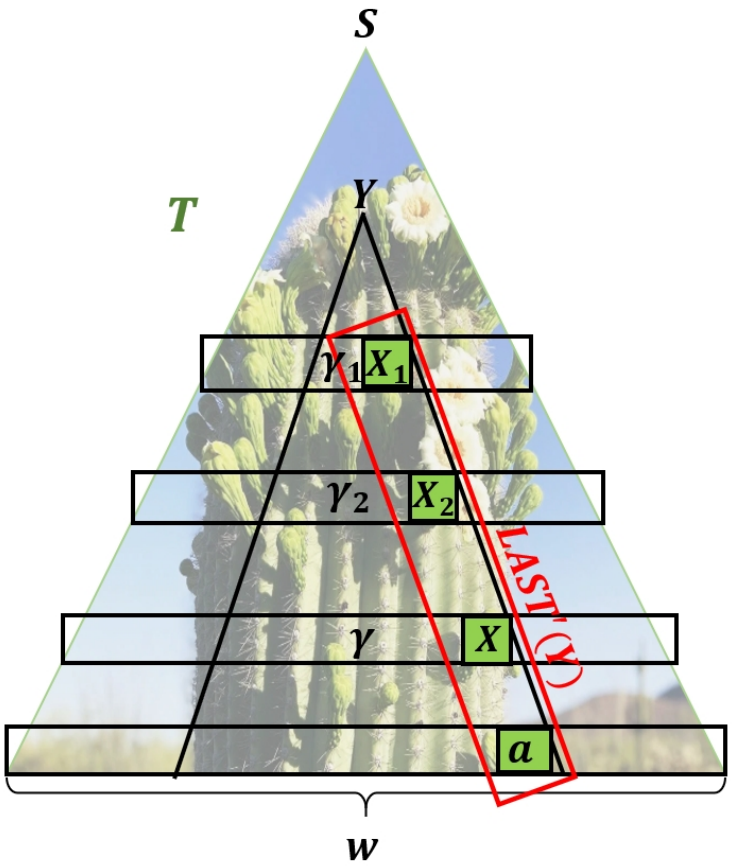


Рис. 25

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

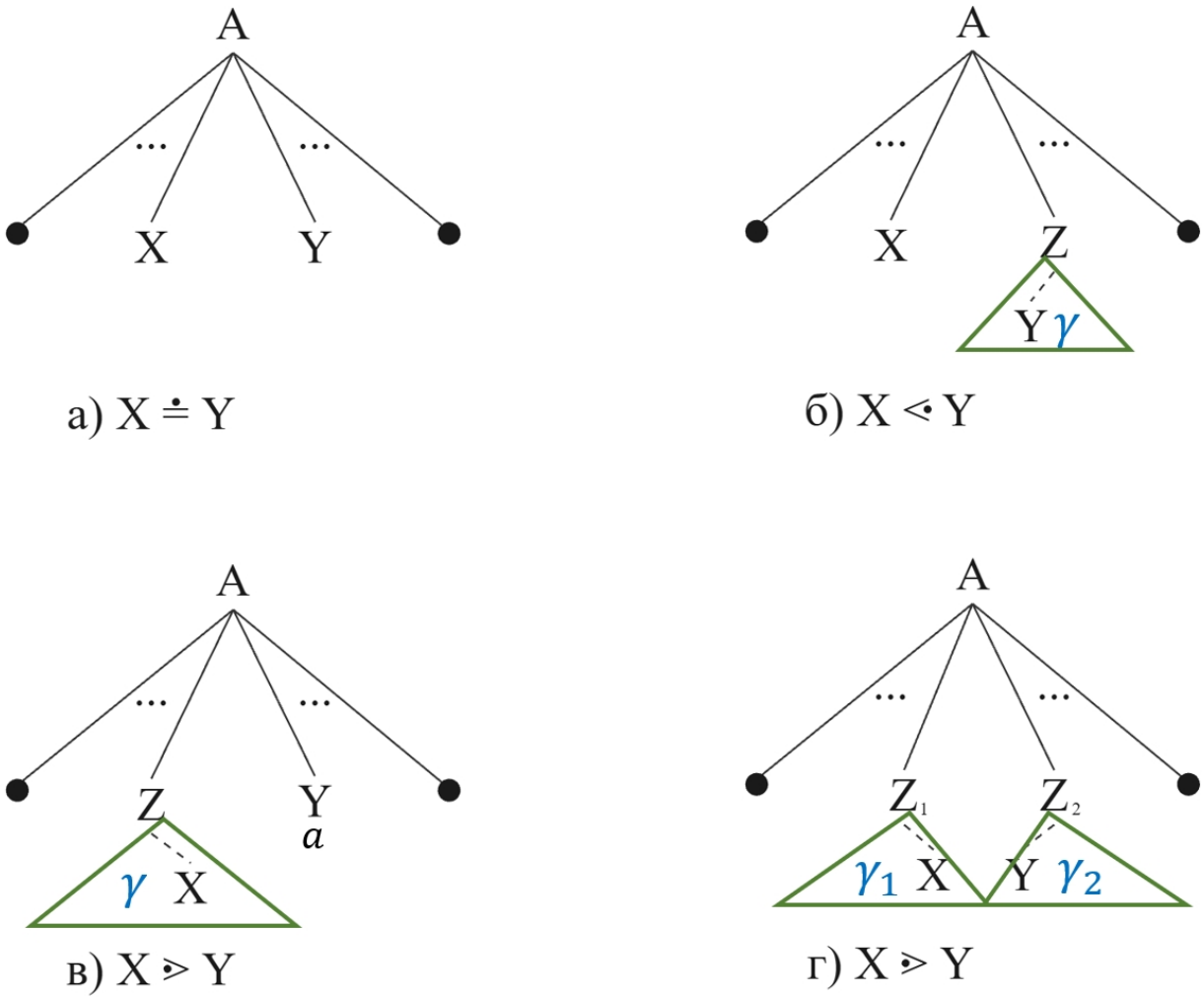
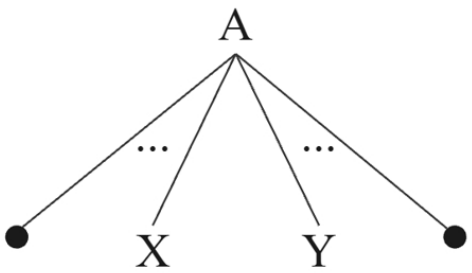
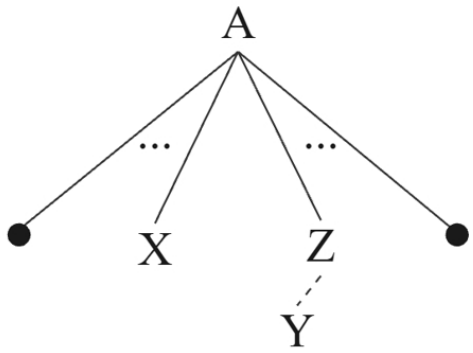


Рис. 24

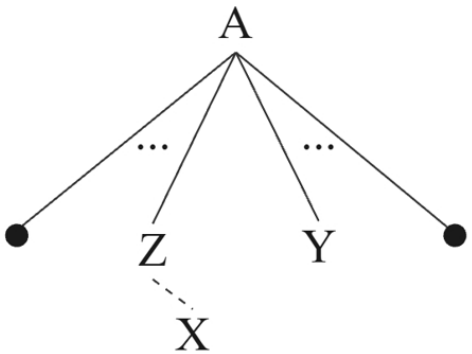
Теорема 2 об отношениях предшествования через *FIRST′* и *LAST′* Иллюстрация



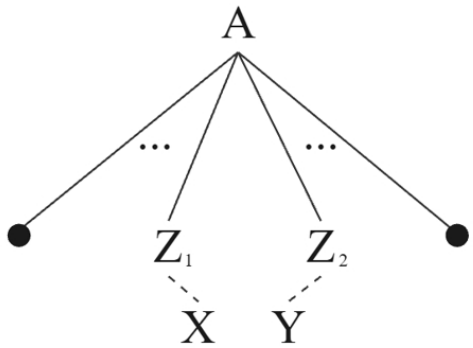
а) $X \doteq Y$



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 $(A \to X_1 X_2 \dots X_n) \in P$:

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 $FIRST(Y)$ для всех терминалов и нетерминалов Y

Алгоритм построения множеств LAST' 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 LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.

Выход: Набор множеств $FIRST(Y)$ для всех терминалов и нетерминалов Y

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

Алгоритм построения множеств LAST' для нетерминалов

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

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

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.

Выход: Набор множеств $FIRST(Y)$ для всех терминалов и нетерминалов Y

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

Алгоритм построения множеств LAST' для нетерминалов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FIRST(Y)$ 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 $FIRST(Y)$ 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 $(A \to X_1 X_2 \dots X_n) \in P$:

Алгоритм построения множеств LAST' для нетерминалов

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

- for $a \in \Sigma$:
  $LAST'(a) = \{a\}$
- 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 $S \rightarrow aSSb \mid c$

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

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

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

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

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

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

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

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

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

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

Восходящий анализ. Основные понятия. Пример 2 (продолжение)

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

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.

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)

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

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. $FIRST'(S) \cup \{S\} = \{S, a, c\}$.
- $S \gtrdot \dashv$, $b \gtrdot \dashv$, $c \gtrdot \dashv$, 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

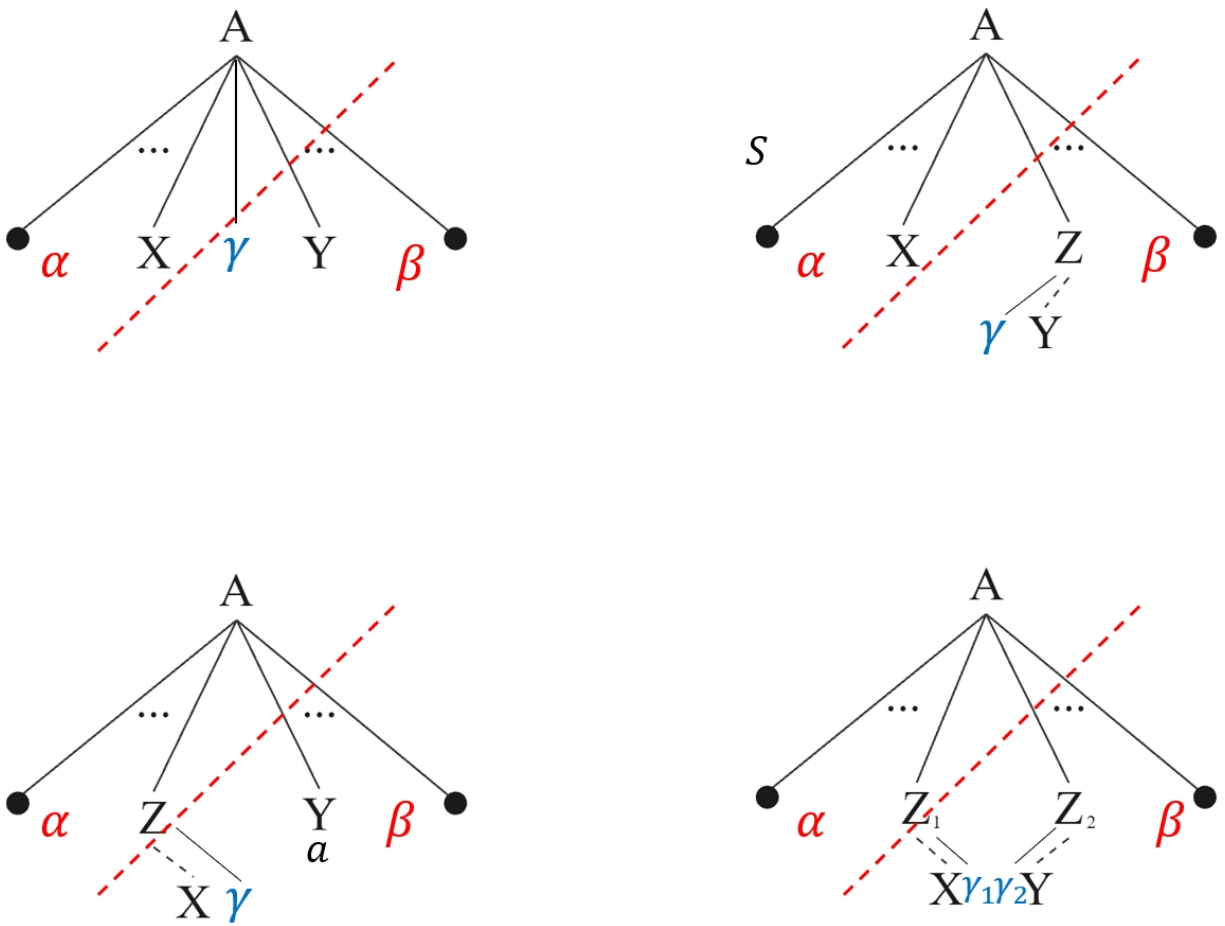


Рис. 26

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 $\vdash \gamma \dashv = X_0X_1...X_nX_{n+1}$. 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 $X_rX_{r+1}\dots X_{s-1}X_s \neq X_kX_{k+1}\dots X_{l-1}X_l$.

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 $X_l \cdot > X_{l+1}$, $1 \le k \le l \le n$, $X_{k-1} < \cdot X_k \doteq X_{k+1} \doteq \dots \doteq X_{l-1} \doteq X_l \cdot > X_{l+1}$.

- **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 $X_rX_{r+1}\dots X_{s-1}X_s \neq X_kX_{k+1}\dots X_{l-1}X_l$.
- 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 $X_rX_{r+1}\dots X_{s-1}X_s \neq X_kX_{k+1}\dots X_{l-1}X_l$.
- Тогда по определению отношений предшествования $X_{r-1} < \cdot X_r \doteq X_{r+1} \doteq \dots = X_{s-1} \doteq X_s \cdot > X_{s+1}$.
- 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 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 $X_0 <\cdot X_1$ 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 $X_0 <\cdot X_1$ 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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ

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

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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')

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 $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i <\cdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ 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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i <\cdot X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ основы $X_k \dots X_l$)
      $k = i + 1$

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 $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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 $X_k \dots X_l$)
      $l = i$

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 $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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: ($X_i \doteq X_{i+1}$)

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 $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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)

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 $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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('$w \notin L(G)$')

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 $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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('$w \notin L(G)$')
  else: (когда нашли основу, произвели свертку по соответствующему правилу $A \to X_k \dots X_l$)

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 $X_0 <\cdot X_1$ либо $X_0 \circ X_1$)
(γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$
  print(γ)
(γ — 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 $X_k \dots X_l$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    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('$w \notin L(G)$')
  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.

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

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.

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

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

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.

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

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

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

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.

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

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

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

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

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.

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

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

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

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

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

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.

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

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

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

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

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

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

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

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

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.

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

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

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

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

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

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

И восстановим соответствующий правосторонний вывод для грамматики

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

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.

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

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

Пример 4 восходящего анализа для ПП грамматики

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

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

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

Грамматика слабого предшествования

Определение

Грамматика называется грамматикой слабого предшествования (СП), если

1. не существует символов X и Y таких, что ($X \lessdot Y$ или $X \doteq Y$) и $X \gtrdot Y$;
2. не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X \lessdot B$ или $X \doteq Y$.

Грамматика слабого предшествования

Определение

Грамматика называется грамматикой слабого предшествования (СП), если

1. не существует символов X и Y таких, что ($X \lessdot Y$ или $X \doteq Y$) и $X \gtrdot Y$;
2. не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X \lessdot B$ или $X \doteq Y$.

Второе условие говорит от том, что никакая основа β, являющаяся суффиксом другой основы $\alpha X \beta$, не может быть свернута раньше основы $\alpha X \beta$.

Грамматика слабого предшествования

Определение

Грамматика называется грамматикой слабого предшествования (СП), если

1. не существует символов X и Y таких, что ($X \lessdot Y$ или $X \doteq Y$) и $X \gtrdot Y$;
2. не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X \lessdot B$ или $X \doteq Y$.

Второе условие говорит от том, что никакая основа β, являющаяся суффиксом другой основы $\alpha X \beta$, не может быть свернута раньше основы $\alpha X \beta$.

Теорема 4 (о связи ПП и СП грамматик)

Любая ПП-грамматика является СП-грамматикой.

Грамматика слабого предшествования

Доказательство. Пусть G — ПП-грамматика. Достаточно проверить (2)-е условие из определения СП-грамматики. Пусть ПП-грамматика G имеет правила вывода $A \to \alpha X \beta$ и $B \to \beta$. Так как G ε-свободна, $\beta \neq \varepsilon$, поэтому $\beta = Y\gamma$ для некоторого символа Y. Тогда $X \doteq Y$, поскольку правило вывода $A \to \alpha X \beta$ и $B \to \beta$ можно записать как $A \to \alpha X Y \gamma$.

Грамматика слабого предшествования

Доказательство. Пусть G — ПП-грамматика. Достаточно проверить (2)-е условие из определения СП-грамматики. Пусть ПП-грамматика G имеет правила вывода $A \to \alpha X \beta$ и $B \to \beta$. Так как G ε-свободна, $\beta \neq \varepsilon$, поэтому $\beta = Y\gamma$ для некоторого символа Y. Тогда $X \doteq Y$, поскольку правило вывода $A \to \alpha X \beta$ и $B \to \beta$ можно записать как $A \to \alpha X Y \gamma$.
Возможны два случая.

1. Пусть $X \doteq B$, тогда существует правило вывода $C \to \mu X B \nu$, поскольку $Y \in FIRST'(B)$, получаем $X \lessdot Y$, что противоречит $X \doteq Y$.

Грамматика слабого предшествования

Доказательство. Пусть G — ПП-грамматика. Достаточно проверить (2)-е условие из определения СП-грамматики. Пусть ПП-грамматика G имеет правила вывода $A \to \alpha X \beta$ и $B \to \beta$. Так как G ε-свободна, $\beta \neq \varepsilon$, поэтому $\beta = Y\gamma$ для некоторого символа Y. Тогда $X \doteq Y$, поскольку правило вывода $A \to \alpha X \beta$ и $B \to \beta$ можно записать как $A \to \alpha X Y \gamma$.
Возможны два случая.

1. Пусть $X \doteq B$, тогда существует правило вывода $C \to \mu X B \nu$, поскольку $Y \in FIRST'(B)$, получаем $X \lessdot Y$, что противоречит $X \doteq Y$.
2. Пусть $X \lessdot B$. Тогда найдется правило вывода $C \to \mu X Z \nu$ такое, что $B \in FIRST'(Z)$, но $Y \in FIRST'(B)$, поэтому $Y \in FIRST'(Z)$. Следовательно, $X \lessdot Y$, что снова противоречит тому, что $X \doteq Y$.

Пример 5 СП-грамматики

Итак, класс ПП-грамматик содержится в классе СП-грамматик. Причем это включение строгое, приведем пример 3 СП-грамматики, которая не является ПП-грамматикой.

Пример 5

- Дана грамматика $S \to S + A \mid A$, $A \to A * B \mid B$, $B \to x \mid (S)$,

Пример 5 СП-грамматики

Итак, класс ПП-грамматик содержится в классе СП-грамматик. Причем это включение строгое, приведем пример 3 СП-грамматики, которая не является ПП-грамматикой.

Пример 5

- Дана грамматика $S \to S + A \mid A$, $A \to A * B \mid B$, $B \to x \mid (S)$,
- Она, очевидно, приведенная, ε-свободная, однозначная.

Пример 5 СП-грамматики

Итак, класс ПП-грамматик содержится в классе СП-грамматик. Причем это включение строгое, приведем пример 3 СП-грамматики, которая не является ПП-грамматикой.

Пример 5

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

Пример 5 СП-грамматики

Итак, класс ПП-грамматик содержится в классе СП-грамматик. Причем это включение строгое, приведем пример 3 СП-грамматики, которая не является ПП-грамматикой.

Пример 5

- Дана грамматика $S \to S + A \mid A$, $A \to A * B \mid B$, $B \to x \mid (S)$,
- 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	$S, A, B, x, ($	$A, B, x,)$
A	$A, B, x, ($	$B, x,)$
B	$x, ($	$x,)$

Восходящий анализ. Основные понятия. Пример 5 (продолжение)

- $\boxed{S{+}A}$: $S \doteq +$
  $A, B, x,) \in LAST'(S) \Rightarrow A{\cdot}{>}+,\ B{\cdot}{>}+,\ x{\cdot}{>}+,\ ){\cdot}{>}+,$

Восходящий анализ. Основные понятия. Пример 5 (продолжение)

- $\boxed{S{+}A}$: $S \doteq +$
  $A, B, x,) \in LAST'(S) \Rightarrow A{\cdot}{>}+,\ B{\cdot}{>}+,\ x{\cdot}{>}+,\ ){\cdot}{>}+,$
- $\boxed{S{+}A}$: $+ \doteq A$
  $A, B, x, (\in FIRST'(A) \Rightarrow +{<}{\cdot}A,\ +{<}{\cdot}B,\ +{<}{\cdot}x,\ +{<}{\cdot}($

Восходящий анализ. Основные понятия. Пример 5 (продолжение)

- $\boxed{S{+}A}$: $S \doteq +$
  $A, B, x,) \in LAST'(S) \Rightarrow A \cdot{>} +,\ B \cdot{>} +,\ x \cdot{>} +,\ ) \cdot{>} +,$
- $\boxed{S{+}A}$: $+ \doteq A$
  $A, B, x, (\in FIRST'(A) \Rightarrow + {<}\cdot A,\ + {<}\cdot B,\ + {<}\cdot x,\ + {<}\cdot ($
- $\boxed{A{*}B}$: $A \doteq *$
  $B, x,) \in LAST'(A) \Rightarrow B \cdot{>} *,\ x \cdot{>} *,\ ) \cdot{>} *$

Восходящий анализ. Основные понятия. Пример 5 (продолжение)

- $\boxed{S{+}A}$: $S \doteq +$
  $A, B, x,) \in LAST'(S) \Rightarrow A \cdot{>}+,\ B \cdot{>}+,\ x \cdot{>}+,\ ) \cdot{>}+,$
- $\boxed{S{+}A}$: $+ \doteq A$
  $A, B, x, (\in FIRST'(A) \Rightarrow +{<}\cdot A,\ +{<}\cdot B,\ +{<}\cdot x,\ +{<}\cdot($
- $\boxed{A{*}B}$: $A \doteq *$
  $B, x,) \in LAST'(A) \Rightarrow B \cdot{>}*,\ x \cdot{>}*,\ ) \cdot{>}*$
- $\boxed{A{*}B}$: $* \doteq B$
  $x, (\in FIRST'(B) \Rightarrow *{<}\cdot x,\ *{<}\cdot($

Восходящий анализ. Основные понятия. Пример 5 (продолжение)

- $\boxed{S{+}A}$: $S \doteq +$
  $A, B, x,) \in LAST'(S) \Rightarrow A{\cdot}{>}+,\ B{\cdot}{>}+,\ x{\cdot}{>}+,\ ){\cdot}{>}+,$
- $\boxed{S{+}A}$: $+ \doteq A$
  $A, B, x, (\in FIRST'(A) \Rightarrow +{<}{\cdot}A,\ +{<}{\cdot}B,\ +{<}{\cdot}x,\ +{<}{\cdot}($
- $\boxed{A{*}B}$: $A \doteq *$
  $B, x,) \in LAST'(A) \Rightarrow B{\cdot}{>}*,\ x{\cdot}{>}*,\ ){\cdot}{>}*$
- $\boxed{A{*}B}$: $* \doteq B$
  $x, (\in FIRST'(B) \Rightarrow *{<}{\cdot}x,\ *{<}{\cdot}($
- $\boxed{(S)}$: $(\doteq S$
  $S, A, B, x, (\in FIRST'(S) \Rightarrow ({<}{\cdot}S,\ ({<}{\cdot}A,\ ({<}{\cdot}B,\ ({<}{\cdot}x,\ ({<}{\cdot}($,

Восходящий анализ. Основные понятия. Пример 5 (продолжение)

- $\boxed{S{+}A}$: $S \doteq +$
  $A, B, x,) \in LAST'(S) \Rightarrow A{\cdot}{>}+,\ B{\cdot}{>}+,\ x{\cdot}{>}+,\ ){\cdot}{>}+,$
- $\boxed{S{+}A}$: $+ \doteq A$
  $A, B, x, (\in FIRST'(A) \Rightarrow +{<}{\cdot}A,\ +{<}{\cdot}B,\ +{<}{\cdot}x,\ +{<}{\cdot}($
- $\boxed{A{*}B}$: $A \doteq *$
  $B, x,) \in LAST'(A) \Rightarrow B{\cdot}{>}*,\ x{\cdot}{>}*,\ ){\cdot}{>}*$
- $\boxed{A{*}B}$: $* \doteq B$
  $x, (\in FIRST'(B) \Rightarrow *{<}{\cdot}x,\ *{<}{\cdot}($
- $\boxed{(S)}$: $(\doteq S$
  $S, A, B, x, (\in FIRST'(S) \Rightarrow ({<}{\cdot}S,\ ({<}{\cdot}A,\ ({<}{\cdot}B,\ ({<}{\cdot}x,\ ({<}{\cdot}(,$
- $\boxed{(S)}$: $S \doteq)$
  $A, B, x,) \in LAST'(S) \Rightarrow A{\cdot}{>}),\ B{\cdot}{>}),\ x{\cdot}{>}),\ ){\cdot}{>})$

СП-грамматики Пример 5

Получим таблицу приоритетов:

	S	A	B	x	$+$	$*$	$($	$)$	$\dashv$
S					$\doteq$			$\doteq$	$\cdot>$
A					$\cdot>$	$\doteq$		$\cdot>$	$\cdot>$
B					$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$($		$<\cdot$	$<\cdot$	$<\cdot$			$<\cdot$		
x					$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$+$		$<\cdot, \doteq$	$<\cdot$	$<\cdot$			$<\cdot$		$\cdot>$
$*$			$\doteq$	$<\cdot$			$<\cdot$		
$($	$<\cdot, \doteq$	$<\cdot$	$<\cdot$	$<\cdot$		$<\cdot$			
$)$					$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$\vdash$	$<\cdot$	$<\cdot$	$<\cdot$	$<\cdot$			$<\cdot$		

СП-грамматики Пример 5

Получим таблицу приоритетов:

	S	A	B	x	$+$	$*$	$($	$)$	$\dashv$
S					$\doteq$			$\doteq$	$\cdot>$
A					$\cdot>$	$\doteq$		$\cdot>$	$\cdot>$
B					$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$($		$<\cdot$	$<\cdot$	$<\cdot$			$<\cdot$		
x					$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$+$		$<\cdot, \doteq$	$<\cdot$	$<\cdot$			$<\cdot$		$\cdot>$
$*$			$\doteq$	$<\cdot$			$<\cdot$		
$($	$<\cdot, \doteq$	$<\cdot$	$<\cdot$	$<\cdot$		$<\cdot$			
$)$					$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$\vdash$	$<\cdot$	$<\cdot$	$<\cdot$	$<\cdot$			$<\cdot$		

Из таблицы видно выполнение первого условия определения СП-граматики.

СП-грамматики. Пример 5

- Проверим выполнение второго условия определения СП-граматики: не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X <\cdot B$.

СП-грамматики. Пример 5

- Проверим выполнение второго условия определения СП-граматики: не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X <\cdot B$.
- Существуют только две пары правил, которые имеют вид $A \to \alpha X \beta$ и $B \to \beta$.

СП-грамматики. Пример 5

- Проверим выполнение второго условия определения СП-граматики: не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X <\cdot B$.
- Существуют только две пары правил, которые имеют вид $A \to \alpha X \beta$ и $B \to \beta$.
- Первая пара: $S \Rightarrow S + A$, $S \Rightarrow A$. Здесь $A = S$, $\alpha = S$, $X = +$, $\beta = A$, $B = S$. Но $+$ и S несравнимы.

СП-грамматики. Пример 5

- Проверим выполнение второго условия определения СП-граматики: не существует правил $A \to \alpha X \beta$ и $B \to \beta$ таких, что $X <\cdot B$.
- Существуют только две пары правил, которые имеют вид $A \to \alpha X \beta$ и $B \to \beta$.
- Первая пара: $S \Rightarrow S + A$, $S \Rightarrow A$. Здесь $A = S$, $\alpha = S$, $X = +$, $\beta = A$, $B = S$. Но $+$ и S несравнимы.

Свойство СП-грамматик (Замечание 3)

Замечание 3

Пусть $S \Rightarrow^+ \mu C w \Rightarrow \nu X \beta w$ — фрагмент правого вывода в СП-грамматике G (здесь S – аксиома, X – символ, C — нетерминал, w – цепочка терминалов, μ, ν и β —- цепочки). Тогда правило вида $B \to \beta$ применялось последним в данном выводе т. и т.т.к. в G нет правила вида $A \to \alpha X \beta$.

- Пусть последним в фрагменте вывода $S \Rightarrow^+ \mu C w \Rightarrow \nu X \beta w$ применялось правило $B \to \beta$. Тогда цепочка $\nu X B w$ является r-формой (почему?) и между символами X и B по замечанию 2 выполнено хотя бы одно из отношений предшествования. Отношение $X \cdot > B$ выполняться не может, так как B – нетерминал (замечание 1). Значит, $X <\cdot B$ или $X \doteq Y$, но тогда правила вида $A \to \alpha X \beta$ не существует по определению СП-грамматики.

Доказательство замечания 3

- Обратно, пусть в G нет правила вида $A \to \alpha X \beta$. Докажем, что β — основа r-формы $\nu X \beta w$. От противного: пусть это не так. Рассмотрим всевозможные случаи.

Доказательство замечания 3

- Обратно, пусть в G нет правила вида $A \to \alpha X \beta$. Докажем, что β — основа r-формы $\nu X \beta w$. От противного: пусть это не так. Рассмотрим всевозможные случаи.
  1. Основа r-формы β находится непосредственно справа от цепочки w. Но тогда предыдущая цепочка в правом выводе не могла бы заканчиваться на Cw (см.рис. 28-29). Приходим к противоречию.

Доказательство замечания 3

- Обратно, пусть в G нет правила вида $A \to \alpha X \beta$. Докажем, что β — основа r-формы $\nu X \beta w$. От противного: пусть это не так. Рассмотрим всевозможные случаи.
  1. Основа r-формы β находится непосредственно справа от цепочки w. Но тогда предыдущая цепочка в правом выводе не могла бы заканчиваться на Cw (см.рис. 28-29). Приходим к противоречию.
  2. Основа r-формы является собственным суффиксом $\nu X \beta w$, т.е., $\beta = \beta' Y \beta''$, β'' — основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, а значит $\nu X \beta' Y B w$ — тоже r-форма (см.рис. 30). Так как B — нетерминал, имеем $Y <\cdot B$ или $Y \doteq B$. Так как β'' — самая правая перед w основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, следовательно в рассматриваемом выводе $\mu C w \Rightarrow \nu X \beta w$ применялось правило $C \to \ldots Y \beta''$, но тогда это противоречит тому в СП-грамматике есть правило $B \to \beta''$.

Доказательство замечания 3

- Обратно, пусть в G нет правила вида $A \to \alpha X \beta$. Докажем, что β — основа r-формы $\nu X \beta w$. От противного: пусть это не так. Рассмотрим всевозможные случаи.
  1. Основа r-формы β находится непосредственно справа от цепочки w. Но тогда предыдущая цепочка в правом выводе не могла бы заканчиваться на Cw (см.рис. 28-29). Приходим к противоречию.
  2. Основа r-формы является собственным суффиксом $\nu X \beta w$, т.е., $\beta = \beta' Y \beta''$, β'' — основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, а значит $\nu X \beta' Y B w$ — тоже r-форма (см.рис. 30). Так как B — нетерминал, имеем $Y <\cdot B$ или $Y \doteq B$. Так как β'' — самая правая перед w основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, следовательно в рассматриваемом выводе $\mu C w \Rightarrow \nu X \beta w$ применялось правило $C \to \ldots Y \beta''$, но тогда это противоречит тому в СП-грамматике есть правило $B \to \beta''$.
  3. Основа имеет суффикс $X\beta$, т.е. $\beta = \alpha X \beta$, но с одной стороны, в грамматике есть правило $B \to \beta$, а с другой стороны, в ней нет правил вида $A \to \alpha X \beta$.

Доказательство замечания 3

- Обратно, пусть в G нет правила вида $A \to \alpha X \beta$. Докажем, что β — основа r-формы $\nu X \beta w$. От противного: пусть это не так. Рассмотрим всевозможные случаи.
  1. Основа r-формы β находится непосредственно справа от цепочки w. Но тогда предыдущая цепочка в правом выводе не могла бы заканчиваться на Cw (см.рис. 28-29). Приходим к противоречию.
  2. Основа r-формы является собственным суффиксом $\nu X \beta w$, т.е., $\beta = \beta' Y \beta''$, β'' — основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, а значит $\nu X \beta' Y B w$ — тоже r-форма (см.рис. 30). Так как B — нетерминал, имеем $Y <\cdot B$ или $Y \doteq B$. Так как β'' — самая правая перед w основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, следовательно в рассматриваемом выводе $\mu C w \Rightarrow \nu X \beta w$ применялось правило $C \to \ldots Y \beta''$, но тогда это противоречит тому в СП-грамматике есть правило $B \to \beta''$.
  3. Основа имеет суффикс $X\beta$, т.е. $\beta = \alpha X \beta$, но с одной стороны, в грамматике есть правило $B \to \beta$, а с другой стороны, в ней нет правил вида $A \to \alpha X \beta$.
- Следовательно, эта основа равна β.

Доказательство замечания 3

- Обратно, пусть в G нет правила вида $A \to \alpha X \beta$. Докажем, что β — основа r-формы $\nu X \beta w$. От противного: пусть это не так. Рассмотрим всевозможные случаи.
  1. Основа r-формы β находится непосредственно справа от цепочки w. Но тогда предыдущая цепочка в правом выводе не могла бы заканчиваться на Cw (см.рис. 28-29). Приходим к противоречию.
  2. Основа r-формы является собственным суффиксом $\nu X \beta w$, т.е., $\beta = \beta' Y \beta''$, β'' — основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, а значит $\nu X \beta' Y B w$ — тоже r-форма (см.рис. 30). Так как B — нетерминал, имеем $Y <\cdot B$ или $Y \doteq B$. Так как β'' — самая правая перед w основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, следовательно в рассматриваемом выводе $\mu C w \Rightarrow \nu X \beta w$ применялось правило $C \to \ldots Y \beta''$, но тогда это противоречит тому в СП-грамматике есть правило $B \to \beta''$.
  3. Основа имеет суффикс $X\beta$, т.е. $\beta = \alpha X \beta$, но с одной стороны, в грамматике есть правило $B \to \beta$, а с другой стороны, в ней нет правил вида $A \to \alpha X \beta$.
- Следовательно, эта основа равна β.
- Замечание доказано.

Доказательство замечания 3. Иллюстрация

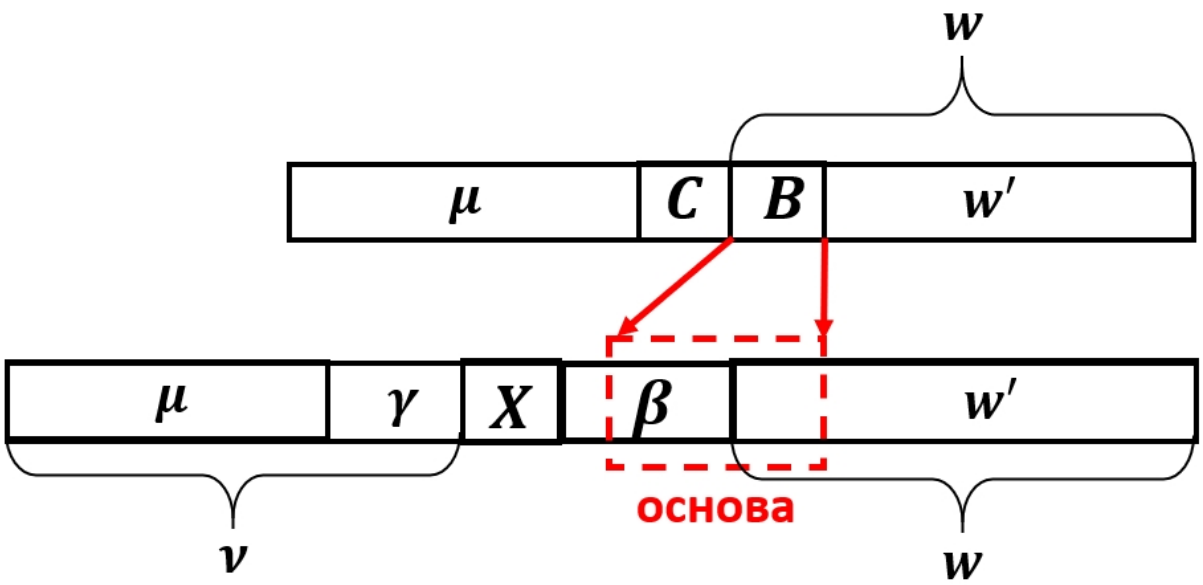


Рис. 28

Доказательство замечания 3. Иллюстрация

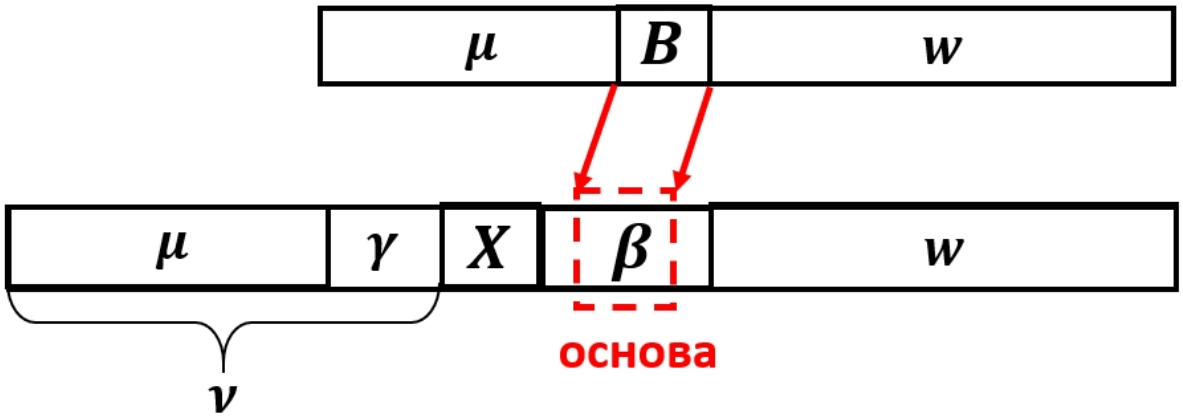


Рис. 29

Доказательство замечания 3. Иллюстрация

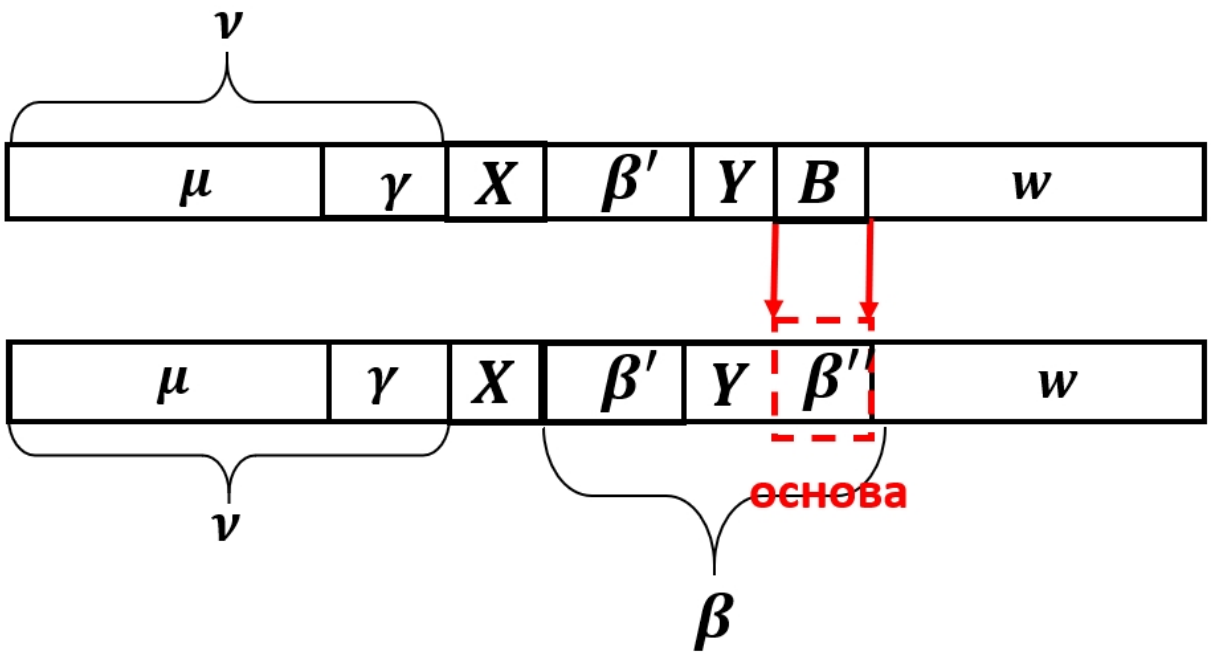


Рис. 30

Грамматика слабого предшествования. Теорема 4

И замечания 3 непосредственно следует

Грамматика слабого предшествования. Теорема 4

И замечания 3 непосредственно следует

Теорема 4

Пусть G — СП-грамматика, γ – ее r-форма и $\vdash \gamma \dashv = X_0X_1...X_nX_{n+1}$. Тогда основой формы γ является цепочка вида $X_{k_1}X_{k_1+1}\dots X_{l-1}X_l$ такая, что

1. l — минимальный номер, для которого $X_l \cdot > X_{l+1}$, $1 \le k_1 \le l \le n$;
2. имеет место следующие отношения

$$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$$

3. k_1 — минимальное такое, что $X_{k_1}X_{k_1+1}\dots X_{l-1}X_l$ — основа. Т.е. $X_{k_1}X_{k_1+1}\dots X_{l-1}X_l$ — максимальная основа среди основ вида $X_{k_i}X_{k_i+1}\dots X_{l-1}X_l$, $1 \le k_i \le l \le n$.

Грамматика слабого предшествования. Теорема 4

И замечания 3 непосредственно следует

Теорема 4

Пусть G — СП-грамматика, γ – ее r-форма и $\vdash \gamma \dashv = X_0X_1...X_nX_{n+1}$. Тогда основой формы γ является цепочка вида $X_{k_1}X_{k_1+1}\dots X_{l-1}X_l$ такая, что

1. l — минимальный номер, для которого $X_l \cdot > X_{l+1}$, $1 \le k_1 \le l \le n$;
2. имеет место следующие отношения

$$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$$

3. k_1 — минимальное такое, что $X_{k_1}X_{k_1+1}\dots X_{l-1}X_l$ — основа. Т.е. $X_{k_1}X_{k_1+1}\dots X_{l-1}X_l$ — максимальная основа среди основ вида $X_{k_i}X_{k_i+1}\dots X_{l-1}X_l$, $1 \le k_i \le l \le n$.

Из теоремы 3 имеем следующий алгоритм (см. следущий слайд).

Алгоритм восходящего анализа для СП-грамматики

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

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma =\vdash w \dashv$: (γ — текущая r-форма)

Алгоритм восходящего анализа для СП-грамматики

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

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0$

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0$

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_{l\cdot} > X_{l+1} \dots$)

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_{l \cdot} > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i <\cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i <\cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)
      $k = i + 1$

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i < \cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)
      $k = i + 1$
    case ($X_i \cdot > X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем правый конец $X_l = X_i$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i < \cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)
      $k = i + 1$
    case ($X_i \cdot > X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем правый конец $X_l = X_i$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$
      $l = i$

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i < \cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)
      $k = i + 1$
    case ($X_i \cdot > X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем правый конец $X_l = X_i$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$
      $l = i$
    else: ($X_i \doteq X_{i+1}$) or ($\stackrel{<\cdot}{\doteq}$)

Алгоритм восходящего анализа для СП-грамматики

Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова w
$\gamma = \vdash w \dashv$: (γ — текущая r-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина γ
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print(γ) (печатаем текущую r-форму вывода)
  $i = 0 \quad k = 0 \quad l = 0$
  while ($i \leq n$) and ($l = 0$): (пока не найдены лев. $X_k = X_{k_1}$ и правый X_l конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$ такой, что
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2-1} \stackrel{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \stackrel{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$)
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')
    case ($X_i <\cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)
      $k = i + 1$
    case ($X_i \cdot > X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем правый конец $X_l = X_i$ цепочки $X_k \dots X_l = X_{k_1} \dots X_l$
      $l = i$
    else: ($X_i \doteq X_{i+1}$) or ($\stackrel{<\cdot}{\doteq}$)
      $i = +1$ (двигаемся дальше, пока не найдем правый и левый конец цепочки $X_k \dots X_l = X_{k_1} \dots X_l$)

Алгоритм восходящего анализа для СП-грамматики (продолжение)

if $l = 0$ (не найден правый конец основы)
  print('$w \notin L(G)$')
else: ($l > 0$: найден правый конец основы)
    $j = k$
    while ($j < l$) and ($X_j \doteq X_{j+1}$) and ($\nexists A$: $A \to X_j \dots X_l$):
      $j+ = 1$ (двигаемся дальше, пока не найдем правый конец $X_k = X_{k_t}$ основы $X_k \dots X_l = X_{k_t} \dots X_l$)
    if ($j = l$) or not($X_{j-1} < \cdot X_j$):
    ($j = l$) — дошли до конца цепочки $X_k \dots X_l = X_{k_1} \dots X_l$, но не нашли основу $X_{k_t} \dots X_l$;
    not($X_{j-1} < \cdot X_j$): $X_j \dots X_l = X_{k_t} \dots X_l$ — основа, но не выполняется $X_j < \cdot X_{j+1}$
      print('$w \notin L(G)$')

Алгоритм восходящего анализа для СП-грамматики (продолжение)

if $l = 0$ (не найден правый конец основы)
  print('$w \notin L(G)$')
else: ($l > 0$: найден правый конец основы)
  $j = k$
  while ($j < l$) and ($X_j \doteq X_{j+1}$) and ($\nexists A$: $A \to X_j \dots X_l$):
    $j+ = 1$ (двигаемся дальше, пока не найдем правый конец $X_k = X_{k_t}$ основы $X_k \dots X_l = X_{k_t} \dots X_l$)
  if ($j = l$) or not($X_{j-1} < \cdot X_j$):
  ($j = l$) — дошли до конца цепочки $X_k \dots X_l = X_{k_1} \dots X_l$, но не нашли основу $X_{k_t} \dots X_l$;
  not($X_{j-1} < \cdot X_j$): $X_j \dots X_l = X_{k_t} \dots X_l$ — основа, но не выполняется $X_j < \cdot X_{j+1}$
    print('$w \notin L(G)$')
  else: (($j < l$) and ($X_j < \cdot X_{j+1}$): когда нашли основу, производим свертку по соотв. правилу $A \to X_k \dots X_l$)

Алгоритм восходящего анализа для СП-грамматики (продолжение)

if $l = 0$ (не найден правый конец основы)
  print('$w \notin L(G)$')
else: ($l > 0$: найден правый конец основы)
  $j = k$
  while ($j < l$) and ($X_j \doteq X_{j+1}$) and ($\nexists A$: $A \to X_j \dots X_l$):
    $j+ = 1$ (двигаемся дальше, пока не найдем правый конец $X_k = X_{k_t}$ основы $X_k \dots X_l = X_{k_t} \dots X_l$)
  if ($j = l$) or not($X_{j-1} < \cdot X_j$):
  ($j = l$) — дошли до конца цепочки $X_k \dots X_l = X_{k_1} \dots X_l$, но не нашли основу $X_{k_t} \dots X_l$;
  not($X_{j-1} < \cdot X_j$): $X_j \dots X_l = X_{k_t} \dots X_l$ — основа, но не выполняется $X_j < \cdot X_{j+1}$
    print('$w \notin L(G)$')
  else: (($j < l$) and ($X_j < \cdot X_{j+1}$): когда нашли основу, производим свертку по соотв. правилу $A \to X_k \dots X_l$)
    $k = j$
    $\gamma = X_0 X_1 \dots X_{k-1} A X_{l+1} \dots X_n X_{n+1}$
(возвращаемся в цикл while ($\gamma \neq \vdash S \dashv$))

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$$

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{A \doteq * <\cdot B} \cdot> \dashv$ нет основы $S+A*B$, но есть основа $A*B$

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{A \doteq * <\cdot B} \cdot> \dashv$ нет основы $S+A*B$, но есть основа $A*B$

$\vdash <\cdot \boxed{S \doteq + <\cdot A} \cdot> \dashv$

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{A \doteq * <\cdot B} \cdot> \dashv$ нет основы $S+A*B$, но есть основа $A*B$

$\vdash <\cdot \boxed{S \doteq + <\cdot A} \cdot> \dashv$

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

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{A \doteq * <\cdot B} \cdot> \dashv$ нет основы $S+A*B$, но есть основа $A*B$

$\vdash <\cdot \boxed{S \doteq + <\cdot A} \cdot> \dashv$

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

И восстановим соответствующий правосторонний вывод для грамматики

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{A \doteq * <\cdot B} \cdot> \dashv$ нет основы $S+A*B$, но есть основа $A*B$

$\vdash <\cdot \boxed{S \doteq + <\cdot A} \cdot> \dashv$

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

И восстановим соответствующий правосторонний вывод для грамматики

$S \Rightarrow \underline{S + A} \Rightarrow S + \underline{A * B} \Rightarrow S + A * \underline{x} \Rightarrow S + \underline{B} * x \Rightarrow S + \underline{x} * x \Rightarrow \underline{A} + x * x \Rightarrow \underline{B} + x * x \Rightarrow \underline{x} + x * x$

Пример 6

Пример 6. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для СП грамматики из примера 3, опираясь на теорему 4, основываясь на алгоритме восходящего анализа для СП-грамматик.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{B} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot \boxed{A} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot S \doteq + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+x$, зато есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{B} \cdot> * <\cdot x \cdot> \dashv$ нет основы $S+B$, зато есть основа B

$\vdash <\cdot S \doteq + <\cdot A \doteq * <\cdot \boxed{x} \cdot> \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash <\cdot S \doteq + <\cdot \boxed{A \doteq * <\cdot B} \cdot> \dashv$ нет основы $S+A*B$, но есть основа $A*B$

$\vdash <\cdot \boxed{S \doteq + <\cdot A} \cdot> \dashv$

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

И восстановим соответствующий правосторонний вывод для грамматики

$S \Rightarrow \underline{S + A} \Rightarrow S + \underline{A * B} \Rightarrow S + A * \underline{x} \Rightarrow S + \underline{B} * x \Rightarrow S + \underline{x} * x \Rightarrow \underline{A} + x * x \Rightarrow \underline{B} + x * x \Rightarrow \underline{x} + x * x$

Операторные грамматики. Определение

Определение операторной грамматики

ε-свободная КС-грамматика G называется операторный, если в правых частях правил вывода нет нетерминалов, стоящих рядом.

Грамматика G из примера 5
$S \to S + A \mid A$, $A \to A * B \mid B$, $B \to x \mid (S)$ является операторной.
Эта грамматика, как доказано в лекции 3, является однозначной.

Операторные грамматики. Определение

Определение операторной грамматики

ε-свободная КС-грамматика G называется операторный, если в правых частях правил вывода нет нетерминалов, стоящих рядом.

Грамматика G из примера 5
$S \to S + A \mid A$, $A \to A * B \mid B$, $B \to x \mid (S)$ является операторной.
Эта грамматика, как доказано в лекции 3, является однозначной.

Грамматика из следующего примера не является однозначной (почему?), но тоже является операторной.

Операторные грамматики. Определение

Определение операторной грамматики

ε-свободная КС-грамматика G называется операторный, если в правых частях правил вывода нет нетерминалов, стоящих рядом.

Грамматика G из примера 5
$S \to S + A \mid A$, $A \to A * B \mid B$, $B \to x \mid (S)$ является операторной.
Эта грамматика, как доказано в лекции 3, является однозначной.

Грамматика из следующего примера не является однозначной (почему?), но тоже является операторной.

Пример 7
$S \to S + S \mid S * S \mid (S) \mid x$

Операторные грамматики. Свойства

Замечание 3

Если некоторая r-форма грамматики выражений содержит цепочку aSb, и ровно один из терминалов a, b принадлежит основе этой формы, то S также принадлежит основе.

Доказательство.

Замечание 3

Если некоторая r-форма грамматики выражений содержит цепочку aSb, и ровно один из терминалов a, b принадлежит основе этой формы, то S также принадлежит основе.

Доказательство.

- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод.

Замечание 3

Если некоторая r-форма грамматики выражений содержит цепочку aSb, и ровно один из терминалов a, b принадлежит основе этой формы, то S также принадлежит основе.

Доказательство.

- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод.
- И пусть $\alpha_i = uaSbv$ для некоторых $u, v \in (\Sigma \cup \Gamma)^*$, причем $u = u_1 u_2$, $\gamma = u_2 a$ — некоторая основа, т.е. есть правило $A \to \gamma$.

Замечание 3

Если некоторая r-форма грамматики выражений содержит цепочку aSb, и ровно один из терминалов a, b принадлежит основе этой формы, то S также принадлежит основе.

Доказательство.

- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод.
- И пусть $\alpha_i = uaSbv$ для некоторых $u, v \in (\Sigma \cup \Gamma)^*$, причем $u = u_1 u_2$, $\gamma = u_2 a$ — некоторая основа, т.е. есть правило $A \to \gamma$.
- Тогда для некоторого $1 \leq j < i$ имеем $\alpha_j = u_1 ASbv$, что невозможно для операторной грамматики (почему).

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.
- Выражение состоит из операндов, операторов и скобок. Мы будем рассматривать операторные грамматики, порождающие язык, состоящий из выражений.

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.
- Выражение состоит из операндов, операторов и скобок. Мы будем рассматривать операторные грамматики, порождающие язык, состоящий из выражений.
- Для пар соседних терминалов r выражения определим отношения приоритета следующим образом:
  1. $a \doteq b$, если a и b должны быть свернуты на одном шаге;

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.
- Выражение состоит из операндов, операторов и скобок. Мы будем рассматривать операторные грамматики, порождающие язык, состоящий из выражений.
- Для пар соседних терминалов r выражения определим отношения приоритета следующим образом:
  1. $a \doteq b$, если a и b должны быть свернуты на одном шаге;
  2. $a < \cdot b$, если b должен быть свернут раньше a;

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.
- Выражение состоит из операндов, операторов и скобок. Мы будем рассматривать операторные грамматики, порождающие язык, состоящий из выражений.
- Для пар соседних терминалов r выражения определим отношения приоритета следующим образом:
  1. $a \doteq b$, если a и b должны быть свернуты на одном шаге;
  2. $a <\cdot b$, если b должен быть свернут раньше a;
  3. $a \cdot > b$, если a должен быть свернут раньше b;

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.
- Выражение состоит из операндов, операторов и скобок. Мы будем рассматривать операторные грамматики, порождающие язык, состоящий из выражений.
- Для пар соседних терминалов r выражения определим отношения приоритета следующим образом:
  1. $a \doteq b$, если a и b должны быть свернуты на одном шаге;
  2. $a < \cdot b$, если b должен быть свернут раньше a;
  3. $a \cdot > b$, если a должен быть свернут раньше b;
  4. $\vdash < \cdot b$, если b может быть первым терминалом выражения;

Операторные грамматики. Отношение приоритета

- Два терминала такой формы назовем соседними, если между ними нет других терминалов.
- Из замечания 3 следует, что между соседними терминалами в форме находится либо одиночный нетерминал, либо не находится ничего.
- Выражение состоит из операндов, операторов и скобок. Мы будем рассматривать операторные грамматики, порождающие язык, состоящий из выражений.
- Для пар соседних терминалов r выражения определим отношения приоритета следующим образом:
  1. $a \doteq b$, если a и b должны быть свернуты на одном шаге;
  2. $a < \cdot b$, если b должен быть свернут раньше a;
  3. $a \cdot > b$, если a должен быть свернут раньше b;
  4. $\vdash < \cdot b$, если b может быть первым терминалом выражения;
  5. $a \cdot > \dashv$, если a может быть последним терминалом выражения.

Операторные грамматики. Приоритет операций

- Для определения приоритетов необязательно, чтобы грамматика была однозначной. Более того, для неоднозначных грамматик легче расставлять приоритеты, поскольку они проще и логичнее выглядят, чем те однозначные, по которым они построены.

Операторные грамматики. Приоритет операций

- Для определения приоритетов необязательно, чтобы грамматика была однозначной. Более того, для неоднозначных грамматик легче расставлять приоритеты, поскольку они проще и логичнее выглядят, чем те однозначные, по которым они построены.
- Приоритет операций таких грамматик, там где он не указан явно при помощи скобок, основан на соглашениях о приоритете операторов и их ассоциативности, т.е. порядке выполнения одноименных операторов.

Операторные грамматики. Приоритет операций

- Расставить отношения приоритета для прочих пар терминалов помогают следующие замечания. Во-первых, атомарные операнды сворачиваются в первую очередь. Далее, содержимое скобок сворачивается раньше того, что за скобками; при вложенных скобках вначале сворачивается содержимое внутренних скобок. Парные скобки сворачиваются на одном шаге, так же как и функциональные конструкции типа $\min(E; E)$.

Операторные грамматики. Приоритет операций

- Расставить отношения приоритета для прочих пар терминалов помогают следующие замечания. Во-первых, атомарные операнды сворачиваются в первую очередь. Далее, содержимое скобок сворачивается раньше того, что за скобками; при вложенных скобках вначале сворачивается содержимое внутренних скобок. Парные скобки сворачиваются на одном шаге, так же как и функциональные конструкции типа $\min(E; E)$.
- Пример 7 Приоритет возведения в степень выше приоритета умножения, который, в свою очередь, выше приоритета сложения. Оператор возведения в степень правоассоциативен : $x^{y^z} = x^{(y^z)}$, а умножения и сложения — левоассоциативен: $x + y + z = (x + y) + z$ и $x * y * z = (x * y) * z$.

Расстановка приоритетов в грамматике арифметических операций. Пример 8

- Пример 8 Расставим приоритеты в грамматике арифметических операций из примера 7

$$S \to S + S \mid S * S \mid (S) \mid x$$

Расстановка приоритетов в грамматике арифметических операций. Пример 8

- Пример 8 Расставим приоритеты в грамматике арифметических операций из примера 7

$$S \to S + S \mid S * S \mid (S) \mid x$$

- Напомним, что приоритеты расставляются только для терминалов из $\Sigma = \{x, +, *, (,)\}$ и значков $\vdash$, $\dashv$.

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.
  1. Между терминалами x и x невозможно расставить приоритет, поскольку слово xx не встречается ни в одной форме, и слово xSx не встречается ни в одной форме.

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.
  1. Между терминалами x и x невозможно расставить приоритет, поскольку слово xx не встречается ни в одной форме, и слово xSx не встречается ни в одной форме.
  2. $x \cdot > +$, поскольку в слове $x+$ основа-операнд x сворачивается раньше оператора $+$, а слово $xS+$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.
  1. Между терминалами x и x невозможно расставить приоритет, поскольку слово xx не встречается ни в одной форме, и слово xSx не встречается ни в одной форме.
  2. $x \cdot > +$, поскольку в слове $x+$ основа-операнд x сворачивается раньше оператора $+$, а слово $xS+$ не встречается ни в одной форме.
  3. $x \cdot > *$ — аналогично

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.
  1. Между терминалами x и x невозможно расставить приоритет, поскольку слово xx не встречается ни в одной форме, и слово xSx не встречается ни в одной форме.
  2. $x \cdot > +$, поскольку в слове $x+$ основа-операнд x сворачивается раньше оператора $+$, а слово $xS+$ не встречается ни в одной форме.
  3. $x \cdot > *$ — аналогично
  4. Между терминалами x и (невозможно расставить приоритет, поскольку поскольку ни слово $x($, ни слово $xS($ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.
  1. Между терминалами x и x невозможно расставить приоритет, поскольку слово xx не встречается ни в одной форме, и слово xSx не встречается ни в одной форме.
  2. $x \cdot > +$, поскольку в слове $x+$ основа-операнд x сворачивается раньше оператора $+$, а слово $xS+$ не встречается ни в одной форме.
  3. $x \cdot > *$ — аналогично
  4. Между терминалами x и (невозможно расставить приоритет, поскольку поскольку ни слово $x($, ни слово $xS($ не встречается ни в одной форме.
  5. $x \cdot >)$, поскольку основа-операнд x сворачивается раньше скобки), а слово $xS)$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала x

- Расставим приоритеты для терминала x.
  1. Между терминалами x и x невозможно расставить приоритет, поскольку слово xx не встречается ни в одной форме, и слово xSx не встречается ни в одной форме.
  2. $x \cdot > +$, поскольку в слове $x+$ основа-операнд x сворачивается раньше оператора $+$, а слово $xS+$ не встречается ни в одной форме.
  3. $x \cdot > *$ — аналогично
  4. Между терминалами x и (невозможно расставить приоритет, поскольку поскольку ни слово $x($, ни слово $xS($ не встречается ни в одной форме.
  5. $x \cdot >)$, поскольку основа-операнд x сворачивается раньше скобки), а слово $xS)$ не встречается ни в одной форме.
  6. $x \cdot > \dashv$, поскольку арифметическое выражение может заканчиваться на основу-операнд x, а слово $xS \dashv$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.
  1. $+ \lessdot \cdot x$, поскольку основа-операнд x сворачивается раньше оператора +, а слово $+Sx$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.
  1. $+ \lessdot \cdot x$, поскольку основа-операнд x сворачивается раньше оператора +, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \cdot \gtrdot +$ так как ввиду левоассоциативности + в слове $+S+$ префикс $+S$ сворачивается раньше суффикса +, а слово $++$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.
  1. $+ \lessdot \cdot x$, поскольку основа-операнд x сворачивается раньше оператора +, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \cdot \gtrdot +$ так как ввиду левоассоциативности + в слове $+S+$ префикс $+S$ сворачивается раньше суффикса +, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot \cdot *$, поскольку приоритет $*$ выше +: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.
  1. $+ \lessdot \cdot x$, поскольку основа-операнд x сворачивается раньше оператора +, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \cdot \gtrdot +$ так как ввиду левоассоциативности + в слове $+S+$ префикс $+S$ сворачивается раньше суффикса +, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot \cdot *$, поскольку приоритет $*$ выше +: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.
  4. $+ \lessdot \cdot ($, поскольку приоритет скобки (выше всех операторов: в слове $+($ суффикс (сворачивается раньше префикса +, а слово $+S($ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.
  1. $+ \lessdot \cdot x$, поскольку основа-операнд x сворачивается раньше оператора +, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \cdot \gtrdot +$ так как ввиду левоассоциативности + в слове $+S+$ префикс $+S$ сворачивается раньше суффикса +, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot \cdot *$, поскольку приоритет $*$ выше +: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.
  4. $+ \lessdot \cdot ($, поскольку приоритет скобки (выше всех операторов: в слове $+($ суффикс (сворачивается раньше префикса +, а слово $+S($ не встречается ни в одной форме.
  5. $+ \cdot \gtrdot)$: в слове $+S)$ префикс $+S$ сворачивается раньше скобки), а слово $+)$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминала +

- Расставим приоритеты для терминала +, предполагая левоассоциативность этой операции.
  1. $+ \lessdot \cdot x$, поскольку основа-операнд x сворачивается раньше оператора +, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \cdot \gtrdot +$ так как ввиду левоассоциативности + в слове $+S+$ префикс $+S$ сворачивается раньше суффикса +, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot \cdot *$, поскольку приоритет $*$ выше +: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.
  4. $+ \lessdot \cdot ($, поскольку приоритет скобки (выше всех операторов: в слове $+($ суффикс (сворачивается раньше префикса +, а слово $+S($ не встречается ни в одной форме.
  5. $+ \cdot \gtrdot)$: в слове $+S)$ префикс $+S$ сворачивается раньше скобки), а слово $+)$ не встречается ни в одной форме.
  6. $+ \cdot \gtrdot \dashv$, поскольку арифметическое выражение может заканчиваться на $+S$, т.е. может быть ситуация $+S \dashv$, но не на +, т.е. не может быть ситуации $+ \dashv$.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\gtrdot *$, $+\lessdot *$, $*\lessdot ($, $*\cdot\gtrdot)$, $*\cdot\gtrdot \dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\gtrdot *$, $+\lessdot *$, $*\lessdot ($, $*\cdot\gtrdot)$, $*\cdot\gtrdot \dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.
  1. $(\lessdot x$, поскольку основа-операнд x сворачивается раньше скобки (, а слово $(Sx$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\gtrdot *$, $+\lessdot *$, $*\lessdot ($, $*\cdot\gtrdot)$, $*\cdot\gtrdot \dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.
  1. $(\lessdot x$, поскольку основа-операнд x сворачивается раньше скобки (, а слово $(Sx$ не встречается ни в одной форме.
  2. $(\lessdot +$ так как в слове $(S+$ суффикс $S+$ сворачивается раньше префикса (, а слово $(+$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\gtrdot *$, $+\lessdot *$, $*\lessdot ($, $*\cdot\gtrdot)$, $*\cdot\gtrdot \dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.
  1. $(\lessdot x$, поскольку основа-операнд x сворачивается раньше скобки (, а слово $(Sx$ не встречается ни в одной форме.
  2. $(\lessdot +$ так как в слове $(S+$ суффикс $S+$ сворачивается раньше префикса (, а слово $(+$ не встречается ни в одной форме.
  3. $(\lessdot *$ так как в слове $(S*$ суффикс $S*$ сворачивается раньше префикса (, а слово $(*$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\!>*$, $+\lessdot *$, $*\lessdot($, $*\cdot\!>)$, $*\cdot\!>\dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.
  1. $(\lessdot x$, поскольку основа-операнд x сворачивается раньше скобки (, а слово $(Sx$ не встречается ни в одной форме.
  2. $(\lessdot +$ так как в слове $(S+$ суффикс $S+$ сворачивается раньше префикса (, а слово $(+$ не встречается ни в одной форме.
  3. $(\lessdot *$ так как в слове $(S*$ суффикс $S*$ сворачивается раньше префикса (, а слово $(*$ не встречается ни в одной форме.
  4. $(\lessdot ($, поскольку открытая скобка справа сворачивается раньше открытой скобки слева, а слово $(S($ не встречается ни в одной форме.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\!>*$, $+\lessdot *$, $*\lessdot($, $*\cdot\!>)$, $*\cdot\!>\dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.
  1. $(\lessdot x$, поскольку основа-операнд x сворачивается раньше скобки (, а слово $(Sx$ не встречается ни в одной форме.
  2. $(\lessdot +$ так как в слове $(S+$ суффикс $S+$ сворачивается раньше префикса (, а слово $(+$ не встречается ни в одной форме.
  3. $(\lessdot *$ так как в слове $(S*$ суффикс $S*$ сворачивается раньше префикса (, а слово $(*$ не встречается ни в одной форме.
  4. $(\lessdot ($, поскольку открытая скобка справа сворачивается раньше открытой скобки слева, а слово $(S($ не встречается ни в одной форме.
  5. $(\doteq)$: так как есть основа (S), а слово $()$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминалов $*$ и (

- Приоритеты для терминала $*$ расставляются аналогично: $*\lessdot x$, $*\cdot\!>*$, $+\lessdot *$, $*\lessdot($, $*\cdot\!>)$, $*\cdot\!>\dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.
  1. $(\lessdot x$, поскольку основа-операнд x сворачивается раньше скобки (, а слово $(Sx$ не встречается ни в одной форме.
  2. $(\lessdot +$ так как в слове $(S+$ суффикс $S+$ сворачивается раньше префикса (, а слово $(+$ не встречается ни в одной форме.
  3. $(\lessdot *$ так как в слове $(S*$ суффикс $S*$ сворачивается раньше префикса (, а слово $(*$ не встречается ни в одной форме.
  4. $(\lessdot ($, поскольку открытая скобка справа сворачивается раньше открытой скобки слева, а слово $(S($ не встречается ни в одной форме.
  5. $(\doteq)$: так как есть основа (S), а слово $()$ не встречается ни в одной форме.
  6. Между (и $\dashv$ невозможно расставить приоритет, поскольку ни слово $(\dashv$ не встречается ни в одной форме, ни слово $(S\dashv$ не встречается ни в одной форме.

Пример 8. Приоритеты для терминалов) и $\vdash$. Таблица приоритетов

- Приоритеты для терминала скобка) расставляются аналогично:
  $)\cdot>+$, $)\cdot>*$, $)\cdot>)$, $)\cdot>\dashv$.
  Между) и другими терминалами нельзя расставить приоритеты.
- Арифметическое выражение может начинаться только с x, $S+$, $S*$, $(S$, поэтому имеем
  $\vdash<\cdot x$, $\vdash<\cdot +$, $\vdash<\cdot *$, $\vdash<\cdot *$.
  Между (и другими терминалами нельзя расставить приоритеты.

Имеем следующую таблицу приоритетов.

	x	$+$	$*$	$($	$)$	$\dashv$
x		$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$+$	$<\cdot$	$\cdot>$	$<\cdot$	$<\cdot$	$\cdot>$	$\cdot>$
$*$	$<\cdot$	$\cdot>$	$\cdot>$	$<\cdot$	$\cdot>$	$\cdot>$
$($	$<\cdot$	$<\cdot$	$<\cdot$	$<\cdot$	$\doteq$	
$)$		$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$\vdash$	$<\cdot$	$<\cdot$	$<\cdot$	$<\cdot$		

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot > * <\cdot x \cdot > \dashv$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot > \dashv$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot > \dashv$

$\vdash <\cdot_s + \boxed{<\cdot_s * \cdot >_s} \dashv$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$$

$$\vdash <\cdot_s + <\cdot \boxed{x} \cdot > * <\cdot x \cdot > \dashv$$

$$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot > \dashv$$

$$\vdash <\cdot_s + \boxed{<\cdot_s * \cdot >_s} \dashv$$

$$\vdash \boxed{<\cdot_s + \cdot >_s} \dashv$$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot > \dashv$

$\vdash <\cdot_s + \boxed{<\cdot_s * \cdot >_s} \dashv$

$\vdash \boxed{<\cdot_s + \cdot >_s} \dashv$

$\vdash S \dashv$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot > + <\cdot x \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot > * <\cdot x \cdot > \dashv$

$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot > \dashv$

$\vdash <\cdot_s + \boxed{<\cdot_s * \cdot >_s} \dashv$

$\vdash \boxed{<\cdot_s + \cdot >_s} \dashv$

$\vdash S \dashv$

И восстановим соответствующий правосторонний вывод для грамматики

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot> \dashv$

$\vdash <\cdot_s + \boxed{<\cdot_s * \cdot>_s} \dashv$

$\vdash \boxed{<\cdot_s + \cdot>_s} \dashv$

$\vdash S \dashv$

И восстановим соответствующий правосторонний вывод для грамматики

$S \Rightarrow \underline{S + S} \Rightarrow S + \underline{S * S} \Rightarrow S + S * \underline{x} \Rightarrow S + \underline{x} * x \Rightarrow \underline{x} + x * x$

Пример 9. Обработка и вывод правильной цепочки для грамматики из примера 8

Пример 9. Рассмотрим восходящий анализ для цепочки $w = x + x * x$ для операторной грамматики из примера 8.

$\vdash <\cdot \boxed{x} \cdot> + <\cdot x \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot_s + <\cdot \boxed{x} \cdot> * <\cdot x \cdot> \dashv$

$\vdash <\cdot_s + <\cdot_s * \boxed{x} \cdot> \dashv$

$\vdash <\cdot_s + \boxed{<\cdot_s * \cdot>_s} \dashv$

$\vdash \boxed{<\cdot_s + \cdot>_s} \dashv$

$\vdash S \dashv$

И восстановим соответствующий правосторонний вывод для грамматики

$S \Rightarrow \underline{S + S} \Rightarrow S + \underline{S * S} \Rightarrow S + S * \underline{x} \Rightarrow S + \underline{x} * x \Rightarrow \underline{x} + x * x$

Обработчик ошибок для грамматики из примера 8

Напишем обработчик ошибок для грамматики из примера 8.
Имеем следующие типы синтаксических ошибок для этой грамматики. Их три:

Обработчик ошибок для грамматики из примера 8

Напишем обработчик ошибок для грамматики из примера 8.
Имеем следующие типы синтаксических ошибок для этой грамматики. Их три:

- e_1: отсутствует оператор (два операнда подряд), вставить $+$;
- e_2: лишняя (, удалить её;
- e_3: лишняя), удалить её.

Имеем следующий обработчик ошибок.

	x	$+$	$*$	$($	$)$	$\dashv$
x	e_1	$\cdot>$	$\cdot>$		$\cdot>$	$\cdot>$
$+$	$<\cdot$	$\cdot>$	$<\cdot$	$<\cdot$	$\cdot>$	$\cdot>$
$*$	$<\cdot$	$\cdot>$	$\cdot>$	$<\cdot$	$\cdot>$	$\cdot>$
$($	$<\cdot$	$<\cdot$	$<\cdot$	$<\cdot$	$\doteq$	e_2
$)$	e_1	$\cdot>$	$\cdot>$	e_2	$\cdot>$	$\cdot>$
$\vdash$	$<\cdot$	$<\cdot$	$<\cdot$	$<\cdot$	e_3	

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3) \cdot > x \;\; x \cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3) \cdot > x \;\; x \cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

$\vdash < \cdot x e_1 x \cdot > + \cdot > \dashv$ ошибка e_1 в позиции 3 исходной цепочки

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3) \cdot > x \;\; x \cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

$\vdash < \cdot x e_1 x \cdot > + \cdot > \dashv$ ошибка e_1 в позиции 3 исходной цепочки

$\vdash < \cdot \boxed{x} \cdot > + \cdot > \;\; x \cdot > + \cdot > \dashv$

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3) \cdot > x \;\; x \cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

$\vdash < \cdot x e_1 x \cdot > + \cdot > \dashv$ ошибка e_1 в позиции 3 исходной цепочки

$\vdash < \cdot \boxed{x} \cdot > + \cdot > \;\; x \cdot > + \cdot > \dashv$

$\vdash \;\; < \cdot_s + < \cdot \boxed{x} \cdot > + \cdot > \dashv$

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3)\cdot > x \;\; x\cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

$\vdash < \cdot x e_1 x \cdot > + \cdot > \dashv$ ошибка e_1 в позиции 3 исходной цепочки

$\vdash < \cdot \boxed{x} \cdot > + \cdot > \; x \cdot > + \cdot > \dashv$

$\vdash \; < \cdot_s + < \cdot \boxed{x} \cdot > + \cdot > \dashv$

$\vdash \boxed{< \cdot_s + \cdot >_s} + \cdot > \dashv$

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3) \cdot > x \;\; x \cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

$\vdash < \cdot x e_1 x \cdot > + \cdot > \dashv$ ошибка e_1 в позиции 3 исходной цепочки

$\vdash < \cdot \boxed{x} \cdot > + \cdot > \; x \cdot > + \cdot > \dashv$

$\vdash \; < \cdot_s + < \cdot \boxed{x} \cdot > + \cdot > \dashv$

$\vdash \boxed{< \cdot_s + \cdot >_s} + \cdot > \dashv$

$\vdash < \cdot_s + \cdot > \dashv$ нет подходящей основы — неизвестная ошибка в позиции 4 исходной цепочки

Пример 10. Обработка ошибочной цепочки $w =)xx+$ для грамматики из примера 8

Пример 10. Рассмотрим обработку ошибочной цепочки $w =)xx+$ для операторной грамматики из примера 8.

$\vdash e_3)\cdot > x \;\; x\cdot > + \cdot > \dashv$ ошибка e_3 в позиции 1 исходной цепочки

$\vdash < \cdot x e_1 x \cdot > + \cdot > \dashv$ ошибка e_1 в позиции 3 исходной цепочки

$\vdash < \cdot \boxed{x} \cdot > + \cdot > \; x \cdot > + \cdot > \dashv$

$\vdash \; < \cdot_s + < \cdot \boxed{x} \cdot > + \cdot > \dashv$

$\vdash \boxed{< \cdot_s + \cdot >_s} + \cdot > \dashv$

$\vdash < \cdot_s + \cdot > \dashv$ нет подходящей основы — неизвестная ошибка в позиции 4 исходной цепочки

Мы видим, что наш обработчик ошибок не смог спарвиться с данной цепочкой и не нашел последнюю ошибку.