Лингвистические основы информатики
Лекция 17, часть 2
Грамматик слабого 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)

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 $X \doteq B$.

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 $X \doteq B$.

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 $X \doteq B$.

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. От противного Пусть ПП-грамматика G имеет правила вывода $A \to \alpha X \beta$ и $B \to \beta$ и при этом $X \doteq B$ или $X \lessdot B$. Так как G ε-свободна, $\beta \neq \varepsilon$, поэтому $\beta = Y\gamma$ для некоторого символа Y. Тогда $X \doteq Y$, поскольку правило вывода $A \to \alpha X \beta$ можно записать как $A \to \alpha XY\gamma$ (см. рис. 1).

Грамматика слабого предшествования

Доказательство. Пусть G — ПП-грамматика. Достаточно проверить (2)-е условие из определения СП-грамматики. От противного Пусть ПП-грамматика G имеет правила вывода $A \to \alpha X \beta$ и $B \to \beta$ и при этом $X \doteq B$ или $X \lessdot B$. Так как G ε-свободна, $\beta \neq \varepsilon$, поэтому $\beta = Y\gamma$ для некоторого символа Y. Тогда $X \doteq Y$, поскольку правило вывода $A \to \alpha X \beta$ можно записать как $A \to \alpha XY\gamma$ (см. рис. 1).

1. Пусть $X \doteq B$, тогда существует правило вывода $C \to \mu XB\nu$, поскольку $Y \in FIRST'(B)$, получаем $X \lessdot Y$, что противоречит $X \doteq Y$ (см. рис. 2a).

Грамматика слабого предшествования

Доказательство. Пусть G — ПП-грамматика. Достаточно проверить (2)-е условие из определения СП-грамматики. От противного Пусть ПП-грамматика G имеет правила вывода $A \to \alpha X \beta$ и $B \to \beta$ и при этом $X \doteq B$ или $X \lessdot B$. Так как G ε-свободна, $\beta \neq \varepsilon$, поэтому $\beta = Y\gamma$ для некоторого символа Y. Тогда $X \doteq Y$, поскольку правило вывода $A \to \alpha X \beta$ можно записать как $A \to \alpha X Y \gamma$ (см. рис. 1).

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 $X \doteq Y$ (см. рис. 2а).
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 $X \doteq Y$ (см. рис. 2б).

Доказательство теоремы 4. Иллюстрация

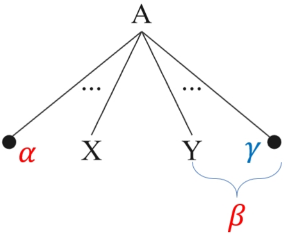

Рис. 1

Доказательство теоремы 4. Иллюстрация

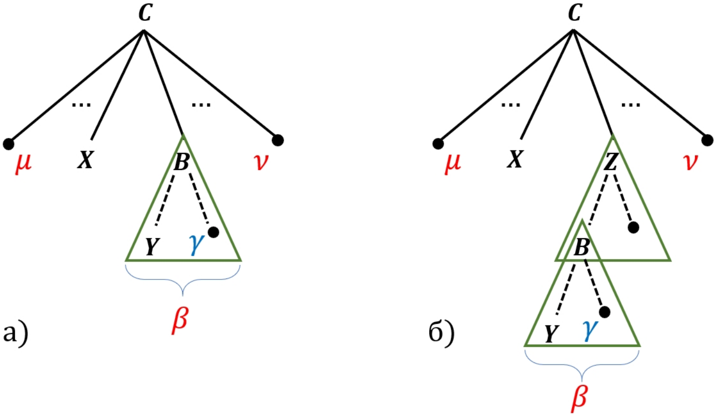

Рис. 2

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 $S \rightarrow S + A \mid A,\ A \rightarrow A * B \mid B,\ B \rightarrow x \mid (S)$,
- 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)

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

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 $X <\cdot B$ или $X \doteq B$.

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 $X \lessdot B$ или $X \doteq B$.

- 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 $X \lessdot B$ или $X \doteq B$.

- Существуют только две пары правил, которые имеют вид $A \to \alpha X \beta$ и $B \to \beta$.

- Первая пара: $S \to S + A$, $S \to A$. Здесь $A = S$, $\alpha = S$, $X = +$, $\beta = A$, $B = S$, $B \to \beta = S \to A$. 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 $X \lessdot B$ или $X \doteq B$.
- Существуют только две пары правил, которые имеют вид $A \to \alpha X \beta$ и $B \to \beta$.
- Первая пара: $S \to S + A$, $S \to A$. Здесь $A = S$, $\alpha = S$, $X = +$, $\beta = A$, $B = S$, $B \to \beta = S \to A$. Но $+$ и S несравнимы.
- Вторая пара: $A \to A * B$, $A \to B$. Здесь $A = A$, $\alpha = A$, $X = *$, $\beta = B$, $B = A$, $B \to \beta = A \to B$. Но $*$ и A 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 грамматически символов) (см. рис. 3). Тогда правило вида $B \to \beta$ применялось последним в данном выводе т. и т.т.к. в G нет правила вида $A \to \alpha X \beta$.

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 грамматически символов) (см. рис. 3). Тогда правило вида $B \rightarrow \beta$ применялось последним в данном выводе т. и т.т.к. в G нет правила вида $A \rightarrow \alpha X \beta$.

- Пусть последним в фрагменте вывода $S \Rightarrow^+ \mu C w \Rightarrow \nu X \beta w$ применялось правило $B \rightarrow \beta$. 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 $X \doteq B$, но тогда правила вида $A \rightarrow \alpha X \beta$ 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. Пусть правый конец основы β'' r-формы $\nu X \beta w$ находится непосредственно справа или слева от левого конца цепочки w, тогда предыдущая r-форма в правом выводе не могла бы заканчиваться на Cw (см.рис. 4-5). 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. Пусть правый конец основы β'' r-формы $\nu X\beta w$ находится непосредственно справа или слева от левого конца цепочки w, тогда предыдущая r-форма в правом выводе не могла бы заканчиваться на Cw (см.рис. 4-5). Приходим к противоречию.
  2. Пусть основа r-формы является собственным суффиксом $\nu X\beta$, т.е., $\beta = \beta' Y\beta''$, β'' — основа r-формы $\nu X\beta w = \nu X\beta' Y\beta'' w$, а значит $\nu X\beta' YDw$ — тоже r-форма (см. рис.6), где $D \to \beta''$ — некоторое правило. Так как D — нетерминал, имеем $Y \lessdot D$ или $Y \doteq D$. Так как β'' — самая правая перед w основа r-формы $\nu X\beta w = \nu X\beta' Y\beta'' w$, следовательно в рассматриваемом выводе $\mu Cw \Rightarrow \nu X\beta w$ применялось правило $C \to \delta Y\beta''$, но тогда это противоречит тому в СП-грамматике есть правило $D \to \beta''$.

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. Пусть правый конец основы β'' r-формы $\nu X \beta w$ находится непосредственно справа или слева от левого конца цепочки w, тогда предыдущая r-форма в правом выводе не могла бы заканчиваться на Cw (см.рис. 4-5). Приходим к противоречию.
  2. Пусть основа r-формы является собственным суффиксом $\nu X \beta$, т.е., $\beta = \beta' Y \beta''$, β'' — основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, а значит $\nu X \beta' Y D w$ — тоже r-форма (см. рис.6), где $D \to \beta''$ — некоторое правило. Так как D — нетерминал, имеем $Y \lessdot D$ или $Y \doteq D$. 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 $C \to \delta Y \beta''$, но тогда это противоречит тому в СП-грамматике есть правило $D \to \beta''$.
  3. Основа β'' имеет суффикс $X \beta$ (см. рис.7), т.е. $\beta'' = \alpha X \beta$, но с одной стороны, в грамматике есть правила $B \to \beta$ и $(D \to \beta'') = (D \to \alpha X \beta)$, а с другой стороны, в ней нет правил вида $A \to \alpha X \beta$. Противоречие.

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. Пусть правый конец основы β'' r-формы $\nu X \beta w$ находится непосредственно справа или слева от левого конца цепочки w, тогда предыдущая r-форма в правом выводе не могла бы заканчиваться на Cw (см.рис. 4-5). Приходим к противоречию.
  2. Пусть основа r-формы является собственным суффиксом $\nu X \beta$, т.е., $\beta = \beta' Y \beta''$, β'' — основа r-формы $\nu X \beta w = \nu X \beta' Y \beta'' w$, а значит $\nu X \beta' Y D w$ — тоже r-форма (см. рис.6), где $D \to \beta''$ — некоторое правило. Так как D — нетерминал, имеем $Y \lessdot D$ или $Y \doteq D$. 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 $C \to \delta Y \beta''$, но тогда это противоречит тому в СП-грамматике есть правило $D \to \beta''$.
  3. Основа β'' имеет суффикс $X\beta$ (см. рис.7), т.е. $\beta'' = \alpha X \beta$, но с одной стороны, в грамматике есть правила $B \to \beta$ и $(D \to \beta'') = (D \to \alpha X \beta)$, а с другой стороны, в ней нет правил вида $A \to \alpha X \beta$. Противоречие.
- Следовательно, основа $\beta'' = \beta$. А в силу однозначности грамматики имеем $(D \to \beta'') = (B \to \beta)$. Замечание доказано.

Доказательство замечания 3. Иллюстрация

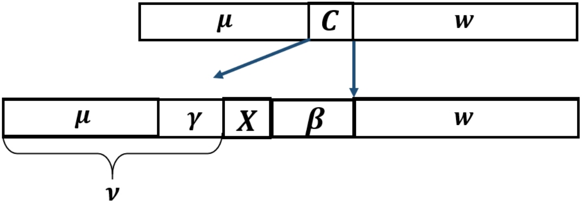

Рис. 3

Доказательство замечания 3. Иллюстрация

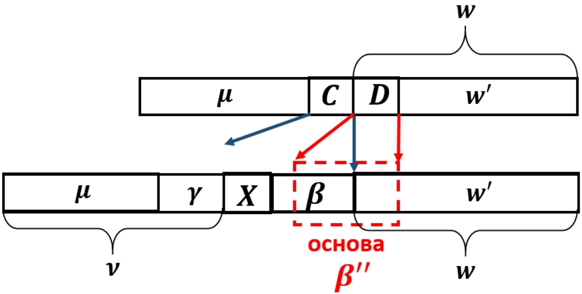

Рис. 4

Доказательство замечания 3. Иллюстрация

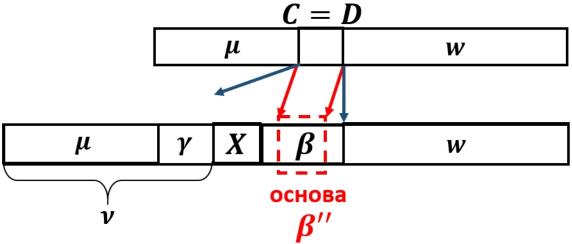

Рис. 5

Доказательство замечания 3. Иллюстрация

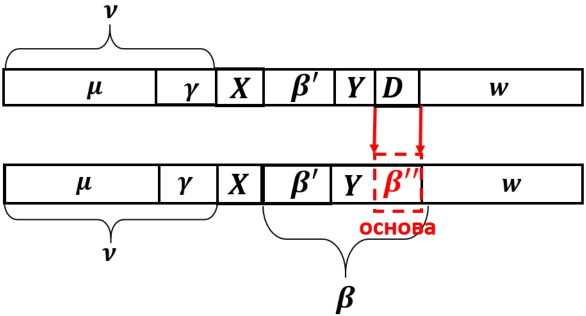

Рис. 6

Доказательство замечания 3. Иллюстрация

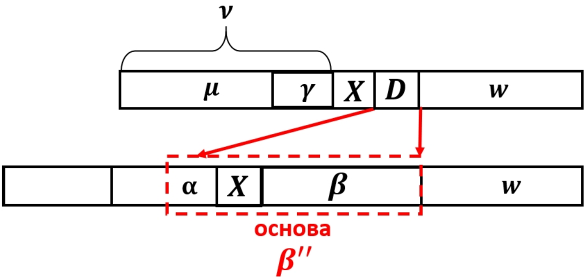

Рис. 7

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 $\vdash \gamma \dashv = X_0 X_1 ... X_n X_{n+1}$. Тогда основой формы γ является цепочка вида $X_{k_1} X_{k_1+1} \ldots X_{l-1} X_l$ такая, что

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

$$\ldots <\!\cdot X_{k_1} \doteq \ldots \doteq X_{k_2-1} \overset{<\cdot}{\doteq} X_{k_2} \doteq \ldots \doteq \overset{<\cdot}{\doteq} X_{k_s} \doteq \ldots \doteq X_l \cdot\!> X_{l+1} \ldots$$

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

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 $X_l \cdot\!> X_{l+1}$, $1 \leq k_1 \leq l \leq n$;
2. имеет место следующие отношения

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

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 $X_{k_i}X_{k_i+1} \dots X_{l-1}X_l$, $1 \leq k_i \leq l \leq n$.

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

```
$\gamma = \vdash w \dashv$: ($\gamma$ — 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

```
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
```

Алгоритм восходящего анализа для СП-грамматики

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

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

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

```
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
    $n =$длина $\gamma$
    $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
    print($\gamma$) (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 $w$
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
    $n =$длина $\gamma$
    $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
    print($\gamma$) (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

```
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print($\gamma$) (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
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2 - 1} \overset{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \overset{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$ )
```

Алгоритм восходящего анализа для СП-грамматики

```
Вход: СП-грамматика $G = (\Sigma, \Gamma, P, S)$ и цепочка $w \in \Sigma^*$.
Выход: Праволинейный вывод слова $w$
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n$ =длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print($\gamma$) (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
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2 - 1} \overset{<\cdot}{\underset{\doteq}{\leq}} X_{k_2} \doteq \dots \doteq \overset{<\cdot}{\underset{\doteq}{\leq}} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$ )
    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

```
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \dots X_n X_{n+1}$ ($X_0 < \cdot X_1$)
  print($\gamma$) (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
$\dots < \cdot X_{k_1} \doteq \dots \doteq X_{k_2 - 1} \overset{<\cdot}{\doteq} X_{k_2} \doteq \dots \doteq \overset{<\cdot}{\doteq} X_{k_s} \doteq \dots \doteq X_l \cdot > X_{l+1} \dots$ )
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$') exit
```

Алгоритм восходящего анализа для СП-грамматики

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

Алгоритм восходящего анализа для СП-грамматики

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

Алгоритм восходящего анализа для СП-грамматики

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

Алгоритм восходящего анализа для СП-грамматики

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

Выход: Праволинейный вывод слова w

```
$\gamma = \vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq \vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0 X_1 X_2 \ldots X_n X_{n+1}$ ($X_0 \lessdot X_1$)
  print($\gamma$) (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
$X_k \ldots X_l = X_{k_1} \ldots X_l$ такой, что
$\ldots \lessdot X_{k_1} \doteq \ldots \doteq X_{k_2-1} \overset{<\cdot}{\doteq} X_{k_2} \doteq \ldots \doteq \overset{<\cdot}{\doteq} X_{k_s} \doteq \ldots \doteq X_l \gtrdot X_{l+1} \ldots$ )
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$')  exit
    case ($X_i \lessdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки
$X_k \ldots X_l = X_{k_1} \ldots X_l$)
      $k = i + 1$
    case ($X_i \gtrdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем правый конец $X_l = X_i$ цепочки
$X_k \ldots X_l = X_{k_1} \ldots X_l$
      $l = i$
    (else: $X_i \doteq X_{i+1}$ or $\overset{<\cdot}{\doteq}$)
```

Алгоритм восходящего анализа для СП-грамматики

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

Выход: Праволинейный вывод слова w

```
$\gamma =\vdash w \dashv$: ($\gamma$ — текущая $r$-форма)
while ($\gamma \neq\vdash S \dashv$): (пока не найден весь вывод)
  $n =$длина $\gamma$
  $\gamma = X_0X_1X_2\ldots X_nX_{n+1}$ ($X_0 < \cdot X_1$)
  print($\gamma$) (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 $X_k \ldots X_l = X_{k_1} \ldots X_l$ такой, что $\ldots < \cdot X_{k_1} \doteq \ldots \doteq X_{k_2-1} \underset{\doteq}{<\cdot} X_{k_2} \doteq \ldots \doteq \underset{\doteq}{<\cdot} X_{k_s} \doteq \ldots \doteq X_l \cdot > X_{l+1} \ldots$ )
    case ($X_i \circ X_{i+1}$): (два соседних символа не сравнимы)
      print('$w \notin L(G)$') exit
    case ($X_i <\cdot X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем левый конец $X_k = X_{i+1}$ цепочки $X_k \ldots X_l = X_{k_1} \ldots X_l$)
      $k = i + 1$
    case ($X_i \cdot> X_{i+1}$) and not($X_i \doteq X_{i+1}$): (найдем правый конец $X_l = X_i$ цепочки $X_k \ldots X_l = X_{k_1} \ldots X_l$
      $l = i$
    (else: $X_i \doteq X_{i+1}$ or $\underset{\doteq}{<\cdot}$ )
    $i = +1$ (двигаемся дальше, пока не найдем правый и левый конец цепочки $X_k \ldots X_l = X_{k_1} \ldots X_l$)
```

Алгоритм восходящего анализа для СП-грамматики (продолжение)

```
if $l = 0$ (не найден правый конец основы)
  print('$w \notin L(G)$')   exit
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 $X_j < \cdot X_{j+1}$, а выполняется только равенство $X_j \doteq X_{j+1}$
      print('$w \notin L(G)$')   exit
```

Алгоритм восходящего анализа для СП-грамматики (продолжение)

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

$$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot \boxed{B} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot \boxed{A} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$$

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.

$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{B} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{A} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot S \doteq + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$ 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.

$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{B} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{A} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot S \doteq + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$ нет основы $S+x$, зато есть основа x

$\vdash \lessdot S \doteq + \lessdot \boxed{B} \gtrdot * \lessdot x \gtrdot \dashv$ нет основы $S+B$, зато есть основа B

$\vdash \lessdot S \doteq + \lessdot A \doteq * \lessdot \boxed{x} \gtrdot \dashv$ 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.

$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{B} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{A} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot S \doteq + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$ нет основы $S+x$, зато есть основа x

$\vdash \lessdot S \doteq + \lessdot \boxed{B} \gtrdot * \lessdot x \gtrdot \dashv$ нет основы $S+B$, зато есть основа B

$\vdash \lessdot S \doteq + \lessdot A \doteq * \lessdot \boxed{x} \gtrdot \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash \lessdot S \doteq + \lessdot \boxed{A \doteq * \lessdot B} \gtrdot \dashv$ 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.

$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{B} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot \boxed{A} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot S \doteq + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$ нет основы $S+x$, зато есть основа x

$\vdash \lessdot S \doteq + \lessdot \boxed{B} \gtrdot * \lessdot x \gtrdot \dashv$ нет основы $S+B$, зато есть основа B

$\vdash \lessdot S \doteq + \lessdot A \doteq * \lessdot \boxed{x} \gtrdot \dashv$ нет основ $S+A*x$, $A*x$, но есть основа x

$\vdash \lessdot S \doteq + \lessdot \boxed{A \doteq * \lessdot B} \gtrdot \dashv$ нет основы $S+A*B$, но есть основа $A*B$

$\vdash \lessdot \boxed{S \doteq + \lessdot A} \gtrdot \dashv$

$\vdash \lessdot S \gtrdot \dashv$

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
$S \rightarrow S + A \mid A,\ A \rightarrow A * B \mid B,\ B \rightarrow x \mid (S)$ 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.

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.

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

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.

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

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.

- Пусть $S = \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_n = w$ — правосторонний вывод.
- И пусть без ограничения общности терминал a пhинадлежит некоторой основе γ, т.е. $\alpha_i = uaSbv$ 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. $+ \lessdot x$, 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. $+ \lessdot x$, поскольку основа-операнд x сворачивается раньше оператора $+$, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \gtrdot +$ 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. $+ \lessdot x$, поскольку основа-операнд x сворачивается раньше оператора $+$, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \gtrdot +$ так как ввиду левоассоциативности $+$ в слове $+S+$ префикс $+S$ сворачивается раньше суффикса $+$, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot *$, 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. $+ \lessdot x$, поскольку основа-операнд x сворачивается раньше оператора $+$, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \gtrdot +$ так как ввиду левоассоциативности $+$ в слове $+S+$ префикс $+S$ сворачивается раньше суффикса $+$, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot *$, поскольку приоритет $*$ выше $+$: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.
  4. $+ \lessdot ($, 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. $+ \lessdot x$, поскольку основа-операнд x сворачивается раньше оператора $+$, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \gtrdot +$ так как ввиду левоассоциативности $+$ в слове $+S+$ префикс $+S$ сворачивается раньше суффикса $+$, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot *$, поскольку приоритет $*$ выше $+$: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.
  4. $+ \lessdot ($, поскольку приоритет скобки $($ выше всех операторов: в слове $+($ суффикс $($ сворачивается раньше префикса $+$, а слово $+S($ не встречается ни в одной форме.
  5. $+ \gtrdot)$: 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. $+ \lessdot x$, поскольку основа-операнд x сворачивается раньше оператора $+$, а слово $+Sx$ не встречается ни в одной форме.
  2. $+ \gtrdot +$ так как ввиду левоассоциативности $+$ в слове $+S+$ префикс $+S$ сворачивается раньше суффикса $+$, а слово $++$ не встречается ни в одной форме.
  3. $+ \lessdot *$, поскольку приоритет $*$ выше $+$: в слове $+S*$ суффикс $S*$ сворачивается раньше префикса $+S$, а слово $+*$ не встречается ни в одной форме.
  4. $+ \lessdot ($, поскольку приоритет скобки $($ выше всех операторов: в слове $+($ суффикс $($ сворачивается раньше префикса $+$, а слово $+S($ не встречается ни в одной форме.
  5. $+ \gtrdot)$: в слове $+S)$ префикс $+S$ сворачивается раньше скобки $)$, а слово $+)$ не встречается ни в одной форме.
  6. $+ \gtrdot \dashv$, 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: $* \lessdot x$, $* \gtrdot *$, $+ \lessdot *$, $* \lessdot ($, $* \gtrdot)$, $* \gtrdot \dashv$.
- Приоритеты для терминала скобка (расставляются естественным образом.

Пример 8. Приоритеты для терминалов $*$ и (

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

$$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot_s + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot_s + \lessdot_s * \boxed{x} \gtrdot \dashv$$

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.

$$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot_s + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot_s + \lessdot_s * \boxed{x} \gtrdot \dashv$$

$$\vdash \lessdot_s + \boxed{\lessdot_s * \gtrdot_s} \dashv$$

$$\vdash \boxed{\lessdot_s + \gtrdot_s} \dashv$$

$$\vdash S \dashv$$

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.

$$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot_s + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$$

$$\vdash \lessdot_s + \lessdot_s * \boxed{x} \gtrdot \dashv$$

$$\vdash \lessdot_s + \boxed{\lessdot_s * \gtrdot_s} \dashv$$

$$\vdash \boxed{\lessdot_s + \gtrdot_s} \dashv$$

$$\vdash S \dashv$$

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.

$\vdash \lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot_S + \lessdot \boxed{x} \gtrdot * \lessdot x \gtrdot \dashv$

$\vdash \lessdot_S + \lessdot_S * \boxed{x} \gtrdot \dashv$

$\vdash \lessdot_S + \boxed{\lessdot_S * \gtrdot_S} \dashv$

$\vdash \boxed{\lessdot_S + \gtrdot_S} \dashv$

$\vdash S \dashv$

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, вставить $+$;
- e_2: лишняя (, удалить её;
- e_3: лишняя), удалить её.

Имеем следующий обработчик ошибок.

	x	$+$	$*$	(	)	$\dashv$
x	e_1	$\gtrdot$	$\gtrdot$		$\gtrdot$	$\gtrdot$
$+$	$\lessdot$	$\gtrdot$	$\lessdot$	$\lessdot$	$\gtrdot$	$\gtrdot$
$*$	$\lessdot$	$\gtrdot$	$\gtrdot$	$\lessdot$	$\gtrdot$	$\gtrdot$
(	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\doteq$	e_2
)	e_1	$\gtrdot$	$\gtrdot$	e_1	$\gtrdot$	$\gtrdot$
$\vdash$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	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.

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

$\vdash \lessdot x e_1 x \gtrdot + \gtrdot \dashv$ ошибка e_1 в позиции 3 исходной цепочки

$\vdash \lessdot \boxed{x} \gtrdot + \gtrdot \; x \gtrdot + \gtrdot \dashv$

$\vdash \; \lessdot_s + \lessdot \boxed{x} \gtrdot + \gtrdot \dashv$

$\vdash \; \boxed{\lessdot_s + \gtrdot_s} + \gtrdot \dashv$

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.

$\vdash e_3) \cdot> x \quad x \cdot> + \cdot> \dashv$ 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

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

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

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

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

$\vdash S \dashv$

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.

$\vdash e_3)\cdot> x \quad x\cdot>+\cdot>\dashv$ 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

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

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

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

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

$\vdash S \dashv$

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

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

$$\vdash <\cdot (<\cdot \boxed{x} \cdot> + <\cdot x \cdot>) \quad x \cdot> \dashv$$

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

$$\vdash <\cdot (<\cdot \boxed{x} \cdot> + <\cdot x \cdot>) \quad x \cdot> \dashv$$

$$\vdash <\cdot (<\cdot_s + <\cdot \boxed{x} \cdot>) \quad x \cdot> \dashv$$

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

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

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

$$\vdash <\cdot\boxed{(\doteq_s)}\ \ x\cdot>\dashv$$

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

$$\vdash \lessdot (\lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot) \quad x \gtrdot \dashv$$

$$\vdash \lessdot (\lessdot_s + \lessdot \boxed{x} \gtrdot) \quad x \gtrdot \dashv$$

$$\vdash \lessdot \boxed{(\doteq_s)} \quad x \gtrdot \dashv$$

$$\vdash \boxed{\lessdot_s x} \gtrdot \dashv$$

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

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

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

$$\vdash <\cdot \boxed{(\doteq_s)} \;\; x \cdot> \dashv$$

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

$$\vdash S \dashv$$

Пример 11. Обработка ошибочной цепочки $w = (x + x)x$ для грамматики из примера 8

Пример 11. Рассмотрим обработку ошибочной цепочки $w = (x + x)x$ для операторной грамматики из примера 8.

$$\vdash \lessdot (\lessdot \boxed{x} \gtrdot + \lessdot x \gtrdot)\ \ x \gtrdot \dashv$$

$$\vdash \lessdot (\lessdot_s + \lessdot \boxed{x} \gtrdot)\ \ x \gtrdot \dashv$$

$$\vdash \lessdot \boxed{(\doteq_s)}\ \ x \gtrdot \dashv$$

$$\vdash \boxed{\lessdot_s x} \gtrdot \dashv$$

$$\vdash S \dashv$$

Мы видим, что наш обработчик ошибок не смог справиться и с этой ошибочной цепочкой и не нашел последнюю ошибку.