Лингвистические основы информатики
Лекция 14
Множество выбора для правила, LL(1)-грамматики

Ю. В. Нагребецкая, И. А. Михайлова,

Уральский федеральный университет
Институт естественных наук и математики
Департамент математики, механики и компьютерных наук
Направления: Математика и компьютерные науки
Компьютерная безопасность
(6 семестр)

Пример 1. Грамматика

Рассмотрим грамматику

$$S \to aA \mid b$$
$$A \to bSA \mid a$$

из лекции 13 и рассмотрим соответствующий ДМПА, построенный в той же лекции (см. след. слайд).

Пример 1

	a	b	$\dashv$
S	$(aA, \_), (b, \_)$	$(aA, \_), (b, \_)$	$(aA, \_), (b, \_)$
A	$(bSA, \_), (a, \_)$	$(bSA, \_), (a, \_)$	$(bSA, \_), (a, \_)$
a	$(\varepsilon, \rightarrow)$		
b		$(\varepsilon, \rightarrow)$	
∇			$\vee$

В дальнейшем будем записывать протокол (последовательность конфигураций автомата) в виде таблицы без дополнительных символов ∇ и $\dashv$, вершина стека находится слева, и вместо позиций указателя будем приводить только необработанную чать входной цепочки. Наша цель — попытаться построить детерминированный автомат, который распознает тот же язык.

Пример 1 (продолжение)

	стек	необработанная часть цепочки
1	S	abba
2	aA	abba
3	A	bba
4	bSA	bba
5	SA	ba
6	bA	ba
7	A	a
8	a	a
9	∇	$\dashv$

Пример 1 (продолжение)

- Правила вывода применяются на 1,3,5,7 шагах. Можно ли на 7 шаге применить вместо $A \to a$ правило вывода $A \to bSA$?

Пример 1 (продолжение)

- Правила вывода применяются на 1,3,5,7 шагах. Можно ли на 7 шаге применить вместо $A \to a$ правило вывода $A \to bSA$?
- Что будет, если в стеке остался хотя бы один символ (терминал или нетерминал), а входная цепочка закончилась?

Пример 1 (продолжение)

- Правила вывода применяются на 1,3,5,7 шагах. Можно ли на 7 шаге применить вместо $A \to a$ правило вывода $A \to bSA$?
- Что будет, если в стеке остался хотя бы один символ (терминал или нетерминал), а входная цепочка закончилась?
- Обратите внимание, что если на некотором шаге в стеке находится α, а u — необработанная часть цепочки, то $\alpha \Rightarrow^* u$ (см. лекцию №13).

Пример 1 (продолжение)

- Правила вывода применяются на 1,3,5,7 шагах. Можно ли на 7 шаге применить вместо $A \to a$ правило вывода $A \to bSA$?
- Что будет, если в стеке остался хотя бы один символ (терминал или нетерминал), а входная цепочка закончилась?
- Обратите внимание, что если на некотором шаге в стеке находится α, а u — необработанная часть цепочки, то $\alpha \Rightarrow^* u$ (см. лекцию №13).
- Как построить детерминированный МП-автомат, распознающий тот же язык, что и исходный МП-автомат?

Разделенные грамматики

- КС грамматики, в которых каждое правило имеет вид $A \to a\beta$, где $a \in \Sigma$, $\beta \in (\Sigma \cup \Gamma)^*$, называются разделенными.

Разделенные грамматики

- КС грамматики, в которых каждое правило имеет вид $A \to a\beta$, где $a \in \Sigma$, $\beta \in (\Sigma \cup \Gamma)^*$, называются разделенными.

Лемма 1 о разделенных грамматиках

По НМПА $\mathcal{M}$, построенному по разделенной грамматике G, можно построить ДМПА $\mathcal{M}'$, распознающий тот же язык, тогда и только тогда, когда для любых правил $A \to a\beta$, $A \to b\gamma$, где $a, b \in \Sigma$, $\beta, \gamma \in (\Sigma \cup \Gamma)^*$, выполняется $a \neq b$ (или, что тоже самое, ни для каких двух правил правые части не имеют нетривиального общего префикса).

Лемма 2

Для того, чтобы доказать лемму 1, сформулируем и докажем (упр.) лемму 2.

Лемма 2

Для того, чтобы доказать лемму 1, сформулируем и докажем (упр.) лемму 2.

Лемма 2

В НМПА (в частности, в ДМПА), построенному по грамматике G (не обязательно разделенной) можно удалить переходы вида $\delta(A, a) = (b\alpha, \_)$, где $a, b \in \Sigma$, $\alpha \in (\Sigma \cup \Gamma)^*$ и $a \neq b$. Полученный НМПА (ДМПА) будет распознавать тот же язык, что и исходный НМПА (ДМПА).

Лемма 2

Для того, чтобы доказать лемму 1, сформулируем и докажем (упр.) лемму 2.

Лемма 2

В НМПА (в частности, в ДМПА), построенному по грамматике G (не обязательно разделенной) можно удалить переходы вида $\delta(A, a) = (b\alpha, \_)$, где $a, b \in \Sigma$, $\alpha \in (\Sigma \cup \Gamma)^*$ и $a \neq b$. Полученный НМПА (ДМПА) будет распознавать тот же язык, что и исходный НМПА (ДМПА).

Доказательство. — упр.
Указание: покажите, что в процессе работы данного НМПА $\mathcal{M}$ ни в одном выводе цепочки $w \in L(\mathcal{M})$ не может быть конфигурации $[bu, av]$, где $u, v \in (\Sigma \cup \Gamma)^*$.

- **Доказательство леммы 2.**

Доказательство леммы 2 о разделенных грамматиках

- **Доказательство леммы 2**.
- Пусть грамматика G — разделенная. Построим ДМПА $\mathcal{M}'$ как в теореме 2 лекции 13.

Доказательство леммы 2 о разделенных грамматиках

- **Доказательство леммы 2**.
- Пусть грамматика G — разделенная. Построим ДМПА $\mathcal{M}'$ как в теореме 2 лекции 13.
- Для любых $A \in \Gamma$, $a \in \Sigma$ из множества $\delta(A, a)$ этого ДМПА (напомним, что для ДМПА $\mathcal{M}'$ бинарное отношение δ вообще не является частично рекурсивной функцией) удалим все правила $A \to b\beta$, $b \in \Sigma \backslash a$.

Доказательство леммы 2 о разделенных грамматиках

- **Доказательство леммы 2.**
- Пусть грамматика G — разделенная. Построим ДМПА $\mathcal{M}'$ как в теореме 2 лекции 13.
- Для любых $A \in \Gamma$, $a \in \Sigma$ из множества $\delta(A, a)$ этого ДМПА (напомним, что для ДМПА $\mathcal{M}'$ бинарное отношение δ вообще не является частично рекурсивной функцией) удалим все правила $A \to b\beta$, $b \in \Sigma \backslash a$.
- Тогда для любых $A \in \Gamma$, $a \in \Sigma$ множество $\delta(A, a)$ станет не более, чем одноэлементным, и полученный МП-автомат станет детерминированным.

Доказательство леммы 2 о разделенных грамматиках

- **Доказательство леммы 2.**
- Пусть грамматика G — разделенная. Построим ДМПА $\mathcal{M}'$ как в теореме 2 лекции 13.
- Для любых $A \in \Gamma$, $a \in \Sigma$ из множества $\delta(A, a)$ этого ДМПА (напомним, что для ДМПА $\mathcal{M}'$ бинарное отношение δ вообще не является частично рекурсивной функцией) удалим все правила $A \to b\beta$, $b \in \Sigma \backslash a$.
- Тогда для любых $A \in \Gamma$, $a \in \Sigma$ множество $\delta(A, a)$ станет не более, чем одноэлементным, и полученный МП-автомат станет детерминированным.
- Покажите, что $L(\mathcal{M}') = L(\mathcal{M})$, опираясь на лемму 1.

Доказательство леммы 2 о разделенных грамматиках

- **Доказательство леммы 2**.
- Пусть грамматика G — разделенная. Построим ДМПА $\mathcal{M}'$ как в теореме 2 лекции 13.
- Для любых $A \in \Gamma$, $a \in \Sigma$ из множества $\delta(A, a)$ этого ДМПА (напомним, что для ДМПА $\mathcal{M}'$ бинарное отношение δ вообще не является частично рекурсивной функцией) удалим все правила $A \to b\beta$, $b \in \Sigma \backslash a$.
- Тогда для любых $A \in \Gamma$, $a \in \Sigma$ множество $\delta(A, a)$ станет не более, чем одноэлементным, и полученный МП-автомат станет детерминированным.
- Покажите, что $L(\mathcal{M}') = L(\mathcal{M})$, опираясь на лемму 1.

Грамматика из примера 1, очевидно, является разделенной, и удовлетворяет условию леммы. Следовательно, по ней можно построить ДМПА по только что приведенному доказательству (см. след. слайд).

ДМПА для примера 1

	a	b	$\dashv$
S	$(aA, \_)$	$(b, \_)$	
A	$(a, \_)$	$(bSA, \_)$	
a	$(\varepsilon, \rightarrow)$		
b		$(\varepsilon, \rightarrow)$	
∇			$\vee$

	стек	необработанная часть цепочки
1	S	abba
2	aA	abba
3	A	bba
4	bSA	bba
5	SA	ba
6	bA	ba
7	A	a
8	a	a
9	∇	$\dashv$

Упрощение ДМПА $\mathcal{M}'$ из примера 1

- Построенный в примере 1 ДМПА $\mathcal{M}'$ можно упростить: посмотрите например на шаги 1-3. На шаге 1 нетерминал S заменяется на aA (причем a окажется на вершине стека), а затем на шаге 2 этот терминал удаляется из стека и происходит сдвиг во входной цепочке.

Упрощение ДМПА $\mathcal{M}'$ из примера 1

- Построенный в примере 1 ДМПА $\mathcal{M}'$ можно упростить: посмотрите например на шаги 1-3. На шаге 1 нетерминал S заменяется на aA (причем a окажется на вершине стека), а затем на шаге 2 этот терминал удаляется из стека и происходит сдвиг во входной цепочке.
- Шаг 2 можно пропустить, если изменить команду $\delta(S, a)$: вместо цепочки aA втолкнуть A и сразу сдвинуться вправо во входной строке. Закончите модифицировать управляющую таблицу.

Упрощение ДМПА $\mathcal{M}'$ из примера 1

- Построенный в примере 1 ДМПА $\mathcal{M}'$ можно упростить: посмотрите например на шаги 1-3. На шаге 1 нетерминал S заменяется на aA (причем a окажется на вершине стека), а затем на шаге 2 этот терминал удаляется из стека и происходит сдвиг во входной цепочке.
- Шаг 2 можно пропустить, если изменить команду $\delta(S, a)$: вместо цепочки aA втолкнуть A и сразу сдвинуться вправо во входной строке. Закончите модифицировать управляющую таблицу.
- Обратите внимание, что в полученном автомате терминалы никогда не попадают в стек, поэтому таблица становится еще более компактной.

Разделенные грамматики

	a	b	⊣
S	$(aA, \_)$	$(b, \_)$	
A	$(a, \_)$	$(bSA, \_)$	
a	$(\varepsilon, \rightarrow)$		
b		$(\varepsilon, \rightarrow)$	
∇			$\vee$

	стек	необработанная часть цепочки
1	S	abba
2	aA	abba
3	A	bba
4	bSA	bba
5	SA	ba
6	bA	ba
7	A	a
8	a	a
9	∇	⊣

Разделенные грамматики

	a	b	⊣
S	$(A, \rightarrow)$	$(b, \_)$	
A	$(a, \_)$	$(bSA, \_)$	
a	$(\varepsilon, \rightarrow)$		
b		$(\varepsilon, \rightarrow)$	
∇			$\vee$

	стек	необработанная часть цепочки
1	S	abba
3	A	bba
4	bSA	bba
5	SA	ba
6	bA	ba
7	A	a
8	a	a
9	∇	⊣

Разделенные грамматики

	a	b	$\dashv$
S	$(A, \rightarrow)$	$(\varepsilon, \rightarrow)$	
A	$(\varepsilon, \rightarrow)$	$(SA, \rightarrow)$	
∇			$\vee$

	стек	необработанная часть цепочки
1	S	abba
3	A	bba
5	SA	ba
7	A	a
9	∇	$\dashv$

Упрощение ДМПА в общем случае

ДМПА, построенный по грамматике G можно упростить таким же образом и в общем случае.

Упрощение ДМПА в общем случае

ДМПА, построенный по грамматике G можно упростить таким же образом и в общем случае.

Лемма 3 об упрощениии ДМПА

Пусть $\mathcal{M}$ — НМПА, построенный по грамматике G (не обязательно разделенной). И пусть $\mathcal{M}'$ — ДМПА, полученный из НМПА $\mathcal{M}$ удалением некоторых элементов из множеств $\delta(A, a)$, и при этом $L(\mathcal{M}') = L(\mathcal{M})$. Тогда существует ДМПА $\mathcal{M}''$ порождающий тот же язык, что и ДМПА $\mathcal{M}'$, при этом терминалы никогда не попадают в его стек.

Лемма 3 об упрощениии ДМПА. Доказательство

- **Доказательство**.

- **Доказательство**.
- По лемме 2 можно считать, в автомате $\mathcal{M}'$ могут быть только переходы $\delta(A, a) = (a\alpha, \_)$, где $a \in \Sigma$, $\alpha \in (\Sigma \cup \Gamma)^*$. Для каждого $a \in \Sigma$ будем удалять из таблицы переходов автомата $\mathcal{M}'$ команду $\delta(A, a) = (a\alpha, \_)$, пока такая есть, заменяя ее командой $\delta(A, a) = (\alpha, \rightarrow)$. А затем удалим команду $\delta(a, a) = (\varepsilon, \rightarrow)$.

Лемма 3 об упрощениии ДМПА. Доказательство

- **Доказательство**.
- По лемме 2 можно считать, в автомате $\mathcal{M}'$ могут быть только переходы $\delta(A, a) = (a\alpha, \_)$, где $a \in \Sigma$, $\alpha \in (\Sigma \cup \Gamma)^*$. Для каждого $a \in \Sigma$ будем удалять из таблицы переходов автомата $\mathcal{M}'$ команду $\delta(A, a) = (a\alpha, \_)$, пока такая есть, заменяя ее командой $\delta(A, a) = (\alpha, \rightarrow)$. А затем удалим команду $\delta(a, a) = (\varepsilon, \rightarrow)$.
- Легко понять, что полученный в результате ДМПА $\mathcal{M}''$ порождает тот же язык, что и ДМПА $\mathcal{M}'$, т.е. язык $L(G)$.

Лемма 3 об упрощениии ДМПА. Доказательство

- **Доказательство**.
- По лемме 2 можно считать, в автомате $\mathcal{M}'$ могут быть только переходы $\delta(A, a) = (a\alpha, \_)$, где $a \in \Sigma$, $\alpha \in (\Sigma \cup \Gamma)^*$. Для каждого $a \in \Sigma$ будем удалять из таблицы переходов автомата $\mathcal{M}'$ команду $\delta(A, a) = (a\alpha, \_)$, пока такая есть, заменяя ее командой $\delta(A, a) = (\alpha, \rightarrow)$. А затем удалим команду $\delta(a, a) = (\varepsilon, \rightarrow)$.
- Легко понять, что полученный в результате ДМПА $\mathcal{M}''$ порождает тот же язык, что и ДМПА $\mathcal{M}'$, т.е. язык $L(G)$.
- В полученном автомате $\mathcal{M}''$ терминалы никогда не попадают в стек.

Определение множеств FIRST и FOLLOW

Разделенные грамматики встречаются довольно редко. Можно значительно расширить этот класс, чтобы можно было аналогичным образом построить ДМПА, распознающий язык $L(G)$.

Определение множеств FIRST и FOLLOW

Разделенные грамматики встречаются довольно редко. Можно значительно расширить этот класс, чтобы можно было аналогичным образом построить ДМПА, распознающий язык $L(G)$.

В предыдущем примере мы выбирали команду $\delta(A, a) = (\gamma, \_)$ (для правила $A \to \gamma$), если γ начиналась с a. В общем случае правая часть правила вывода может начинаться с нетерминала и тогда следует смотреть на первые символы строк, которые можем вывести из γ.

Определение множеств FIRST и FOLLOW

Разделенные грамматики встречаются довольно редко. Можно значительно расширить этот класс, чтобы можно было аналогичным образом построить ДМПА, распознающий язык $L(G)$.

В предыдущем примере мы выбирали команду $\delta(A, a) = (\gamma, \_)$ (для правила $A \to \gamma$), если γ начиналась с a. В общем случае правая часть правила вывода может начинаться с нетерминала и тогда следует смотреть на первые символы строк, которые можем вывести из γ.

Введем понятия множеств *FIRST* и *FOLLOW*.

Определение множеств FIRST и FOLLOW

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$.
  Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что

Определение множеств FIRST и FOLLOW

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$.
  Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.

Определение множеств FIRST и FOLLOW

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$. Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.
  2. $\varepsilon \in FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* \varepsilon$.

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$.
Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.
  2. $\varepsilon \in FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* \varepsilon$.

Предположим теперь, что $A\beta \Rightarrow^* av$, и в этом выводе применяются правила $A \to \gamma$, $\gamma \Rightarrow^* \varepsilon$. Тогда $A\beta \Rightarrow \underline{\gamma}\beta \Rightarrow^* \beta \Rightarrow \underline{av}$. Поменяем порядок вывода и получим $A\beta \Rightarrow^* A\underline{av}$. Символ a стоит после A.

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$. Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.
  2. $\varepsilon \in FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* \varepsilon$.

Предположим теперь, что $A\beta \Rightarrow^* av$, и в этом выводе применяются правила $A \to \gamma$, $\gamma \Rightarrow^* \varepsilon$. Тогда $A\beta \Rightarrow \underline{\gamma}\beta \Rightarrow^* \beta \Rightarrow \underline{av}$. Поменяем порядок вывода и получим $A\beta \Rightarrow^* A\underline{av}$. Символ a стоит после A.

- Пусть $A \in \Gamma$. Множеством $FOLLOW(A)$ назовем такое подмножество множества $\Sigma \cup \{\dashv\}$, что

Определение множеств FIRST и FOLLOW

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$.
  Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.
  2. $\varepsilon \in FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* \varepsilon$.

Предположим теперь, что $A\beta \Rightarrow^* av$, и в этом выводе применяются правила $A \to \gamma$, $\gamma \Rightarrow^* \varepsilon$. Тогда $A\beta \Rightarrow \underline{\gamma}\beta \Rightarrow^* \beta \Rightarrow \underline{av}$. Поменяем порядок вывода и получим $A\beta \Rightarrow^* A\underline{av}$. Символ a стоит после A.

- Пусть $A \in \Gamma$. Множеством $FOLLOW(A)$ назовем такое подмножество множества $\Sigma \cup \{\dashv\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FOLLOW(A)$, если и только если $S \Rightarrow^* \alpha A a \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$.

Определение множеств FIRST и FOLLOW

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$.
  Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.
  2. $\varepsilon \in FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* \varepsilon$.

Предположим теперь, что $A\beta \Rightarrow^* av$, и в этом выводе применяются правила $A \to \gamma$, $\gamma \Rightarrow^* \varepsilon$. Тогда $A\beta \Rightarrow \underline{\gamma}\beta \Rightarrow^* \beta \Rightarrow \underline{av}$. Поменяем порядок вывода и получим $A\beta \Rightarrow^* A\underline{av}$. Символ a стоит после A.

- Пусть $A \in \Gamma$. Множеством $FOLLOW(A)$ назовем такое подмножество множества $\Sigma \cup \{\dashv\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FOLLOW(A)$, если и только если $S \Rightarrow^* \alpha A a \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$.
  2. $\dashv \in FOLLOW(A)$, если и только если $S \Rightarrow^* \alpha A$.

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $\gamma \in (\Sigma \cup \Gamma)^*$. Множеством $FIRST(\gamma)$ назовем такое подмножество множества $\Sigma \cup \{\varepsilon\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* a\gamma'$ для некоторого $\gamma' \in (\Sigma \cup \Gamma)^*$.
  2. $\varepsilon \in FIRST(\gamma)$, если и только если $\gamma \Rightarrow^* \varepsilon$.

Предположим теперь, что $A\beta \Rightarrow^* av$, и в этом выводе применяются правила $A \to \gamma$, $\gamma \Rightarrow^* \varepsilon$. Тогда $A\beta \Rightarrow \underline{\gamma}\beta \Rightarrow^* \beta \Rightarrow \underline{av}$. Поменяем порядок вывода и получим $A\beta \Rightarrow^* A\underline{av}$. Символ a стоит после A.

- Пусть $A \in \Gamma$. Множеством $FOLLOW(A)$ назовем такое подмножество множества $\Sigma \cup \{\dashv\}$, что
  1. терминал $a \in \Sigma$ принадлежит множеству $FOLLOW(A)$, если и только если $S \Rightarrow^* \alpha A a \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$.
  2. $\dashv \in FOLLOW(A)$, если и только если $S \Rightarrow^* \alpha A$.
  3. $\dashv \in FOLLOW(S)$, поскольку можно считать, что $S \Rightarrow^* S$.

Пример 1. Множество FIRST

Рассмотрим грамматику

$$\begin{aligned} S &\to CA \mid CB \\ A &\to bBC \mid cb \\ B &\to b \\ C &\to dC \mid \varepsilon \end{aligned}$$

Тогда

Пример 1. Множество FIRST

Рассмотрим грамматику

$$
\begin{aligned}
S &\to CA \mid CB \\
A &\to bBC \mid cb \\
B &\to b \\
C &\to dC \mid \varepsilon
\end{aligned}
$$

Тогда

- $FIRST(a) = \{a\}$ (так как $a \Rightarrow a$)

Пример 1. Множество FIRST

Рассмотрим грамматику

$$
\begin{aligned}
S &\to CA \mid CB \\
A &\to bBC \mid cb \\
B &\to b \\
C &\to dC \mid \varepsilon
\end{aligned}
$$

Тогда

- $FIRST(a) = \{a\}$ (так как $a \Rightarrow a$)
- $FIRST(\varepsilon) = \{\varepsilon\}$ (так как $\varepsilon \Rightarrow \varepsilon$)

Пример 1. Множество FIRST

Рассмотрим грамматику

$$\begin{aligned} S &\to CA \mid CB \\ A &\to bBC \mid cb \\ B &\to b \\ C &\to dC \mid \varepsilon \end{aligned}$$

Тогда

- $FIRST(a) = \{a\}$ (так как $a \Rightarrow a$)
- $FIRST(\varepsilon) = \{\varepsilon\}$ (так как $\varepsilon \Rightarrow \varepsilon$)
- $FIRST(C) = \{d, \varepsilon\}$ (так как $C \Rightarrow dC$)

Пример 1. Множество FIRST

Рассмотрим грамматику

$$S \to CA \mid CB$$
$$A \to bBC \mid cb$$
$$B \to b$$
$$C \to dC \mid \varepsilon$$

Тогда

- $FIRST(a) = \{a\}$ (так как $a \Rightarrow a$)
- $FIRST(\varepsilon) = \{\varepsilon\}$ (так как $\varepsilon \Rightarrow \varepsilon$)
- $FIRST(C) = \{d, \varepsilon\}$ (так как $C \Rightarrow dC$)
- $FIRST(BC) = \{b\}$ (так как $BC \Rightarrow \underline{b}C$)

Пример 1. Множество FIRST

Рассмотрим грамматику

$$S \to CA \mid CB$$
$$A \to bBC \mid cb$$
$$B \to b$$
$$C \to dC \mid \varepsilon$$

Тогда

- $FIRST(a) = \{a\}$ (так как $a \Rightarrow a$)
- $FIRST(\varepsilon) = \{\varepsilon\}$ (так как $\varepsilon \Rightarrow \varepsilon$)
- $FIRST(C) = \{d, \varepsilon\}$ (так как $C \Rightarrow dC$)
- $FIRST(BC) = \{b\}$ (так как $BC \Rightarrow \underline{b}C$)
- $FIRST(AC) = \{b, c\}$ (так как $AC \Rightarrow \underline{bBC}C$ и $AC \Rightarrow \underline{cb}C$)

Пример 1. Множество FOLLOW

- $FOLLOW(S) = \{\dashv\}$

Пример 1. Множество FOLLOW

- $FOLLOW(S) = \{\dashv\}$
- $\dashv \in FOLLOW(A)$, $\dashv \in FOLLOW(B)$, $\dashv \in FOLLOW(C)$
  (так как $S \Rightarrow CA$, $S \Rightarrow CB$, $A \Rightarrow aBC$)

Пример 1. Множество FOLLOW

- $FOLLOW(S) = \{\dashv\}$
- $\dashv \in FOLLOW(A)$, $\dashv \in FOLLOW(B)$, $\dashv \in FOLLOW(C)$
  (так как $S \Rightarrow CA$, $S \Rightarrow CB$, $A \Rightarrow aBC$)
- $d \in FOLLOW(B)$
  (так как $S \Rightarrow \underline{CA} \Rightarrow \underline{aBC} \Rightarrow aB\underline{dC}$)

Построение множества FIRST

Вначале приведем алгоритм вычисления множеств FIRST для всех символов грамматики (терминалов и нетерминалов). Он основан на следующем замечании, вытекающем непосредственно из определения множества FIRST.

Построение множества FIRST

Вначале приведем алгоритм вычисления множеств FIRST для всех символов грамматики (терминалов и нетерминалов). Он основан на следующем замечании, вытекающем непосредственно из определения множества FIRST.

Замечание 1 о множестве FIRST для терминалов и нетерминалов

- Если a – терминал, то $FIRST(a) = \{a\}$. Если A – нетерминал, то

$$FIRST(A) = \bigcup_{A \to \beta} FIRST(\beta)$$

Множество FIRST состоит только из терминалов!

Построение множества FIRST

Вначале приведем алгоритм вычисления множеств FIRST для всех символов грамматики (терминалов и нетерминалов). Он основан на следующем замечании, вытекающем непосредственно из определения множества FIRST.

Замечание 1 о множестве FIRST для терминалов и нетерминалов

- Если a – терминал, то $FIRST(a) = \{a\}$. Если A – нетерминал, то
$$FIRST(A) = \bigcup_{A \to \beta} FIRST(\beta)$$
- Пусть $\beta = X_1 X_2 \ldots X_n$.
Тогда терминалы из $FIRST(X_1)$ входят в $FIRST(\beta)$,
терминалы из $FIRST(X_2)$ входят в $FIRST(\beta)$ при условии $\varepsilon \in FIRST(X_1)$,
терминалы из $FIRST(X_3)$ входят в $FIRST(\beta)$ при условии $\varepsilon \in FIRST(X_1) \cap FIRST(X_2)$ и т. д.
Наконец, если $\varepsilon \in FIRST(X_1) \cap FIRST(X_2) \cap \ldots \cap FIRST(X_n)$, то $\varepsilon \in FIRST(\beta)$.

Множество FIRST состоит только из терминалов!

Алгоритм построения множеств FIRST для нетермин.

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

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

- for $a \in \Sigma$:

Алгоритм построения множеств FIRST для нетермин.

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

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

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
    $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
    $FIRST(A) = \emptyset$
- 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 нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $X_0 = \varepsilon$

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $X_0 = \varepsilon$
  2. $i = 0$

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $X_0 = \varepsilon$
  2. $i = 0$
  3. while $(\varepsilon \in FIRST(X_i))$ and $(i < n)$:

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \ldots X_n) \in P$:
  1. $X_0 = \varepsilon$
  2. $i = 0$
  3. while $(\varepsilon \in FIRST(X_i))$ and $(i < n)$:
     1. i+=1

<u>Вход:</u> КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
<u>Выход:</u> Набор множеств $FIRST(A)$ для всех терминалов и нетерминалов $A \in \Gamma$

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $X_0 = \varepsilon$
  2. $i = 0$
  3. while $(\varepsilon \in FIRST(X_i))$ and $(i < n)$:
     1. i+=1
     2. $FIRST(A) = FIRST(A) \cup (FIRST(X_i) \setminus \{\varepsilon\})$

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $X_0 = \varepsilon$
  2. $i = 0$
  3. while $(\varepsilon \in FIRST(X_i))$ and $(i < n)$:
     1. i+=1
     2. $FIRST(A) = FIRST(A) \cup (FIRST(X_i) \backslash \{\varepsilon\})$
  4. if $(\varepsilon \in FIRST(X_n))$ and $(i = n)$:

Алгоритм построения множеств FIRST для нетермин.

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

- for $a \in \Sigma$:
  $FIRST(a) = \{a\}$
- for $(A \to \gamma) \in P$:
  if $(A \to \varepsilon) \in P$:
  $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$
  else:
  $FIRST(A) = \emptyset$
- while (все множества $FIRST(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $X_0 = \varepsilon$
  2. $i = 0$
  3. while $(\varepsilon \in FIRST(X_i))$ and $(i < n)$:
     1. i+=1
     2. $FIRST(A) = FIRST(A) \cup (FIRST(X_i) \setminus \{\varepsilon\})$
  4. if $(\varepsilon \in FIRST(X_n))$ and $(i = n)$:
     $FIRST(A) = FIRST(A) \cup \{\varepsilon\}$

Построение множеств FIRST в примере 1 для терминалов и нетерминалов

Перенумеруем правила в примере 1.

$$S \rightarrow CA \ (1)$$
$$S \rightarrow CB \ (2)$$
$$A \rightarrow bBC \ (3)$$
$$A \rightarrow cb \ (4)$$
$$B \rightarrow b \ (5)$$
$$C \rightarrow dC \ (6)$$
$$C \rightarrow \varepsilon \ (7)$$

И напишем протокол построения множества FIRST для всех нетерминалов.

№ такта	$FIRST(S)$	$FIRST(A)$	$FIRST(B)$	$FIRST(C)$
1	$\emptyset$	$\emptyset$	$\emptyset$	$\{\varepsilon\}$
2	$(1):\emptyset$	$\emptyset$	$\emptyset$	$\{\varepsilon\}$
3	$(2):\emptyset$	$\emptyset$	$\emptyset$	$\{\varepsilon\}$
4	$\emptyset$	$(3):\{b\}$	$\emptyset$	$\{\varepsilon\}$
5	$\emptyset$	$(4):\{b,c\}$	$\emptyset$	$\{\varepsilon\}$
6	$\emptyset$	$\{b,c\}$	$(5):\{b\}$	$\{\varepsilon\}$
7	$\emptyset$	$\{b,c\}$	$\{b\}$	$(6):\{d,\varepsilon\}$
8	$\emptyset$	$\{b,c\}$	$\{b\}$	$(7):\{d,\varepsilon\}$
9	$(1):\{b,c,d\}$	$\{b,c\}$	$\{b\}$	$\{d,\varepsilon\}$
10	$(2):\{b,c,d\}$	$\{b,c\}$	$\{b\}$	$\{d,\varepsilon\}$
11	$\{b,c,d\}$	$(3):\{b,c\}$	$\{b\}$	$\{d,\varepsilon\}$
12	$\{b,c,d\}$	$(4):\{b,c\}$	$\{b\}$	$\{d,\varepsilon\}$
13	$\{b,c,d\}$	$\{b,c\}$	$(5):\{b\}$	$\{d,\varepsilon\}$
14	$\{b,c,d\}$	$\{b,c\}$	$\{b\}$	$(6):\{d,\varepsilon\}$
15	$\{b,c,d\}$	$\{b,c\}$	$\{b\}$	$(7):\{d,\varepsilon\}$

Построение множества FIRST для цепочек из грамматических символов

- Теперь модифицируем алгоритм для вычисления множества $FIRST(\alpha)$ для произвольного $\alpha \in (\Sigma \cup \Gamma)^*$.

Построение множества FIRST для цепочек из грамматических символов

- Теперь модифицируем алгоритм для вычисления множества $FIRST(\alpha)$ для произвольного $\alpha \in (\Sigma \cup \Gamma)^*$.
- При этом считается, что множества $FIRST$ для терминалов и терминалов уже построены.

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2 \dots \alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2\ldots\alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2 \dots \alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2\dots\alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$
- $i = 0$

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2\dots\alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$
- $i = 0$
- while $(\varepsilon \in FIRST(\alpha_i))$ and $(i < n)$:

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2\dots\alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$
- $i = 0$
- while $(\varepsilon \in FIRST(\alpha_i))$ and $(i < n)$:
    1. i+=1

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2\ldots\alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$
- $i = 0$
- while $(\varepsilon \in FIRST(\alpha_i))$ and $(i < n)$:
  1. i+=1
  2. $FIRST(\alpha) = FIRST(\alpha) \cup (FIRST(\alpha_i) \backslash \{\varepsilon\})$

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2 \dots \alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$
- $i = 0$
- while $(\varepsilon \in FIRST(\alpha_i))$ and $(i < n)$:
  1. i+=1
  2. $FIRST(\alpha) = FIRST(\alpha) \cup (FIRST(\alpha_i) \setminus \{\varepsilon\})$
- if $(\varepsilon \in FIRST(\alpha_n))$ and $(i = n)$:

Алгоритм построения множества FIRST для цепочек из грамматических символов

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$, $\alpha = \alpha_1\alpha_2\ldots\alpha_n$, $\alpha_i \in \Sigma \cup \Gamma$.
Выход: Множество $FIRST(\alpha)$

- $FIRST(\alpha) = \emptyset$
- $\alpha_0 = \varepsilon$
- $i = 0$
- while $(\varepsilon \in FIRST(\alpha_i))$ and $(i < n)$:
  1. i+=1
  2. $FIRST(\alpha) = FIRST(\alpha) \cup (FIRST(\alpha_i) \backslash \{\varepsilon\})$
- if $(\varepsilon \in FIRST(\alpha_n))$ and $(i = n)$:
  $FIRST(\alpha) = FIRST(\alpha) \cup \{\varepsilon\}$

Пример 1 построения множества FIRST для цепочек

- Напомним, что мы уже вычислили $FIRST(a) = \{a\}$, $FIRST(b) = \{b\}$, $FIRST(c) = \{c\}$, $FIRST(S) = \{b, c, d\}$, $FIRST(A) = \{b, c\}$, $FIRST(B) = \{b\}$, $FIRST(C) = \{d, \varepsilon\}$

Пример 1 построения множества FIRST для цепочек

- Напомним, что мы уже вычислили
  $FIRST(a) = \{a\}$, $FIRST(b) = \{b\}$, $FIRST(c) = \{c\}$,
  $FIRST(S) = \{b, c, d\}$, $FIRST(A) = \{b, c\}$, $FIRST(B) = \{b\}$,
  $FIRST(C) = \{d, \varepsilon\}$
- Тогда
  $FIRST(AaC) = FIRST(A) = \{b, c\}$,
  $FIRST(cAC) = FIRST(c) = \{c\}$,
  $FIRST(CAb) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{d\} \cup \{b, c\} = \{b, c, d\}$

Лемма 4 о множестве FOLLOW

Лемма 4 о множестве FOLLOW

Для любого $X \in \Gamma$ если $S \Rightarrow^* \alpha X \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$, то $(FIRST(\beta) \setminus \{\varepsilon\}) \subseteq FOLLOW(X)$

Лемма 4 о множестве FOLLOW

Для любого $X \in \Gamma$ если $S \Rightarrow^* \alpha X \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$, то $(FIRST(\beta) \setminus \{\varepsilon\}) \subseteq FOLLOW(X)$

Доказательство. Пусть терминал $a \in \Sigma$ принадлежит множеству $FIRST(\beta)$. Тогда $\beta \Rightarrow^* a\gamma$ для некоторого $\gamma \in (\Sigma \cup \Gamma)^*$. Следовательно, $S \Rightarrow^* \alpha X a \gamma$, а значит, $a \in FOLLOW(X)$.

Лемма 4 о множестве FOLLOW

Для любого $X \in \Gamma$ если $S \Rightarrow^* \alpha X \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$, то $(FIRST(\beta) \setminus \{\varepsilon\}) \subseteq FOLLOW(X)$

Доказательство. Пусть терминал $a \in \Sigma$ принадлежит множеству $FIRST(\beta)$. Тогда $\beta \Rightarrow^* a\gamma$ для некоторого $\gamma \in (\Sigma \cup \Gamma)^*$. Следовательно, $S \Rightarrow^* \alpha X a \gamma$, а значит, $a \in FOLLOW(X)$.

Следующее замечание (**доказательство.**-упр.) поможет нам построить алгоритм для поиска множеств *FOLLOW* для всех нетерминалов.

Лемма 4 о множестве FOLLOW

Лемма 4 о множестве FOLLOW

Для любого $X \in \Gamma$ если $S \Rightarrow^* \alpha X \beta$ для некоторых $\alpha, \beta \in (\Sigma \cup \Gamma)^*$, то $(FIRST(\beta) \backslash \{\varepsilon\}) \subseteq FOLLOW(X)$

Доказательство. Пусть терминал $a \in \Sigma$ принадлежит множеству $FIRST(\beta)$. Тогда $\beta \Rightarrow^* a\gamma$ для некоторого $\gamma \in (\Sigma \cup \Gamma)^*$. Следовательно, $S \Rightarrow^* \alpha X a \gamma$, а значит, $a \in FOLLOW(X)$.

Следующее замечание (**доказательство.**-упр.) поможет нам построить алгоритм для поиска множеств $FOLLOW$ для всех нетерминалов.

Замечание 2

Пусть $A \to X_1 X_2 \dots X_n$. Тогда

1. Если $X_n \in \Gamma$, то $FOLLOW(A) \subseteq FOLLOW(X_n)$.
2. Если $X_i \in \Gamma$ и $\varepsilon \in FOLLOW(X_{i+1}) \cap \dots \cap FOLLOW(X_n)$, то $FOLLOW(A) \subseteq FOLLOW(X_i)$
3. Если $X_i \in \Gamma$, то $FIRST(X_{i+1} \dots X_n) \subseteq FOLLOW(X_i)$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \ldots X_n) \in P$:
  1. $i = n$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \ldots X_n) \in P$:
  1. $i = n$
  2. $ann = true$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $i = n$
  2. $ann = true$
  3. while $(X_i \in \Gamma)$ and $(i > 1)$:

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $i = n$
  2. $ann = true$
  3. while $(X_i \in \Gamma)$ and $(i > 1)$:
     1. $FOLLOW(X_i) = FOLLOW(X_i) \cup (FIRST(X_{i+1} \dots X_{n+1}) \backslash \{\varepsilon\})$

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $i = n$
  2. $ann = true$
  3. while $(X_i \in \Gamma)$ and $(i > 1)$:
     1. $FOLLOW(X_i) = FOLLOW(X_i) \cup (FIRST(X_{i+1} \dots X_{n+1}) \backslash \{\varepsilon\})$
     2. if ann:
        $FOLLOW(X_i) = FOLLOW(X_i) \cup FOLLOW(A)$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $i = n$
  2. $ann = true$
  3. while $(X_i \in \Gamma)$ and $(i > 1)$:
     1. $FOLLOW(X_i) = FOLLOW(X_i) \cup (FIRST(X_{i+1} \dots X_{n+1}) \setminus \{\varepsilon\})$
     2. if ann:
        $FOLLOW(X_i) = FOLLOW(X_i) \cup FOLLOW(A)$
     3. $ann = ann \wedge (\varepsilon \in FIRST(X_i))$

Алгоритм построения множества FOLLOW

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $i = n$
  2. $ann = true$
  3. while $(X_i \in \Gamma)$ and $(i > 1)$:
     1. $FOLLOW(X_i) = FOLLOW(X_i) \cup (FIRST(X_{i+1} \dots X_{n+1}) \backslash \{\varepsilon\})$
     2. if ann:
        $FOLLOW(X_i) = FOLLOW(X_i) \cup FOLLOW(A)$
     3. $ann = ann \wedge (\varepsilon \in FIRST(X_i))$
     4. i=i-1

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Набор множеств $FOLLOW(A)$ для всех нетерминалов $A \in \Gamma$

- for $A \in \Gamma$:
  $FOLLOW(A) = \emptyset$
- $FOLLOW(S) = \{\dashv\}$
- $P = P \cup (X_{n+1} \to \varepsilon)$
- while (все множества $FOLLOW(A)$ для всех $A \in \Gamma$ не стабилизировались):
  for $(A \to X_1 X_2 \dots X_n) \in P$:
  1. $i = n$
  2. $ann = true$
  3. while $(X_i \in \Gamma)$ and $(i > 1)$:
     1. $FOLLOW(X_i) = FOLLOW(X_i) \cup (FIRST(X_{i+1} \dots X_{n+1}) \backslash \{\varepsilon\})$
     2. if ann:
        $FOLLOW(X_i) = FOLLOW(X_i) \cup FOLLOW(A)$
     3. $ann = ann \wedge (\varepsilon \in FIRST(X_i))$
     4. i=i-1

Множество FOLLOW строится только для нетерминалов!

Построение множества FOLLOW в примере 1 для терминалов и нетерминалов

Снова рассмотрим грамматику из примера 1.

$$S \rightarrow CA \ (1)$$
$$S \rightarrow CB \ (2)$$
$$A \rightarrow bBC \ (3)$$
$$A \rightarrow cb \ (4)$$
$$B \rightarrow b \ (5)$$
$$C \rightarrow dC \ (6)$$
$$C \rightarrow \varepsilon \ (7)$$

Построение множества FOLLOW в примере 1 для терминалов и нетерминалов

Снова рассмотрим грамматику из примера 1.

$$S \to CA \ (1)$$
$$S \to CB \ (2)$$
$$A \to bBC \ (3)$$
$$A \to cb \ (4)$$
$$B \to b \ (5)$$
$$C \to dC \ (6)$$
$$C \to \varepsilon \ (7)$$

Вспомним, что
$FIRST(S) = \{b, c, d\}$, $FIRST(A) = \{b, c\}$, $FIRST(B) = \{b\}$, $FIRST(C) = \{d, \varepsilon\}$

Построение множества FOLLOW в примере 1 для терминалов и нетерминалов

Снова рассмотрим грамматику из примера 1.

$$S \to CA \ (1)$$
$$S \to CB \ (2)$$
$$A \to bBC \ (3)$$
$$A \to cb \ (4)$$
$$B \to b \ (5)$$
$$C \to dC \ (6)$$
$$C \to \varepsilon \ (7)$$

Вспомним, что
$FIRST(S) = \{b, c, d\}$, $FIRST(A) = \{b, c\}$, $FIRST(B) = \{b\}$, $FIRST(C) = \{d, \varepsilon\}$

И напишем протокол построения множества FOLLOW для всех нетерминалов.

Протокол постр. FOLLOW для нетермин. в примере 1

№ такта	$FOLLOW(S)$	$FOLLOW(A)$	$FOLLOW(B)$	$FOLLOW(C)$
1	$\{\dashv\}$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{\dashv\}$	$(1):\{\dashv\}$	$\emptyset$	$(1):\{b,c\}$
3	$\{\dashv\}$	$\{\dashv\}$	$(2):\{\dashv\}$	$(2):\{b,c\}$
4	$\{\dashv\}$	$\{\dashv\}$	$(3):\{\dashv,d\}$	$(3):\{\dashv,b,c\}$
5	$\{\dashv\}$	$\{\dashv\}$	$(3):\{\dashv,d\}$	$(3):\{\dashv,b,c\}$
6	$\{\dashv\}$	$(1):\{\dashv\}$	$\{\dashv,d\}$	$(1):\{\dashv,b,c\}$
7	$\{\dashv\}$	$\{\dashv\}$	$(2):\{\dashv\}$	$(2):\{\dashv,b,c\}$
8	$\{\dashv\}$	$\{\dashv\}$	$(3):\{\dashv,d\}$	$(3):\{\dashv,b,c\}$

№ такта	$FOLLOW(S)$	$FOLLOW(A)$	$FOLLOW(B)$	$FOLLOW(C)$
1	$\{\dashv\}$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{\dashv\}$	$(1) : \{\dashv\}$	$\emptyset$	$(1) : \{b, c\}$
3	$\{\dashv\}$	$\{\dashv\}$	$(2) : \{\dashv\}$	$(2) : \{b, c\}$
4	$\{\dashv\}$	$\{\dashv\}$	$(3) : \{\dashv, d\}$	$(3) : \{\dashv, b, c\}$
5	$\{\dashv\}$	$\{\dashv\}$	$(3) : \{\dashv, d\}$	$(3) : \{\dashv, b, c\}$
6	$\{\dashv\}$	$(1) : \{\dashv\}$	$\{\dashv, d\}$	$(1) : \{\dashv, b, c\}$
7	$\{\dashv\}$	$\{\dashv\}$	$(2) : \{\dashv\}$	$(2) : \{\dashv, b, c\}$
8	$\{\dashv\}$	$\{\dashv\}$	$(3) : \{\dashv, d\}$	$(3) : \{\dashv, b, c\}$

Таким образом,
$FOLLOW(S) = \{\dashv\}$, $FOLLOW(A) = \{\dashv\}$, $FOLLOW(B) = \{\dashv, d\}$,
$FOLLOW(C) = \{\dashv, b, c\}$

Определение множества SELECT и LL(1)-грамматики

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $(A \to \gamma) \in P$. Множеством $SELECT(A \to \gamma)$ (правилом выбора для правила $A \to \gamma$) называется множество

$$\begin{cases} FIRST(\beta), \text{ если } \varepsilon \notin FIRST(\beta) \\ (FIRST(\beta) \backslash \{\varepsilon\}) \cup FOLLOW(A), \text{ иначе} \end{cases}$$

Определение множества SELECT и LL(1)-грамматики

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $(A \to \gamma) \in P$. Множеством $SELECT(A \to \gamma)$ (правилом выбора для правила $A \to \gamma$) называется множество

$$\begin{cases} FIRST(\beta), \text{ если } \varepsilon \notin FIRST(\beta) \\ (FIRST(\beta) \backslash \{\varepsilon\}) \cup FOLLOW(A), \text{ иначе} \end{cases}$$

- Грамматика G называется *$LL(1)$-грамматикой*, если для любых двух правил $A \to \gamma_1$, $A \to \gamma_2 \in P$ из того, что $\gamma_1 \neq \gamma_2$ следует, что $SELECT(A \to \gamma_1) \cap SELECT(A \to \gamma_2) = \emptyset$
(т.е. множество правил выбора для альтернатив $A \to \gamma_1 \mid \gamma_2$ не пересекаются).

Определение множества SELECT и LL(1)-грамматики

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС грамматика и $(A \to \gamma) \in P$. Множеством $SELECT(A \to \gamma)$ (правилом выбора для правила $A \to \gamma$) называется множество

$$\begin{cases} FIRST(\beta), \text{ если } \varepsilon \notin FIRST(\beta) \\ (FIRST(\beta) \backslash \{\varepsilon\}) \cup FOLLOW(A), \text{ иначе} \end{cases}$$

- Грамматика G называется $LL(1)$-грамматикой, если для любых двух правил $A \to \gamma_1$, $A \to \gamma_2 \in P$ из того, что $\gamma_1 \neq \gamma_2$ следует, что $SELECT(A \to \gamma_1) \cap SELECT(A \to \gamma_2) = \emptyset$
(т.е. множество правил выбора для альтернатив $A \to \gamma_1 \mid \gamma_2$ не пересекаются).
- Если грамматика G является разделенной, то она, легко понять, является $LL(1)$-грамматикой (почему?).

Снова рассмотрим грамматику из примера 1

$$S \to CA \ (1)$$
$$S \to B \ (2)$$
$$A \to bBC \ (3)$$
$$A \to cb \ (4)$$
$$B \to b \ (5)$$
$$C \to dC \ (6)$$
$$C \to \varepsilon \ (7)$$

Множества FIRST, FOLLOW мы уже посчитали:

$FIRST(S) = \{b, c, d\}$, $FIRST(A) = \{b, c\}$, $FIRST(B) = \{b\}$, $FIRST(C) = \{d, \varepsilon\}$
и

$FOLLOW(S) = \{\dashv\}$, $FOLLOW(A) = \{\dashv\}$, $FOLLOW(B) = \{\dashv, d\}$,
$FOLLOW(C) = \{\dashv, b, c\}$

Найдем множества $SELECT(A \to \gamma)$ в грамматике G.

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества *FIRST* для правых частей всех грамматики G:

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества $FIRST$ для правых частей всех грамматики G:

$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\}$,

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества *FIRST* для правых частей всех грамматики G:

$$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\},$$

$$FIRST(CB) = (FIRST(C) \backslash \{\varepsilon\}) \cup (FIRST(B)) = \{b, d\}$$

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества *FIRST* для правых частей всех грамматики G:

$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\}$,

$FIRST(CB) = (FIRST(C) \backslash \{\varepsilon\}) \cup (FIRST(B)) = \{b, d\}$

$FIRST(bBC) = FIRST(b) = \{b\}$

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества *FIRST* для правых частей всех грамматики G:

$$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\},$$

$$FIRST(CB) = (FIRST(C) \backslash \{\varepsilon\}) \cup (FIRST(B)) = \{b, d\}$$

$$FIRST(bBC) = FIRST(b) = \{b\}$$

$$FIRST(cb) = FIRST(c) = \{c\}$$

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества *FIRST* для правых частей всех грамматики G:

$$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\},$$

$$FIRST(CB) = (FIRST(C) \backslash \{\varepsilon\}) \cup (FIRST(B)) = \{b, d\}$$

$$FIRST(bBC) = FIRST(b) = \{b\}$$

$$FIRST(cb) = FIRST(c) = \{c\}$$

$$FIRST(b) = FIRST(b) = \{b\}$$

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества $FIRST$ для правых частей всех грамматики G:

$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\}$,

$FIRST(CB) = (FIRST(C) \backslash \{\varepsilon\}) \cup (FIRST(B)) = \{b, d\}$

$FIRST(bBC) = FIRST(b) = \{b\}$

$FIRST(cb) = FIRST(c) = \{c\}$

$FIRST(b) = FIRST(b) = \{b\}$

$FIRST(dC) = FIRST(d) = \{d\}$

Построение множеств SELECT для примера 1 (продолжение)

Для этого найдем сначала множества *FIRST* для правых частей всех грамматики G:

$$FIRST(CA) = (FIRST(C) \backslash \{\varepsilon\}) \cup FIRST(A) = \{b, c, d\},$$

$$FIRST(CB) = (FIRST(C) \backslash \{\varepsilon\}) \cup (FIRST(B)) = \{b, d\}$$

$$FIRST(bBC) = FIRST(b) = \{b\}$$

$$FIRST(cb) = FIRST(c) = \{c\}$$

$$FIRST(b) = FIRST(b) = \{b\}$$

$$FIRST(dC) = FIRST(d) = \{d\}$$

$$FIRST(\varepsilon) = \varepsilon$$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества *SELECT*:

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества *SELECT*:

$SELECT(S \to CA) = FIRST(CA) = \{b, c, d\}$,

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества *SELECT*:

$SELECT(S \rightarrow CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S \rightarrow CB) = FIRST(CB) = \{b, d\}$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества *SELECT*:

$SELECT(S \to CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S \to CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \to bBC) = FIRST(bBC) = \{b\}$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества $SELECT$:

$SELECT(S{\rightarrow}CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S{\rightarrow}CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \rightarrow bBC) = FIRST(bBC) = \{b\}$

$SELECT(A \rightarrow cb) = FIRST(cb) = \{c\}$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества *SELECT*:

$SELECT(S \to CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S \to CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \to bBC) = FIRST(bBC) = \{b\}$

$SELECT(A \to cb) = FIRST(cb) = \{c\}$

$SELECT(B \to b) = FIRST(b) = \{b\}$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества *SELECT*:

$SELECT(S{\rightarrow}CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S{\rightarrow}CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \rightarrow bBC) = FIRST(bBC) = \{b\}$

$SELECT(A \rightarrow cb) = FIRST(cb) = \{c\}$

$SELECT(B \rightarrow b) = FIRST(b) = \{b\}$

$SELECT(C \rightarrow dC) = FIRST(dC) = \{d\}$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества $SELECT$:

$SELECT(S \to CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S \to CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \to bBC) = FIRST(bBC) = \{b\}$

$SELECT(A \to cb) = FIRST(cb) = \{c\}$

$SELECT(B \to b) = FIRST(b) = \{b\}$

$SELECT(C \to dC) = FIRST(dC) = \{d\}$

$$SELECT(C \to \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(C) =$$
$$= (\{\varepsilon\} \setminus \{\varepsilon\}) \cup FOLLOW(C) = FOLLOW(C) = \{\dashv, b, c\}$$

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества $SELECT$:

$SELECT(S{\to}CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S{\to}CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \to bBC) = FIRST(bBC) = \{b\}$

$SELECT(A \to cb) = FIRST(cb) = \{c\}$

$SELECT(B \to b) = FIRST(b) = \{b\}$

$SELECT(C \to dC) = FIRST(dC) = \{d\}$

$SELECT(C \to \varepsilon) = (FIRST(\varepsilon)\setminus\{\varepsilon\}) \cup FOLLOW(C) =$
$= (\{\varepsilon\}\setminus\{\varepsilon\}) \cup FOLLOW(C) = FOLLOW(C) = \{\dashv, b, c\}$

Видно, что эта грамматика не является $LL(1)$-грамматикой (почему?).

Построение множеств SELECT для примера 1 (продолжение)

Теперь найдем множества $SELECT$:

$SELECT(S \to CA) = FIRST(CA) = \{b, c, d\}$,

$SELECT(S \to CB) = FIRST(CB) = \{b, d\}$

$SELECT(A \to bBC) = FIRST(bBC) = \{b\}$

$SELECT(A \to cb) = FIRST(cb) = \{c\}$

$SELECT(B \to b) = FIRST(b) = \{b\}$

$SELECT(C \to dC) = FIRST(dC) = \{d\}$

$$SELECT(C \to \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(C) =$$
$$= (\{\varepsilon\} \setminus \{\varepsilon\}) \cup FOLLOW(C) = FOLLOW(C) = \{\dashv, b, c\}$$

Видно, что эта грамматика не является $LL(1)$-грамматикой (почему?).

Алгоритм построения нисходящего анализатора

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Управляющая таблица для δ нисходящего анализатора с одним состоянием со стековым алфавитом $\Delta = \Sigma \cup \Gamma$ и с начальным содержимым стека S

Алгоритм построения нисходящего анализатора

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Управляющая таблица для δ нисходящего анализатора с одним состоянием со стековым алфавитом $\Delta = \Sigma \cup \Gamma$ и с начальным содержимым стека S

1. for $(A \to \gamma) \in P$ найдем $SELECT(A \to \gamma)$

Алгоритм построения нисходящего анализатора

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Управляющая таблица для δ нисходящего анализатора с одним состоянием со стековым алфавитом $\Delta = \Sigma \cup \Gamma$ и с начальным содержимым стека S

1. for $(A \to \gamma) \in P$ найдем $SELECT(A \to \gamma)$
2. for $A \in \Gamma$:
   for $(A \to \gamma) \in P$:
     for $a \in SELECT(A \to \gamma)$:
       $\delta(A, a) = (\gamma, \_)$

Алгоритм построения нисходящего анализатора

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Управляющая таблица для δ нисходящего анализатора с одним состоянием со стековым алфавитом $\Delta = \Sigma \cup \Gamma$ и с начальным содержимым стека S

1. for $(A \to \gamma) \in P$ найдем $SELECT(A \to \gamma)$
2. for $A \in \Gamma$:
   for $(A \to \gamma) \in P$:
   for $a \in SELECT(A \to \gamma)$:
   $\delta(A, a) = (\gamma, \_)$
3. for $a \in \Sigma$:
   $\delta(a, a) = (\varepsilon, \to)$

Алгоритм построения нисходящего анализатора

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Управляющая таблица для δ нисходящего анализатора с одним состоянием со стековым алфавитом $\Delta = \Sigma \cup \Gamma$ и с начальным содержимым стека S

1. for $(A \to \gamma) \in P$ найдем $SELECT(A \to \gamma)$
2. for $A \in \Gamma$:
   for $(A \to \gamma) \in P$:
   for $a \in SELECT(A \to \gamma)$:
   $\delta(A, a) = (\gamma, \_)$
3. for $a \in \Sigma$:
   $\delta(a, a) = (\varepsilon, \to)$
4. Команда допуска — $\delta(\nabla, \dashv) = \vee$

Алгоритм построения нисходящего анализатора

Алгоритм построения нисходящего анализатора

Вход: КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход: Управляющая таблица для δ нисходящего анализатора с одним состоянием со стековым алфавитом $\Delta = \Sigma \cup \Gamma$ и с начальным содержимым стека S

1. for $(A \to \gamma) \in P$ найдем $SELECT(A \to \gamma)$
2. for $A \in \Gamma$:
   for $(A \to \gamma) \in P$:
   for $a \in SELECT(A \to \gamma)$:
   $\delta(A, a) = (\gamma, \_)$
3. for $a \in \Sigma$:
   $\delta(a, a) = (\varepsilon, \to)$
4. Команда допуска — $\delta(\nabla, \dashv) = \vee$

Если грамматика является $LL(1)$ грамматикой, то такой автомат будет детерминированным (почему?).

Теорема о корректности работы алгоритма нисходящего анализатора

Теорема о корректности работы алгоритма нисходящего анализатора

Алгоритм построения нисходящего анализатора работает корректно, то есть автомат, построенный по этому алгоритму распознает язык $L(G)$.

- Заметим, что МП-автомат $\mathcal{M}'$, построенный по алгоритму, содержит только часть переходов НПМА $\mathcal{M}$, который строился по КС-грамматике G при доказательстве теоремы 2 в лекции 13 и который допускал язык $L(G)$. Следовательно, если для каждого $w \in L(G)$ в автомате $\mathcal{M}'$ есть последовательность переходов $[S, w \dashv] \models^* [\varepsilon, \dashv]$, то, эта последовательность есть в $\mathcal{M}$, и значит, как ранее было доказано в теореме 2 лекции 13, $w \in L(G)$. Таким образом, $L(\mathcal{M}') \subseteq L(G)$.

Теорема о корректности работы алгоритма нисходящего анализатора

Теорема о корректности работы алгоритма нисходящего анализатора

Алгоритм построения нисходящего анализатора работает корректно, то есть автомат, построенный по этому алгоритму распознает язык $L(G)$.

- Заметим, что МП-автомат $\mathcal{M}'$, построенный по алгоритму, содержит только часть переходов НПМА $\mathcal{M}$, который строился по КС-грамматике G при доказательстве теоремы 2 в лекции 13 и который допускал язык $L(G)$. Следовательно, если для каждого $w \in L(G)$ в автомате $\mathcal{M}'$ есть последовательность переходов $[S, w \dashv] \models^* [\varepsilon, \dashv]$, то, эта последовательность есть в $\mathcal{M}$, и значит, как ранее было доказано в теореме 2 лекции 13, $w \in L(G)$. Таким образом, $L(\mathcal{M}') \subseteq L(G)$.
- Покажем, что если $w \in L(G)$, то слово w распознается нисходящим анализатором $\mathcal{M}'$, т.е. $L(G) \subseteq L(\mathcal{M}')$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора

- Пусть $S \Rightarrow^* u_i A\beta_i \Rightarrow u_i\gamma\beta_i \Rightarrow^* u_i v_i = w$ — левосторонний вывод в грамматике G, покажем что в автомате $\mathcal{M}'$ найдется переход $\delta(A, x) = (\gamma, \_)$ для некоторого $x \in \Sigma \cup \{\dashv\}$, т.е. найдется $x \in \Sigma \cup \{\dashv\}$, что $x \in SELECT(A \to \gamma)$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора

- Пусть $S \Rightarrow^* u_i A \beta_i \Rightarrow u_i \gamma \beta_i \Rightarrow^* u_i v_i = w$ — левосторонний вывод в грамматике G, покажем что в автомате $\mathcal{M}'$ найдется переход $\delta(A, x) = (\gamma, \_)$ для некоторого $x \in \Sigma \cup \{\dashv\}$, т.е. найдется $x \in \Sigma \cup \{\dashv\}$, что $x \in SELECT(A \to \gamma)$.
- Ясно, что $A\beta_i \Rightarrow \gamma\beta_i \Rightarrow^* v_i$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора

- Пусть $S \Rightarrow^* u_i A \beta_i \Rightarrow u_i \gamma \beta_i \Rightarrow^* u_i v_i = w$ — левосторонний вывод в грамматике G, покажем что в автомате $\mathcal{M}'$ найдется переход $\delta(A, x) = (\gamma, \_)$ для некоторого $x \in \Sigma \cup \{\dashv\}$, т.е. найдется $x \in \Sigma \cup \{\dashv\}$, что $x \in SELECT(A \to \gamma)$.
- Ясно, что $A\beta_i \Rightarrow \gamma\beta_i \Rightarrow^* v_i$.
- Пусть в этом выводе $\gamma \not\Rightarrow^* \varepsilon$, тогда, в частности, $v_i \neq \varepsilon$, и найдется $x \in \Sigma$ — первый символ слова v_i и при этом $x \in FIRST(\gamma)$. Так как $\varepsilon \notin FIRST(\gamma)$, имеем $SELECT(A \to \gamma) = FIRST(\gamma)$. Значит, $x \in SELECT(A \to \gamma)$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора (продолжение)

- Пусть в этом выводе $\gamma \not\Rightarrow^* \varepsilon$, тогда, в частности, $v_i \neq \varepsilon$, и найдется $x \in \Sigma$ — первый символ слова v_i и при этом $x \in FIRST(\gamma)$. Так как $\varepsilon \notin FIRST(\gamma)$, имеем $SELECT(A \to \gamma) = FIRST(\gamma)$. Значит, $x \in SELECT(A \to \gamma)$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора (продолжение)

- Пусть в этом выводе $\gamma \not\Rightarrow^* \varepsilon$, тогда, в частности, $v_i \neq \varepsilon$, и найдется $x \in \Sigma$ — первый символ слова v_i и при этом $x \in FIRST(\gamma)$. Так как $\varepsilon \notin FIRST(\gamma)$, имеем $SELECT(A \to \gamma) = FIRST(\gamma)$. Значит, $x \in SELECT(A \to \gamma)$.
- Пусть в этом выводе $\gamma \Rightarrow^* \varepsilon$. Тогда $\varepsilon \in FIRST(\gamma)$ и потому $SELECT(A \to \gamma) = (FIRST(\gamma) \backslash \{\varepsilon\}) \cup FOLLOW(A)$ и $\beta_i \Rightarrow^* v_i$, откуда $S \Rightarrow^* u_i A \beta_i \Rightarrow^* u_i A v_i$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора (продолжение)

- Пусть в этом выводе $\gamma \not\Rightarrow^* \varepsilon$, тогда, в частности, $v_i \neq \varepsilon$, и найдется $x \in \Sigma$ — первый символ слова v_i и при этом $x \in FIRST(\gamma)$. Так как $\varepsilon \notin FIRST(\gamma)$, имеем $SELECT(A \to \gamma) = FIRST(\gamma)$. Значит, $x \in SELECT(A \to \gamma)$.
- Пусть в этом выводе $\gamma \Rightarrow^* \varepsilon$. Тогда $\varepsilon \in FIRST(\gamma)$ и потому $SELECT(A \to \gamma) = (FIRST(\gamma) \backslash \{\varepsilon\}) \cup FOLLOW(A)$ и $\beta_i \Rightarrow^* v_i$, откуда $S \Rightarrow^* u_i A \beta_i \Rightarrow^* u_i A v_i$.
  1. Если $v_i = x v_i'$, то $S \Rightarrow^* u_i A x v_i'$ и $x \in FIRST(v_i) \cap FOLLOW(A) \subseteq SELECT(A \to \gamma)$.

Доказательство теоремы о корректности работы алгоритма нисходящего анализатора (продолжение)

- Пусть в этом выводе $\gamma \not\Rightarrow^* \varepsilon$, тогда, в частности, $v_i \neq \varepsilon$, и найдется $x \in \Sigma$ — первый символ слова v_i и при этом $x \in FIRST(\gamma)$. Так как $\varepsilon \notin FIRST(\gamma)$, имеем $SELECT(A \to \gamma) = FIRST(\gamma)$. Значит, $x \in SELECT(A \to \gamma)$.
- Пусть в этом выводе $\gamma \Rightarrow^* \varepsilon$. Тогда $\varepsilon \in FIRST(\gamma)$ и потому $SELECT(A \to \gamma) = (FIRST(\gamma) \setminus \{\varepsilon\}) \cup FOLLOW(A)$ и $\beta_i \Rightarrow^* v_i$, откуда $S \Rightarrow^* u_i A \beta_i \Rightarrow^* u_i A v_i$.
  1. Если $v_i = x v_i'$, то $S \Rightarrow^* u_i A x v_i'$ и $x \in FIRST(v_i) \cap FOLLOW(A) \subseteq SELECT(A \to \gamma)$.
  2. Если $v_i = \varepsilon$, то $x = \dashv$, $S \Rightarrow^* u_i A$ и $\dashv \in FOLLOW(A) \subseteq SELECT(A \to \gamma)$. И следовательно, $\dashv \in SELECT(A \to \gamma)$.