11. Множества FIRST, FOLLOW и SELECT. Алгоритмы их построения.

Множество FIRST

Для **произвольной цепочки** (не обязательно один нетерминал) α, множество $FIRST(\alpha)$ — это подмножество терминалов, с которых могут начинаться терминальные цепочки, выводимые из α.
Если из α можно вывести λ, то $\lambda \in FIRST(\alpha)$.

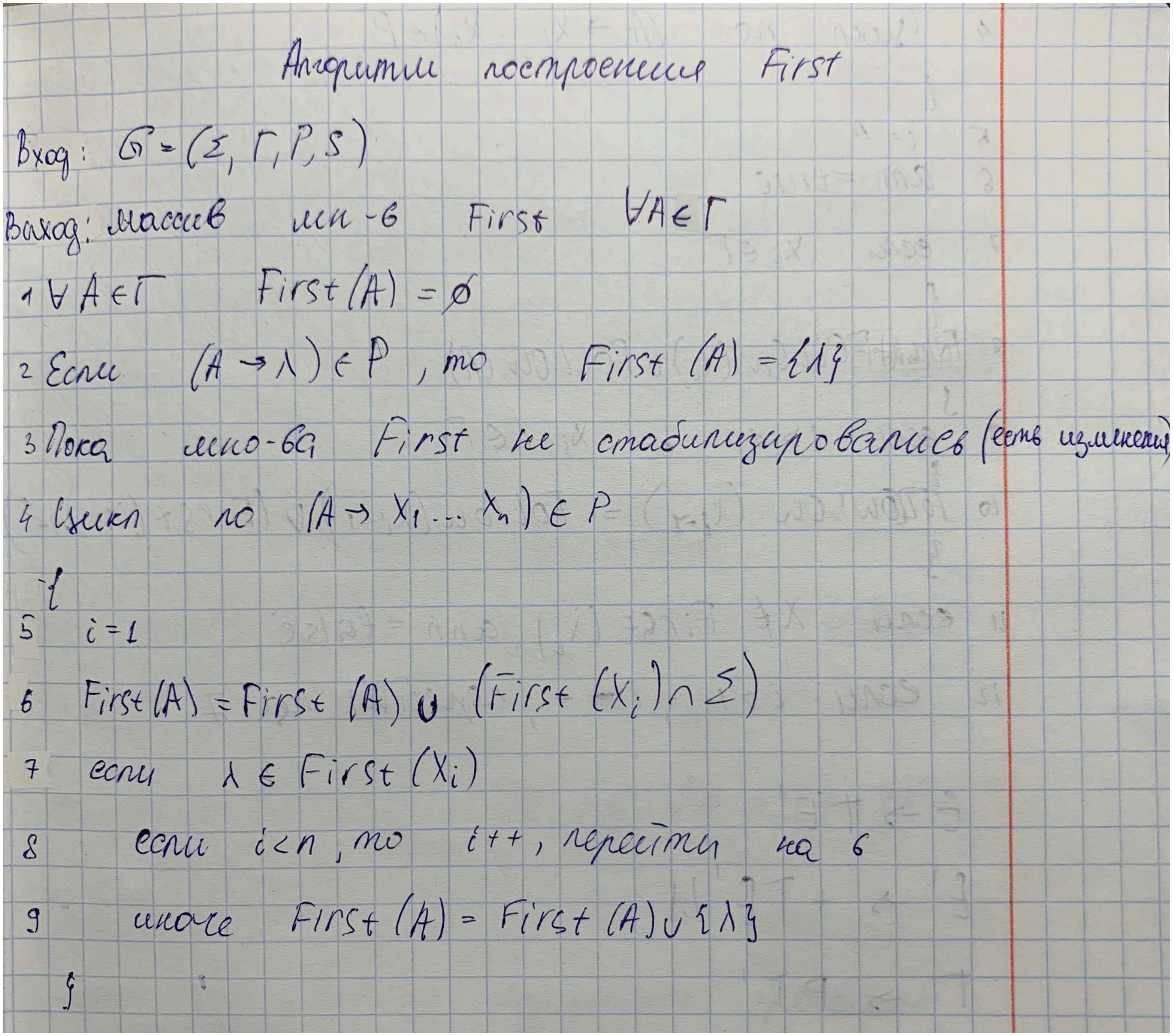
Пусть $G = (\Sigma, \Gamma, P, S), \alpha \in (\Sigma \cup \Gamma)^*$.

Определение

Множество FIRST(α) — подмножество $\Sigma \cup \{\lambda\}$ такое, что:
$a \in \mathrm{FIRST}(\alpha) \Leftrightarrow \alpha \Rightarrow^* a\beta$;
$\lambda \in \mathrm{FIRST}(\alpha) \Leftrightarrow \alpha \Rightarrow^* \lambda$.

Пример. Рассмотрим грамматику
$S \to AC, A \to abC|bB, B \to b, C \to c|\lambda$.
$\mathrm{FIRST}(AC) = \{a, b\}$
$\mathrm{FIRST}(CA) = \{a, b, c\}$

Алгоритм построения:



Алгоритм построения First

Вход: $G = (\Sigma, \Gamma, P, S)$

Выход: массив мн-в First $\forall A \in \Gamma$

1 $\forall A \in \Gamma$ $First(A) = \emptyset$

2 Если $(A \to \lambda) \in P$, то $First(A) = \{\lambda\}$

3 Пока мн-ва First не стабилизировались (есть изменения)

4 Цикл по $(A \to X_1 \dots X_n) \in P$

{

5 $i = 1$

6 $First(A) = First(A) \cup (First(X_i) \cap \Sigma)$

7 если $\lambda \in First(X_i)$

8 если $i < n$, то $i{+}{+}$, перейти на 6

9 иначе $First(A) = First(A) \cup \{\lambda\}$

}

Текстом:

1. Для каждого терминала a: $FIRST(a) = \{a\}$.
2. Для каждого нетерминала A: если есть правило $A \to \lambda$, добавить λ в $FIRST(A)$.
3. Итеративно для каждого правила $A \to X_1 X_2 \dots X_n$:
    - Добавлять в $FIRST(A)$ все неаннулирующие символы из $FIRST(X_1)$.
    - Если X_1 может быть аннулирован ($\lambda \in FIRST(X_1)$), переходить к $FIRST(X_2)$ и так далее.
    - Если все символы в правой части — аннулирующие, добавить λ в $FIRST(A)$.

Множество FOLLOW

Для нетерминала A, множество $FOLLOW(A)$ — это подмножество терминалов, которые могут следовать непосредственно за A в цепочках, выводимых из аксиомы.

Определение

Множество FOLLOW(A) — подмножество $\Sigma \cup \{\dashv\}$ такое, что:
$a \in \mathrm{FOLLOW}(A) \Leftrightarrow S \Rightarrow^* \alpha A a \beta$;
$\dashv \in \mathrm{FOLLOW}(A) \Leftrightarrow S \Rightarrow^* \alpha A$.

Для грамматики $S \to AC, A \to abC|bB, B \to b, C \to c|\lambda$:
$\mathrm{FOLLOW}(A) = \{c, \dashv\}$, $\mathrm{FOLLOW}(B) = \{c, \dashv\}$

Алгоритм построения:

Алгоритм построение FOLLOW

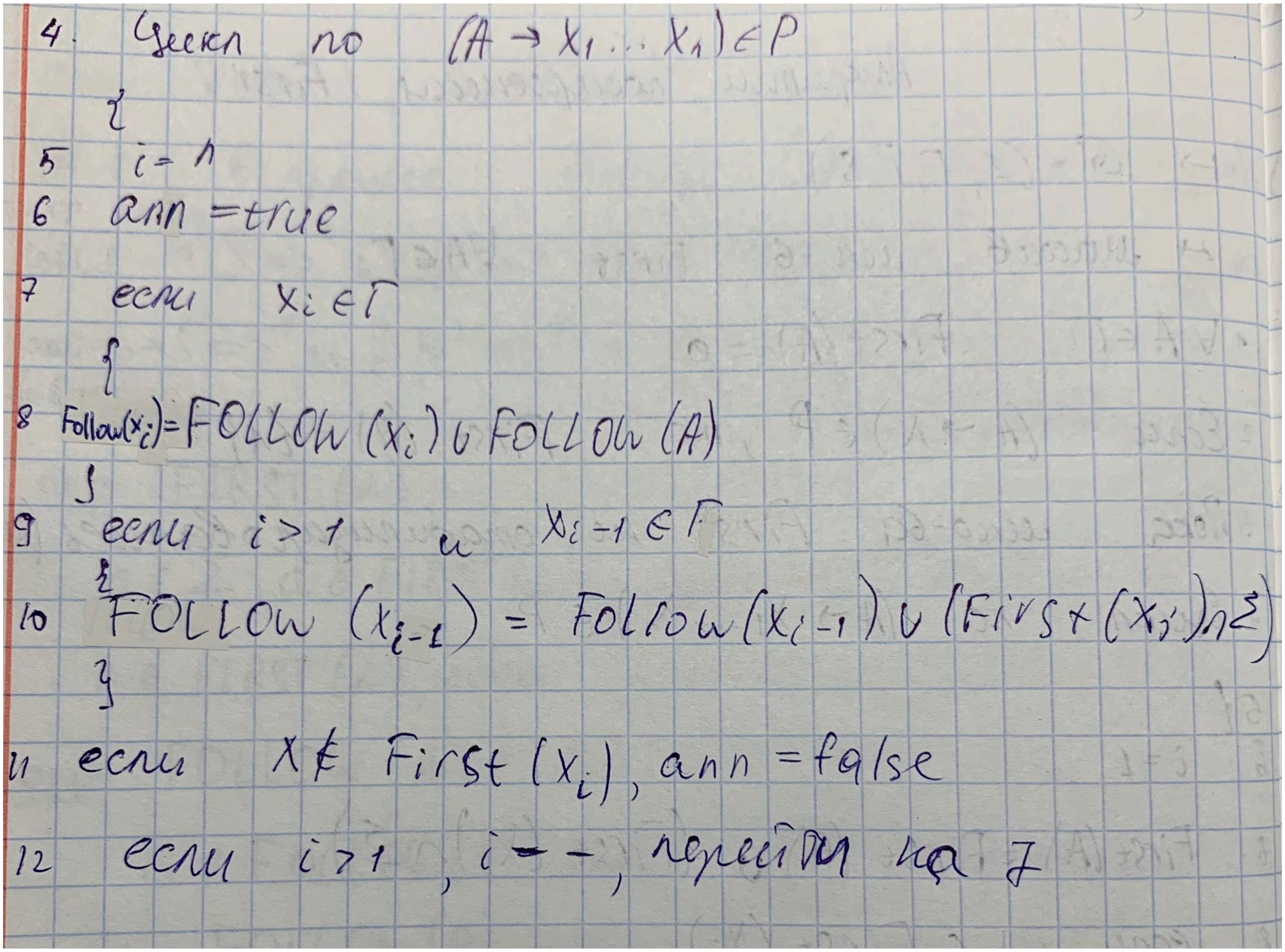
Вход: $G = (\Sigma, \Gamma, P, S)$

Выход: массив FOLLOW(A) $\forall A \in \Gamma$.

1 $\forall A \in \Gamma$ $\mathrm{FOLLOW}(A) = \varnothing$

2 $\mathrm{FOLLOW}(S) = \{\dashv\}$

3 Пока FOLLOW не стабилизирует. (есть изменение)



Текстом:

1. Начинаем с аксиомы $S: FOLLOW(S) = \{\dashv\}$.
2. Итеративно просматривать все правила вывода вида $A \to \alpha B \beta$:
    - Все **терминалы** из $FIRST(\beta)$ (кроме λ) добавляются в $FOLLOW(B)$.
    - Если β — пустая цепочка или аннулируемая ($\lambda \in FIRST(\beta)$), то всё множество $FOLLOW(A)$ добавляется в $FOLLOW(B)$.

Множество SELECT

Множество выбора $SELECT(A \to \gamma)$ определяется для каждого конкретного правила вывода.
Оно указывает, при каких входных символах следует применять данное правило.

Расчет SELECT:

- Если $\lambda \notin FIRST(\gamma)$, то $SELECT(A \to \gamma) = FIRST(\gamma)$.
- Иначе $SELECT(A \to \gamma) = (FIRST(\gamma) \setminus \{\lambda\}) \cup FOLLOW(A)$.

Алгоритма на паре не давала.