

Давайте сконструируем язык **L** через **обратную польскую запись**.

Зададим алфавит $\Sigma_0 = \{a, b, \dots, z, A, B, \dots, Z, 0, \dots, 9, _ \}$ - буквы, цифры и нижнее подчёркивание

Зададим наш терминальный алфавит $\Sigma = \Sigma_0 \cup \{ , \dots , \}$, то есть это Σ_0 и все пробельные символы

Нетерминальный алфавит $N = \{s_0, E_0, V_0, D_0, _ \}$, где s_0 - стартовое состояние, E_0 - это Expr, V_0 - это Var и Ident (в данном случае это одно и тоже), D_0 - это числа, $_$ - все пробельные символы

В нашем языке есть ключевые слова **If, While, Assign, Read, Write** и **Seq**

Для каждого из них введём правило:

- $s_0 \rightarrow E_0_s_0_s_0_If$
(Порядок веток прямой, соответственно если условие E_0 выполнится, то будет исполняться первый s_0 , иначе второй)
- $s_0 \rightarrow E_0_s_0_While$
- $s_0 \rightarrow V_0_E_0_Assign$
- $s_0 \rightarrow V_0_Read$
- $s_0 \rightarrow E_0_Write$
- $s_0 \rightarrow s_0_s_0_Seq$
(Порядок прямой, соответственно вначале идёт первое действие, затем второе)

Выражения:

Правила для бинарных операторов (порядок везде прямой):

- $E_0 \rightarrow E_0_E_0_ -$
(Из первого E_0 вычитаем второе E_0)
- $E_0 \rightarrow E_0_E_0_ +$
...
- $E_0 \rightarrow E_0_E_0_ \&\&$
- $E_0 \rightarrow E_0_E_0_ ||$

Правила для унарных операторов (пока что только унарный минус):

- $E_0 \rightarrow E_0_ - -$

Правила для чисел и переменных:

- $E_0 \rightarrow D_0$
- $E_0 \rightarrow V_0$

Переменные и идентификаторы - это строки, которые начинаются обязательно с буквы либо нижнего подчёркивания и состоят из символов Σ_0 , причём обязательно **НЕ совпадают** с ключевыми словами

- $V_0 \rightarrow (a|b|\dots|_|_a|\dots|aa|ab|\dots|Alya_0|\dots|Fkb1aa1c_d3|\dots) \Sigma_0^*$

(Перечисляем все корректные слова длины < 15 (**кроме ключевых слов**) и дальше добавляем что угодно, то есть Σ_0^*)

Числа:

- $D_0 \rightarrow (0|1|\dots|9)^+$

Пробелы:

- $\rightarrow \backslash t \mid \backslash n \mid \dots$ (все пробельные символы)