

Язык древних русов

Для счёта древних русов используется служебный символ \$, а сами цифры выглядят как

- 0. \$NOL\$
- 1. \$CELKOVIIY\$
- 2. \$POLUSHKA\$
- 3. \$CHETVERTUSHKA\$
- 4. \$OSMUSHKA\$
- 5. \$PUDOVICHOK\$
- 6. \$MEDYACHOK\$
- 7. \$SEREBRYACHOK\$
- 8. \$ZOLOTNICHOK\$
- 9. \$DEVYATICHOK\$

Числа из них порождаются стандартно, обозначим число за *number*.

Идентификаторы – слова из $[r', u', s', R', U', S']$, причём сначала идут маленькие, затем заглавные символы (возможно, строка пустая):

$$\begin{aligned} S_0 &\rightarrow @S@ \\ S &\rightarrow '' |r'|'u'|'s'|'R'|'U'|'S' \\ S &\rightarrow (|r'|'u'|'s')S \\ S &\rightarrow S(|R'|'U'|'S') \end{aligned}$$

Примеры : "\$@uuuURR@", @\$SRU@, @\$rusRUS@. Обозначим слово из языка этой грамматики за *ident*.

Каждый идентификатор соответствует одной глобальной переменной. Изначально каждая переменная равна 0.

Введём понятие выражений – *expr*.

Операторы – стандартные, со стандартными ассоциативностями/приоритетами. Логические работают как обычно (ноль – ноль, не ноль – единица)

Выражения *expr* строятся следующим образом :

$$\begin{aligned} S_0 &\rightarrow : S : \\ S &\rightarrow number | ident | binop S S | unop S \end{aligned}$$

Пример: : \$PUDOVICHOK\$ * (\$@ruu@ + @\$S@ * \$NOL\$) || (\$@ruu@ & & \$NOL\$) :

Блоки кода должны начинаться на служебный символ { и заканчиваться на }. Внутренность же должна начинаться на ключевое слово и быть в должной конструкции, задаваемой оным. Обозначим за *blok*.

Ключевые слова – обособляются служебным символом *#*, то есть блок выглядит следующим образом {*#KEYWORD#body*}.

- *#PUSTO#* – пустой блок, кроме ключевого слова ничего не должно быть.
- *#ROBIT#*. Синтаксис:
#ROBIT#blok1 blok2 blok3.. – выполняет последовательно блоки.
- *#ZVYAZATI#*. Синтаксис:
#ZVYAZATI# ident num_expr – кладёт в переменную значение данного выражения.
- *#KOLI#*, *#TADI#*, *#PO – INOMU#*. Синтаксис :
#KOLI# expr #TADI# blok #PO – INOMU# blok – выполняет один из двух блоков, в зависимости от значения *log_expr*.
- *#PAKUL#*. Синтаксис:
#PAKUL# expr blok – выполняет блока, пока верно условие *expr*.
- *#CHITATSBERESTI#*. Синтаксис:
#CHITATSBERESTI#ident – считывает значение переменной.
- *#NAPISATNABERESTU#*. Синтаксис:
#NAPISATNABERESTU#expr – выводит на бересту значение выражения.

Для корректности программы достаточно одно условие – она начинается и заканчивается в одном блоке *#ROBIT#*

Примеры смотри в тестах парсера.

Пробелы, табы, переносы строки не поддерживаются(