Интерпретатор модельного языка программирования

Требуется разработать и реализовать интерпретатор модельного языка программирования на основе языка Паскаль. Инструментальный язык — C++.

Синтаксис модельного языка:

Синтаксис описан с помощью расширенной БНФ:

a) запись вида {α} означает итерацию цепочки α, т.е. в порождаемой цепочке в этом месте может находиться либо ε, либо α, либо αα, либо ααα и т.д.

b) запись вида [α] означает, что в порождаемой цепочке в этом месте может находиться либо α, либо ε.

c) жирным шрифтом выделены служебные слова модельного языка.

P → **program** D1 ; B.

D1 → eps | OD {, D1} | OP {, D1} | OF {, D1}

OD → **var** D {, D}

OP → **procedure** I (eps | DP {, DP}); OD; B

OF → **function** I (eps | DP {, DP}): [integer | boolean]; OD; B

D → I {, I}: [**integer** | **boolean** | **array** [N] **of** [**integer** | **boolean**]]

B → **begin** S {; S} **end**

S → I ([eps | I {, I}]) |

I := [E | **if** E **then** S **else** S | **while** E **do** S | B | **read** (I) | **write** (E)]

E → E1 [= | < | > | <= | >= | !=] E1 | E1

E1 → T { [+ | - | **or**] T }

T → F { [* | / | **and**] F }

F → I | I[N] | I ([eps | I {, I}]) | N | L | **not** F | (E)

L → **true** | **false**

I → a | b | ...| z | Ia | Ib |...| Iz | I0 | I1 |...| I9

N → 0 | 1 |...| 9 | N0 | N1 |...| N9

Набор операций и их старшинство:

- not (логическое отрицание)
- / (умножение и деление)
- + − (сложение и вычитание)
- < > <= >= == != (операции отношения)
- and (логическое умножение)
- or (логическое сложение)
- = (присваивание)

Семантика:

1. Любое имя, используемое в программе, должно быть описано и только один раз.

2. В операторе присваивания типы переменной и выражения должны совпадать.

3. В условном операторе и в операторе цикла в качестве условия возможно только логическое выражение.

4. Операнды операции отношения должны быть целочисленными.

5. Тип выражения и совместимость типов операндов в выражении определяются по обычным (паскалевским) правилам; старшинство операций задано синтаксисом.

6. В любом месте программы, кроме идентификаторов, служебных слов и числовых констант, может находиться произвольное число пробельных литер.

7. Внутри идентификаторов, служебных слов, числовых констант и разделителей, состоящих из нескольких символов, пробельные литеры недопустимы.

8. Между идентификаторами, числами и служебными словами должен находиться хотя бы один разделитель текста. Разделитель текста — это пробельная литера либо разделитель, определенный в алфавите языка (* / % + − < > <= >= == != , ; : ( )).

9. Семантика операторов if и while общепринятая; оператор read — оператор ввода значения переменной ⟨идентификатор⟩; write — оператор вывода значений списка выражений, указанных в круглых скобках. Числовые константы записываются в десятичной системе счисления.

Фазы работы интерпретатора модельного языка

Концептуально интерпретатором выполняются следующие фазы:

- лексический анализ,
- синтаксический анализ,
- семантический анализ (контроль контекстных условий),
- генерация программы на внутреннем языке (в качестве внутреннего языка предлагается использовать польскую инверсную запись — ПОЛИЗ),
- интерпретация программы на внутреннем языке

Конечный автомат с действиями, лежащий в основе лексического анализатора модельного языка