Семантические действия:

a - записать в ОПС идентификатор переменной

k - записать в ОПС константу (целую или вещественную) (будем хранить как значения а не как строки)

+, -, \*, /, i, <, >, !, == - записать в ОПС соответствующую бинарную операцию

w - записать в ОПС оператор записи

r - записать в ОПС оператор чтения

init - записать в ОПС оператор чтения инициализатора массива

GEN - записать в ОПС оператор генерации массива

j - безусловный переход по метке

jf - переход по метке при условии false

i - индексатор массива

Семантические программы:

Программа 1:

1) Записать значение лексемы в стек Data\_types

Программа 2:

1) Дописать в последний элемент стека Data\_types "arr"

Программа 3:

1) Достать значение из стека Data\_types

2) Переключение на заполнение таблицы переменных типа data\_type

3) Проверка имени, поступающего из входной цепочки, на совпадение с именами, ранее занесенными в таблицы. Если есть совпадение, то сигнализация об ошибке "повторное описание переменной"

4) Если ошибки нет, то добавление имени в одну из таблиц переменных (на которую ранее было переключение)

5) Добавить имя в ОПС

Программа 5:

1) Добавить операцию init в ОПС

2) Добавить оператор списка в ОПС

Программа 6:

1) В магазин меток записывается k

Программа 7:

1) В магазин меток записывается k

2) В ОПС записывается пустой элемент - место для будущей метки

3) В ОПС записывается операция jf - переход при условии false

Программа 8:

1) Через верхний элемент магазина меток, как ссылку на ранее заготовленное место для метки, записывается k + 2

2) В ОПС записывается метка, значение для которой читается из магазина меток

3) В ОПС записывается операция j – безусловный переход

Программа 9:

1) В магазин меток записывается k.

2) В ОПС записывается пустой элемент – место для будущей метки.

3) В ОПС записывается операция jf – переход при условии false.

Программа 10:

1) Через верхний элемент магазина меток, как ссылку на ранее заготовленное место для метки, записывается k.

Программа 11:

1) Через верхний элемент магазина меток, как ссылку на ранее заготовленное место для метки, записывается k + 2.

2) В магазин меток записывается k.

3) В ОПС записывается пустой элемент – место для будущей метки.

4) В ОПС записывается операция j – безусловный переход

Программа 12:

1) Достать значение из стека Data\_types

2) Переключение на заполнение таблицы переменных типа data\_type

3) Проверка имени, поступающего из входной цепочки, на совпадение с именами, ранее занесенными в таблицы. Если есть совпадение, то сигнализация об ошибке "повторное описание переменной"

4) Если ошибки нет, то добавление имени в одну из таблиц переменных (на которую ранее было переключение)

5) Добавить имя в ОПС

6) Добавить оператор $GEN в ОПС

Формат ОПС:

Представляет собой структуру с полями для всех типов, как для хранения строк так и для хранения значений констант к примеру, чтобы не конвертировать каждый раз, структура со значениями/строками.

В ОПС записываются следующие значения из TokenType:

PLUS,

MINUS,

MULTIPLY,

DIVIDE,

EQUALS,

LT,

GT,

EX,

числа,

идентификаторы переменных,

$JF,

$J

$i,

$r,

$w,

LIST,

$GEN

Чтобы различать типы значений, содержащихся в ОПС, используются следующие методы:

isOperator,

isInteger,

isReal,

isIntVariable,

isRealVariable,

isIntArrayVariable,

isRealArrayVariable,

isPointer

#Сделать массив или список из структур, в котором будут храниться значения переменных и констант