|  |  |
| --- | --- |
| **startDecl** | Запомнить тип в глобальную переменную **dataType**, установить флаг описания данных |
| **endDecl** | Сбросить флаг описания данных |
| **setIdent** | Заносим идентификатор в таблицу с типом **dataType**, проверяем дубликаты. Запоминаем переменную. |
| **startLevel** | Начать новый уровень в дереве. |
| **endLevel** | Вернуться к началу текущего уровня |
| **startFunc** | Создаем вершину – функцию (запоминаем тип возвращаемых данных), проверяем дубликаты. Запоминаем указатель на функцию в переменную «**k**». |
| **newBlack** | Новая черная вершина, необходима, чтобы отделить параметры функции от локальных переменных. |
| endFunc |  |
| **plusParam** | увеличиваем число параметров у функции (вершина с указателем «**k**»). **countParams++** |
| **find** | Ищем идентификатор в таблице. |
| **checkDubl** | Проверка идентификатора на дублирование. |
| **saveVariable** | -- |
| **matchLeft** | Контроль приведения типов при присваивании. |
| **match** | Контроль приведения типов в операциях. Достаем из магазина два типа, проверяем, возможно ли приведение типов. Кладем в магазин тип операции. |
| **push\_t** | Кладем в магазин тип текущей лексемы. |
| **callFunc** | Вызов функции. Запоминаем идентификатор вызываемой функции |
| **start\_parameter\_counting** | Начинаем подсчет параметров вызываемой функции. В переменную **parameter\_counting** записываем 0; |
| **end\_parameter\_counting** | Проверяем количество параметров, с сигнатурой вызываемой функции.  **parameter\_counting** должен быть равен **countParams** |
| **plus\_parameter\_counting** | инкрементируем количество параметров вызываемой функции. **parameter\_counting ++** |