# **Диагностика и нейтрализация**

Диагностика – это установка места возникновения и типа синтаксической ошибки. Кроме того, обработанная ошибка должна быть визуализирована пользователем в виде, удобном для ее обнаружения.

Нейтрализация предполагает исключение синтаксически неверной конструкции в тексте безболезненно для дальнейшего разбора всего текста [1].

## **Метод Айронса**

Н. Айронс в 1968 г. предложил метод локализации и отсечения «больных» кустов дерева при нисходящем разборе программы. Метод не претендует на универсальность, однако в нем выработаны здравые концепции нейтрализации ошибок при нисходящем разборе.

Основная идея – по контексту без возврата отбрасывать литеры, которые привели к тупиковой ситуации (когда продолжение анализа по грамматике невозможно), и продолжать разбор.

В общем случае обнаружение ошибки соответствует следующей схеме разбора:

где – построенная часть куста; – недостроенная часть куста, которую нельзя построить с помощью G[Z] [1].

При обнаружении ошибки (во входной цепочке в процессе разбора встречается символ, который не соответствует ни одному из ожидаемых символов), входная цепочка символов выглядит следующим образом: Tt, где T – следующий символ во входном потоке (ошибочный символ), t – оставшаяся во входном потоке цепочка символов после T. Алгоритм нейтрализации состоит из следующих шагов:

1. Определяются недостроенные кусты дерева разбора;

2. Формируется множество L – множество остаточных символов недостроенных кустов дерева разбора;

3. Из входной цепочки удаляется следующий символ до тех пор, пока цепочка не примет вид Tt, такой, что U => T, где U ∈ L, то есть до тех пор, пока следующий в цепочке символ T не сможет быть выведен из какого-нибудь из остаточных символов недостроенных кустов.

4. Определяется, какой из недостроенных кустов стал причиной появления символа U в множестве L (иначе говоря, частью какого из недостроенных кустов является символ U).

Таким образом, определяется, к какому кусту в дереве разбора можно «привязать» оставшуюся входную цепочку символов после удаления из текста ошибочного фрагмента.

## **Метод Айронса для автоматной грамматики**

Разрабатываемый синтаксический анализатор построен на базе автоматной грамматики. Реализация алгоритма Айронса для автоматной грамматики имеет следующую особенность.

Дерево разбора с использованием автоматной грамматики представлено на Рисунке 2.

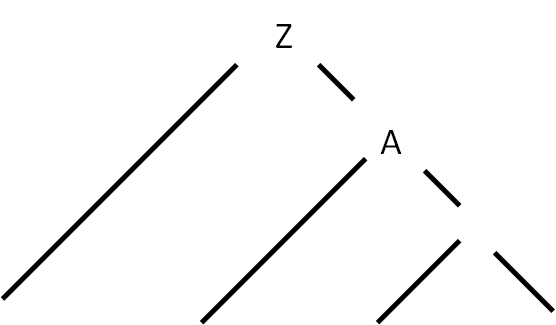


Рисунок 2 – Структура дерева разбора для автоматной грамматики

Таким образом, при возникновении синтаксической ошибки в процессе разбора с использованием автоматной грамматики, в дереве разбора всегда будет только один недостроенный куст (Рисунок 3).

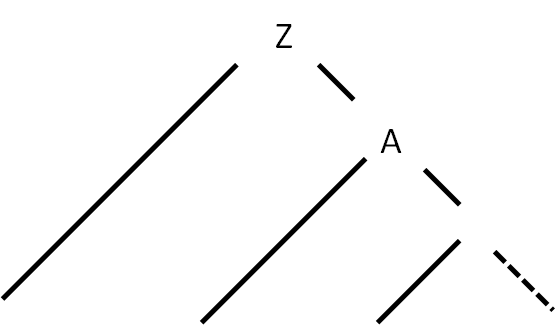


Рисунок 3 – Недостроенный куст при возникновении синтаксической ошибки (выделен пунктиром)

Поскольку единственный недостроенный куст – это тот, во время построения которого возникла синтаксическая ошибка, то это единственный куст, к которому можно привязать оставшуюся входную цепочку символов.

Обнаружение и нейтрализация ошибок для грамматики G[<Constant>] методом Айронса:

1. Некорректное начало модификатора «final»

* Обнаружение: шаги парсера для выявления некорректного начала оператора цикла в соответствии с грамматикой.
* Нейтрализация: методы предложения вариантов исправления ошибки, основанные на синтаксических правилах грамматики циклов и их корректных началах.

1. Некорректное определение типа переменной

* Обнаружение: как парсер определяет ошибку в левой части метода на основе грамматики.
* Нейтрализация: методы предложения возможных вариантов исправления ошибок в левой части метода, ориентированные на синтаксические структуры условий.

1. Некорректная переменная

* Обнаружение: как парсер определяет ошибку в правой части метода в контексте грамматики.
* Нейтрализация: методы исправления некорректных выражений в правой части метода, учитывая синтаксические структуры блоков команд.

1. Отсутствие знака присваивания (=) после переменной

* Обнаружение: описание алгоритма обнаружения данной ошибки в рамках грамматики.
* Нейтрализация: методы исправления ошибки в соответствии с правилами грамматики и подходы к автоматической или полуавтоматической коррекции.

1. Некорректное выражение в правой части метода

* Обнаружение: шаги парсера для выявления ошибок в правой части метода на основе грамматики.
* Нейтрализация: подходы к исправлению некорректных выражений в правой части метода, ориентированные на синтаксические правила языка программирования.

1. Отсутствие окончания выражения (;)

* Обнаружение: как парсер идентифицирует неправильный конец метода, учитывая структуру грамматики.
* Нейтрализация: подходы к автоматическому или полуавтоматическому исправлению ошибок в завершении метода с учётом грамматических правил.

Каждая ошибка описана с учетом её обнаружения и возможных путей исправления в контексте грамматики. Это поможет лучше понять, как парсер обрабатывает и исправляет подобные синтаксические ошибки.