

# Преобразования синтаксических деревьев при реализации синтаксического сахара в PascalABC.NET

Чумакова Евгения Геннадьевна

Научный руководитель:

доц., к.ф.-м.н. Михалкович С. С.

## Постановка задачи:

- ▶ Внедрить, используя методы реализации синтаксически сахарных расширений в компиляторах, следующие сахарные конструкции:
  - ▶ NULL-условный оператор (?.)
  - ▶ оператор объединения со значением NULL (??)
  - ▶ тип integer?
- ▶ Встроить эти конструкции в грамматику языка, определив уровень приоритета и ассоциируемость
- ▶ Решить проблему повторяющихся вычислений в секции операторов для операторов ?. и ??

# Актуальность.

## Понятие «синтаксического сахара»

- ▶ **Синтаксический сахар** – синтаксическая конструкция в языке программирования, дублирующая существующую
- ▶ «Сахар» выражается через конструкции базового языка
- ▶ Имеет простую синтаксическую форму. Упрощает восприятие
- ▶ Пример «синтаксического сахара»:

```
b := a > 0 ? c : 0;
```

Аналогичная запись с использованием конструкций базового языка:

```
if (a > 0) then  
    b := c  
else  
    b := 0;
```

# Реализация «сахара» в PascalABC.NET

## Общая структура компилятора PascalABC.NET



# Реализация «сахара» в PascalABC.NET

## Общая схема десахаризации — **DESUGARING**

- ▶ Исходный язык представлен базовыми узлами синтаксического дерева с типами —  $t_1 \dots t_n$ .
- ▶ После введения новых конструкций появляются сахарные синтаксические узлы с типами —  $s_1 \dots s_m$
- ▶ **Desugaring** происходит на этапе перехода от синтаксического дерева к семантическому при помощи специального визитора (конвертера деревьев), который обрабатывает сахарный узел:

```
visit(sug)
{
    Создание несакхарного узла unsug;
    Замена sug на unsug;
    Посещение подузлов узла sug;
}
```

# Реализация «сахара» в PascalABC.NET

**В какой момент делать необходимые семантические проверки?**

- ➔ **Способ 1.** Проверка в визиторе преобразования в семантическое дерево
- ➔ **Способ 2.** Проверка в специальном узле синтаксического дерева

# Реализация «сахара» в PascalABC.NET

**Семантический Desugaring** — десахаризация и проверки осуществляются в визиторе преобразования в семантическое дерево



# Реализация «сахара» в PascalABC.NET

**Семантический Desugaring** — десакхаризация и проверки осуществляются в визиторе преобразования в семантическое дерево

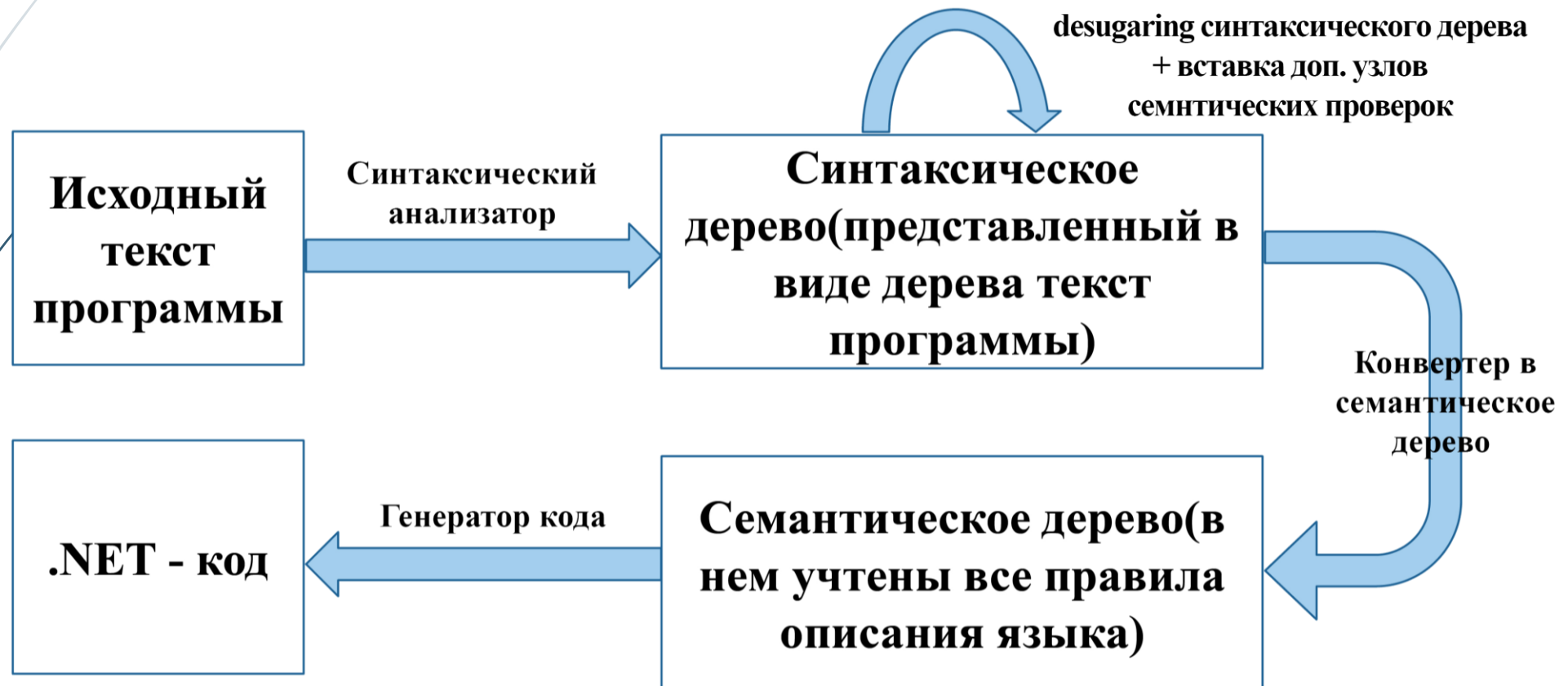
```
visit (sug:sug_node)
{
    SemCheckOrError(sug); //семантическая проверка
    var unsug = new unsug_node(sug);
    Замена sug на unsug;
    visit (unsug);
}
```

Замена sug на unsug осуществляется на случай повторного обхода синтаксического дерева



# Реализация «сахара» в PascalABC.NET

**Синтаксический Desugaring** — десакхаризация осуществляется в визиторе, преобразующем синтаксическое дерево



# Реализация «сахара» в PascalABC.NET

**Синтаксический Desugaring** — десакхаризация осуществляется в визиторе, преобразующем синтаксическое дерево

```
visit (sug:sug_node)
{
    var check = sem_check_statement(sug);
    var unsug = new unsug_node(sug);
    Замена sug на (check, unsug);
    visit (unsug);
}
```

`sem_check_statement` – конструирование специального узла семантических проверок

# Введение pull-условного оператора ?.

## Структура:

```
[операнд1] ?. [операнд2]
```

## 1. Изменения грамматики языка

### ▶ Новый терминальный символ:

```
"?." { return (int)Tokens.tkQuestionPoint; }
```

### ▶ Новые лексемы:

```
%token <ti> tkQuestionPoint  
%type <ex> var_question_point
```

### ▶ var\_question\_point имеет тип ex, т.е. expression – тип выражения.

# Введение null-условного оператора ?.

## 1. Изменения грамматики языка

- ▶ NULL-условный оператор ?. имеет наивысший приоритет, поэтому он вводится на одном уровне с оператором @.
- ▶ Грамматические правила:

```
var_question_point
: variable tkQuestionPoint variable
| variable tkQuestionPoint var_question_point
;
```

# Введение pull-условного оператора ?.

## 2. Генерация нового узла синтаксического дерева

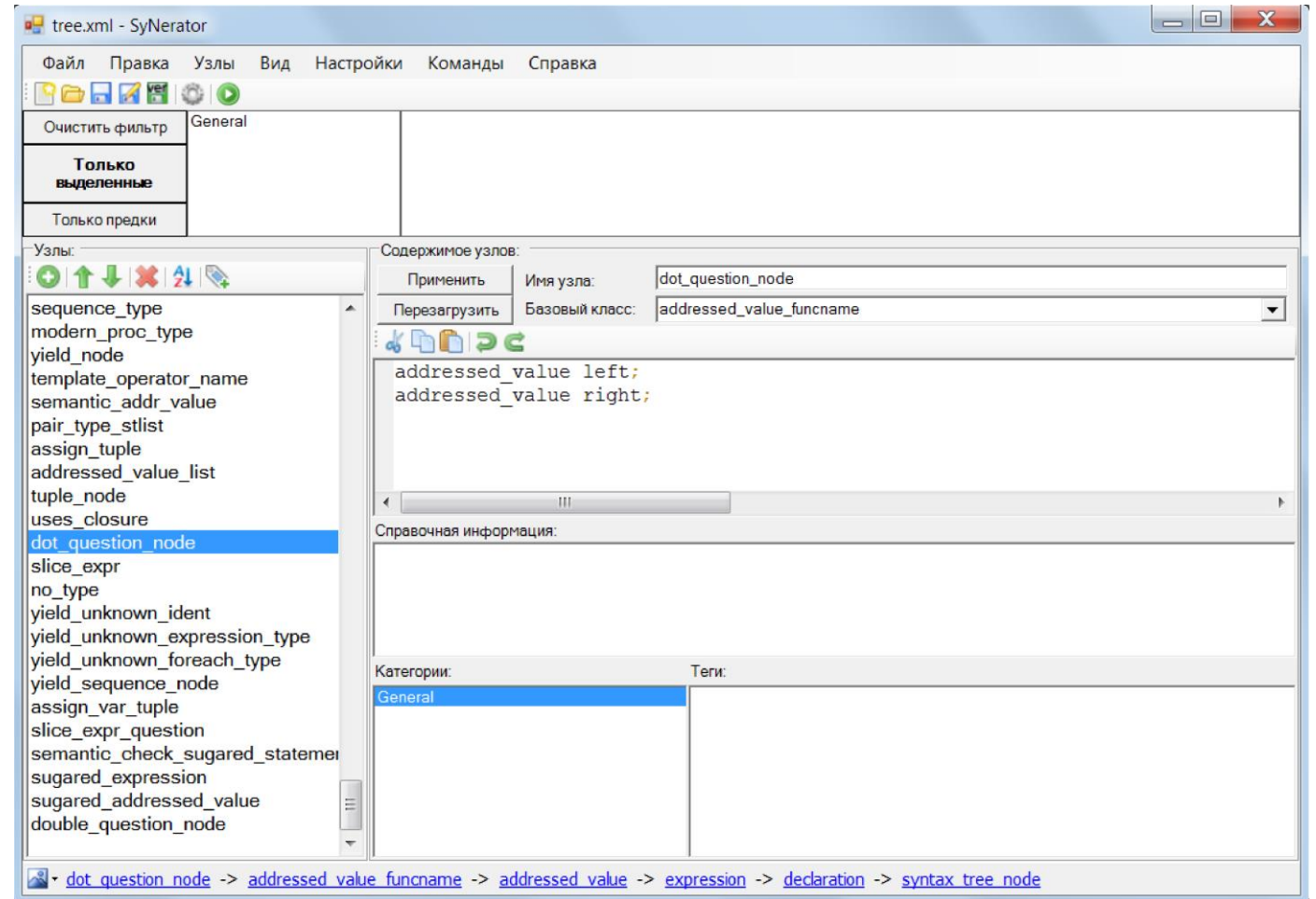
➔ НОВЫЙ узел —

`dot_question_node`

➔ ПОДУЗЛЫ,  
СООТВЕТСТВУЮЩИЕ  
ЛЕВОМУ И ПРАВОВОМУ  
ОПЕРАНДАМ:

`addressed_value left;`

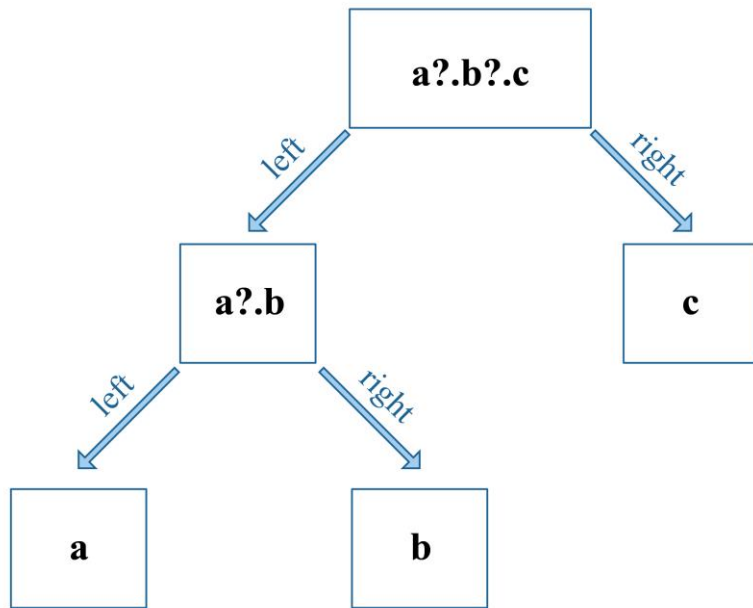
`addressed_value right;`



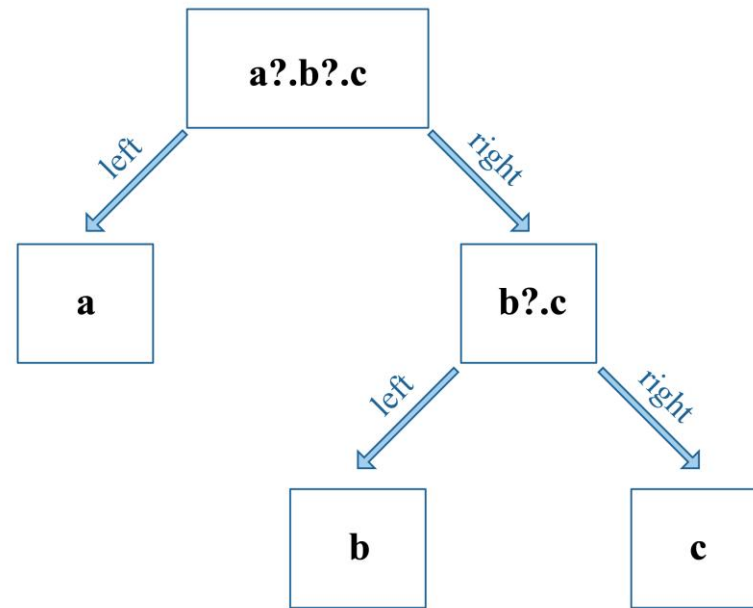
# Введение null-условного оператора ?.

## 3. Десахаризация. Проблема ошибочного ассоциирования

```
a ?. b → a = nil ? nil : a.b
```



Корректный разбор



Ошибочный разбор

# Введение pull-условного оператора ?.

## 3. Десахаризация. Проблема ошибочного ассоциирования

Пример:  $a?.b( ) [ ] ^$

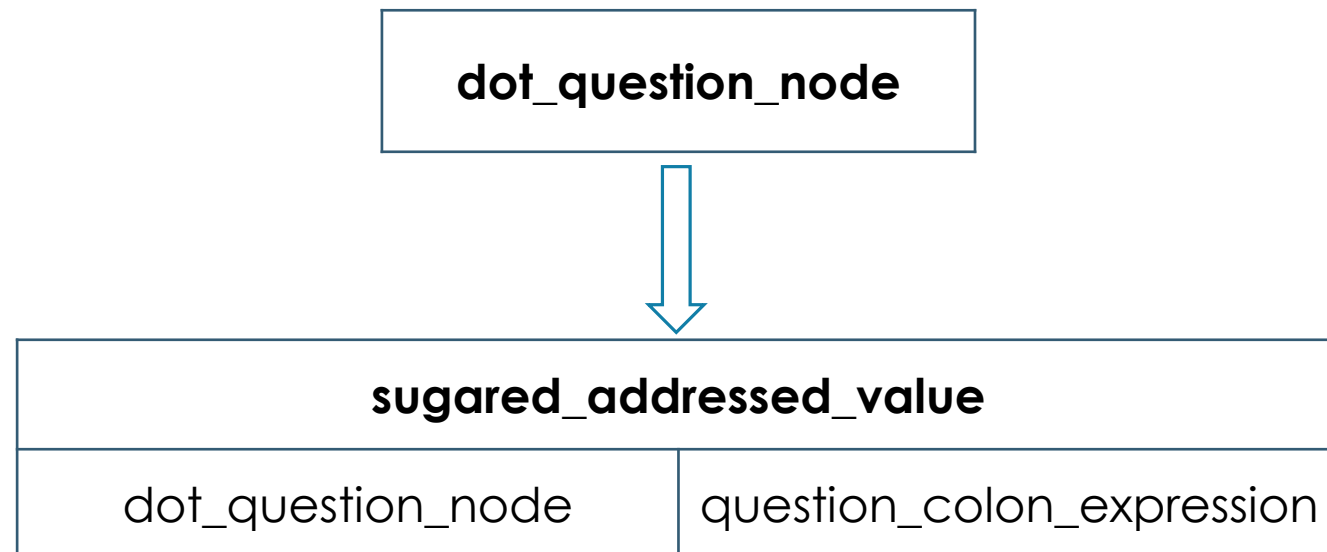
Задача:  $a?.(((b)( ) [ ] ) ^)$

- $(a?.b( ) [ ] ) ^$
- $(a?.b( ) ) [ ] ^$
- $(a?.b) ( ) [ ] ^$
- $a.b( ) [ ] ^$

# Введение null-условного оператора ?.

## 4. Семантические проверки

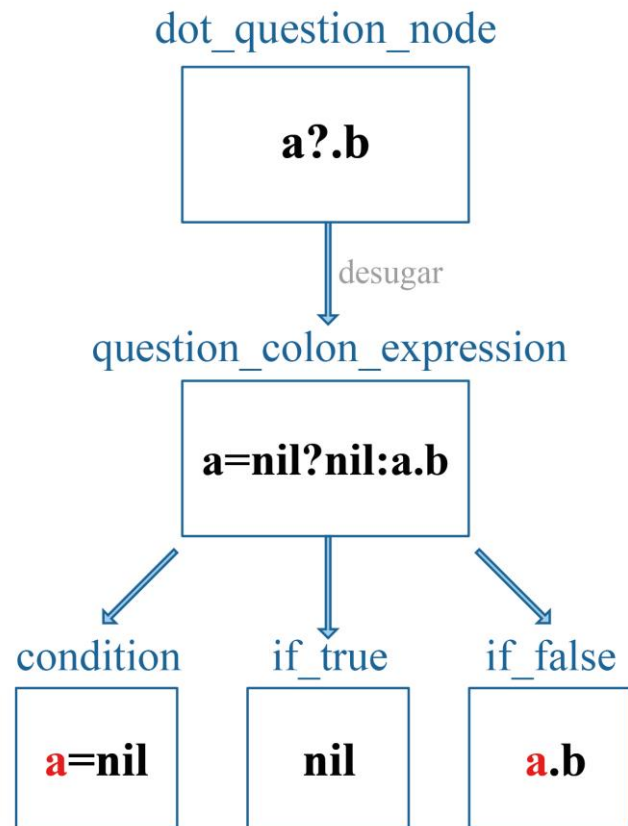
- ▶ Левый операнд должен иметь ссылочный тип
- ▶ Если правый операнд имеет размерный тип T, то преобразовать его к типу `Nullable <T>`





# Введение null-условного оператора ?.

## 5. Проблема повторяющихся вычислений



**Решение:** создание новой уникальной переменной. Тогда один оператор, содержащий ?. :

```
var c := a?.b ;
```

заменяется на:

```
begin
  var #tt := a ;
  var c := #tt?.b ;
end;
```

**Недостаток:** только для секции операторов

# Введение оператора объединения со значением `null` — ??

- ▶ Структура:

```
[операнд1] ?. [операнд2]
```

- ▶ Оператор возвращает левый операнд, если он не равен `null`; в противном случае возвращается правый операнд.
- ▶ Пара операторов `?.` и `??` — это удобно:

Синтаксический сахар:

```
var d := a ?. b ?? c;
```

Конструкция базового языка:

```
var d := (a = nil ? nil : a.b) = nil ? c : a.b
```

# Типы, допускающие значение null

```
var s: string := nil; // Допустимо, строка - ссылочный тип
var i: integer := nil; // Ошибка. Операции над размерными
                        // типами и nil недопустимы
```

```
var i : System.Nullable<integer>;
i := nil; // Допустимо
```

## Синтаксический сахар:

```
var i: integer? := nil;
```

- ▶ Десахаризации происходит на самом раннем этапе – при построении синтаксического дерева программы
- ▶ Семантические проверки: тип должен быть размерным

## Что реализовано:

- ✓ Введены с использованием методов реализации синтаксически сахарных расширений в компиляторах следующие сахарные конструкции:
  - NULL-условный оператор (?.)
  - оператор объединения со значением NULL (??)
  - тип integer?
- ✓ Перечисленные конструкции встроены в грамматику языка, определен уровень приоритета и ассоциируемость
- ✓ Решена проблема повторяющихся вычислений в секции операторов для операторов ?. и ??