

Реализация полиномиальной сложности оптимальных принтер-комбинаторов с выбором на платформе .NET

Булгаков Андрей Вадимович
Математико-Механический факультет СПбГУ

Научный руководитель:
старший преподаватель кафедры системного
программирования
Григорьев С. В.

Предметная область

- Синтаксический анализ кода
- Работа с абстрактным синтаксическим деревом
- Преобразование
- **Вывод кода**

Какой вариант удобнее?

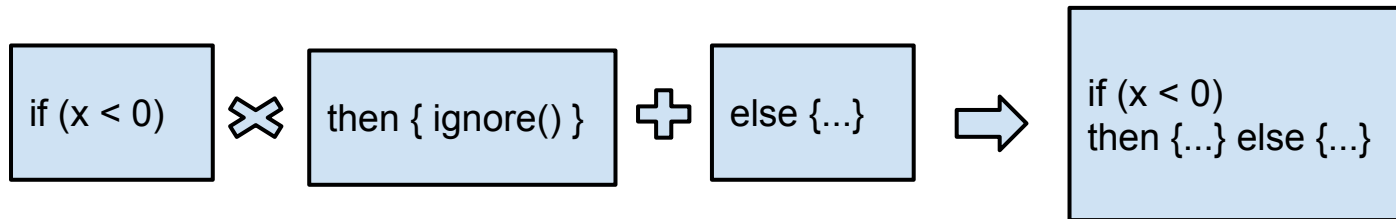
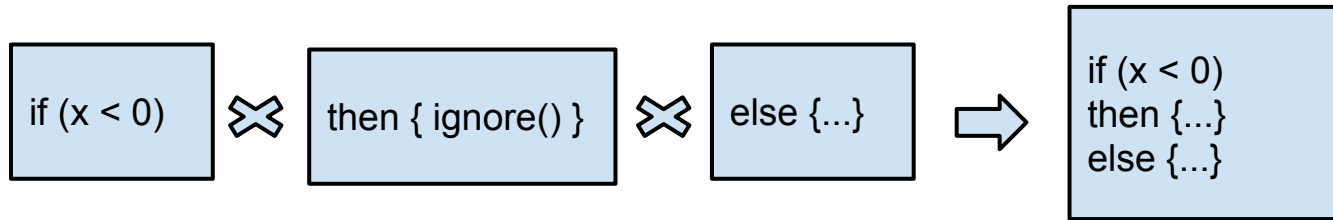
```
1 read(x);read(n);res:=1;
2 while (n > 0) {res:=res*x;n:=n-1;
3 if (n % 2 = 0) {write(n);}else
4 {write(n + 1);}}
5 write(res);
6
7
8
9
10
```

Вариант 1

```
1 read(x);
2 read(n);
3 res := 1;
4 while (n > 0) {
5     res := res * x;
6     n := n - 1;
7     if (n % 2 = 0) {write(n);}
8     else {write(n + 1);}
9 }
10 write(res);
```

Вариант2

Комбинаторы



Оптимальность

- Локальная
- Глобальная

Аналоги

FSharpX.Text.StructuredFormat

- + Высокая скорость работы**
- Локально оптимальна**



Polynomial Pretty Printer Combinators

- + Глобально оптимальна**
- Не существует аналога на .NET**

Задачи

- 1) Реализовать принтер-комбинаторную библиотеку для платформы .NET
- 2) Реализовать алгоритм принтера, работающий за полиномиальное время
- 3) Получить оптимальный вывод
- 4) Протестировать библиотеку в реальных условиях

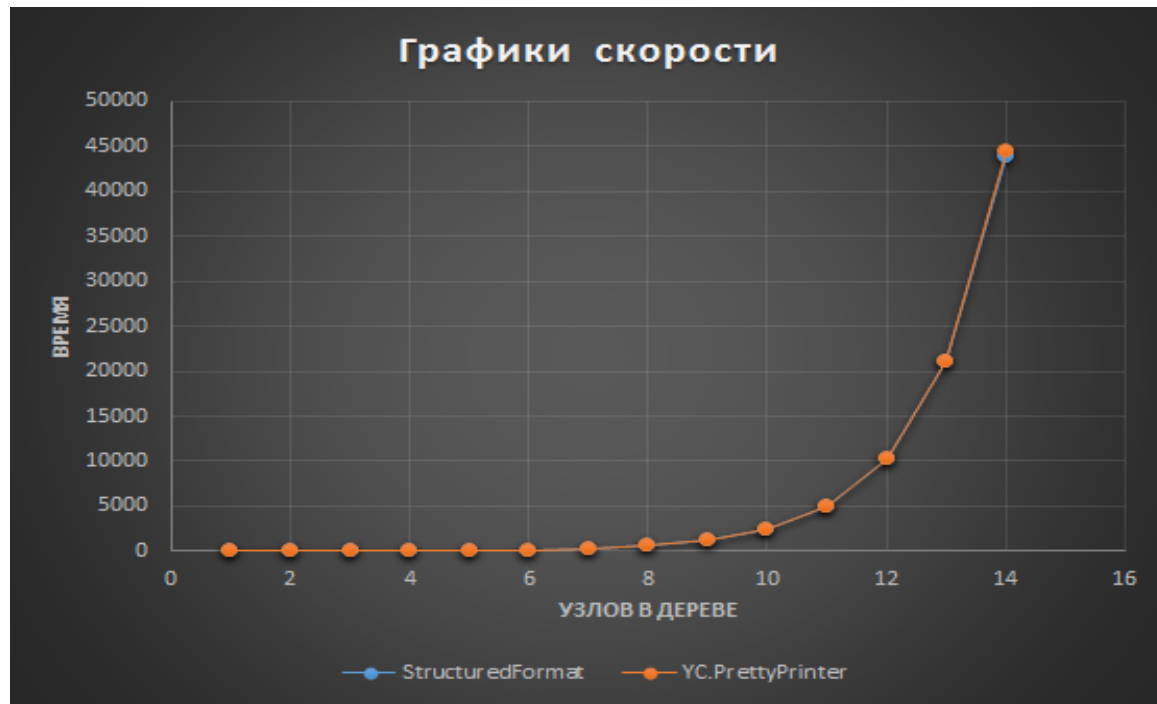
Реализация

- Изучение библиотеки **Polynomial Pretty Printer Combinators** и документации к ней
- Реализация аналога на .NET
- Учесть особенности платформы

Что выбрать?

- **Map** - Неизменяем. Имеет структуру AVL дерева. $O(\log n)$
- **Dictionary** - Изменяем. Имеет структуру хештаблиц. $O(1)$

Производительность



Результаты

- Реализована принтер-комбинаторная библиотека для платформы .NET
- Проведены тесты сравнения производительности с Text.StructuredFormat
- Пройдены тесты синтаксической и семантической правильности вывода на библиотеки Brahma.FSharp