

# Форматирование текста программ на основе комбинаторов, сопоставления с образцом и синтаксических шаблонов

Подкопаев Антон, студент 445 группы, podkoav239@gmail.com

СПбГУ

Научный руководитель: к.ф.-м.н. Булычев Д.Ю.

15 апреля 2013

## Языковые процессоры

## Языковые процессоры

- Компиляция
- Суперкомпиляция
- Интерпретация
- Статический анализ кода
- Декомпиляция
- Рефакторинг
- Реинжиниринг

# План работы ЯП

- Синтаксический анализ
- Преобразование
- Представление результата

# План работы ЯП

- Синтаксический анализ
- Преобразование
- Представление результата
  - Код программы
  - ...

- Компиляция
- Суперкомпиляция
- Интерпретация
- Статический анализ кода
- Декомпиляция
- Рефакторинг
- Реинжиниринг

- Компиляция
- Суперкомпиляция
- Интерпретация
- Статический анализ кода
- Декомпиляция
- Рефакторинг
- Реинжиниринг

Почему нельзя просто взять и напечатать?



Почему нельзя просто взять и напечатать?

```
int foo(int k){if(k<1||k>2){printf("out of range\n");  
printf("this function requires a value of 1 or 2\n");}else{  
printf("Switching\n");switch(k){case 1:printf("1\n");break;case  
2:printf("2\n");break;}}}
```

```
if e then s1 else s2
```

```
if e then s1  
    else s2
```

```
if e  
then s1  
else s2
```

# Существующие решения

## Комбинаторы

$$\boxed{\text{if } e} \langle + \rangle \boxed{(\text{then } s1 \text{ } \$\$ \text{ else } s2)} = \boxed{\begin{array}{l} \text{if } e \text{ then } s1 \\ \text{else } s2 \end{array}}$$

$$\boxed{\text{if } e \text{ } \$\$ \text{ then } s1} \langle + \rangle \boxed{\text{else } s2} = \boxed{\begin{array}{l} \text{if } e \\ \text{then } s1 \text{ else } s2 \end{array}}$$

Построение документов с помощью некоторого набора комбинаторов

# Пример комбинаторного принтера

```
docFromOperation :: Operation -> Doc
docFromOperation (Read varName) =
    (text "read(" <> (text varName) <> text(")")
docFromOperation (Write exp) =
    (text "write(" <> (docFromExpression exp) <> text(")")
docFromOperation (IfThenElse exp trueOp falseOp) =
    (text "if" <+> sep[(docFromExpression exp),
        (text "then" <+> (docFromOperation trueOp),
        (text "else" <+> (docFromOperation falseOp))]
docFromOperation (exp 'WhileDo' loopOp) =
    (text "while" <+> sep [(docFromExpression exp),
        (text "do" <+> (docFromOperation loopOp))]
```

## Шаблоны

```
t_start  
write(@-)  
t_end
```

```
t_start  
write(@  
    @  
)  
t_end
```

- Получение расширенным парсером языка образцы
- Составление документа по дереву

## Небольшой Pascal-like язык

```
{  
  read (k);  
  read (n);  
  r := 1;  
  while k > 0 do {  
    if k % 2 != 0 then r := r * n else skip;  
    n := n * n;  
    k := k / 2  
  };  
  write (r)  
}
```

# Пример шаблонов

```
while @- do while @- do
```

```
@|
```

```
@|
```

```
while @- do
```

```
@-
```

```
while @|
```

```
@|
```

```
do
```

```
@|
```

```
@|
```



- Доказательство состоятельности подхода
- Реализация подхода для языка  $L$