

# **Обзор методов одновременного задания синтаксических анализаторов и принтеров для различных языков программирования**

Студент: Алиев Мирза, 344 группа  
Научный руководитель: асп. каф. СП Подкопаев А.В.

2015

# Применимость

- Совмещение схожих структур
- Проще поддерживать программу
- Сохранение инварианта

# Цель

- Получение знаний в данной предметной области на основе изучения предложенных статей
- Выявление особенностей подходов с точки зрения форматирования
- Реализации для учебного языка L

# Статъи

- T. Rendel K. Ostermann: Invertible Syntax Descriptions: Unifying Parsing and Pretty Printing, 2010
- K. Matsuda M. Wang: FliPpr: A Prettier Invertible Printing System, 2013
- M. Boespflug: Functional Pearl: Replaying the stack for parsing and pretty printing, 2012

# Статья Т. Rendel К. Ostermann

parserSK = exp1 where

exp0 = s <\$ tok "S"

<|> k <\$ tok "K"

<|> tok "(" \*> exp1 <\* tok ")"

exp1 = foldl app <\$> exp0 <\*> many exp0

tok k = string k <\*> spaces

# Статья Т. Rendel К. Ostermann

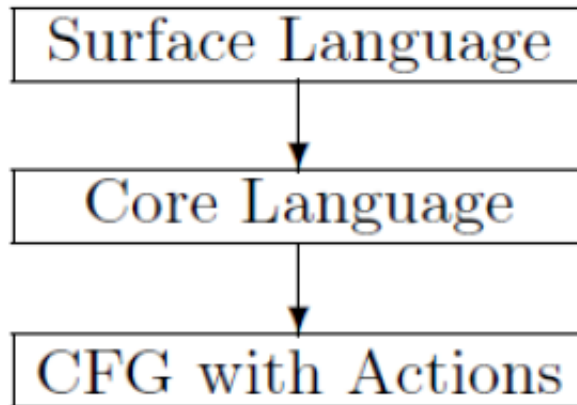
**data** *Iso*  $\alpha \beta$   
= *Iso* ( $\alpha \rightarrow \text{Maybe } \beta$ ) ( $\beta \rightarrow \text{Maybe } \alpha$ )

*inverse* :: *Iso*  $\alpha \beta \rightarrow \text{Iso } \beta \alpha$   
*inverse* (*Iso*  $f g$ ) = *Iso*  $g f$

*apply* :: *Iso*  $\alpha \beta \rightarrow \alpha \rightarrow \text{Maybe } \beta$   
*apply* (*Iso*  $f g$ ) =  $f$

*unapply* :: *Iso*  $\alpha \beta \rightarrow \beta \rightarrow \text{Maybe } \alpha$   
*unapply* = *apply*  $\circ$  *inverse*

# Статья К. Matsuda M. Wang



# Статья М. Voespflug

```
data K7 a b c d = K7 { sideA :: a -> b, sideB :: d -> c }
```

```
type C r = (String -> r) -> String -> r
```

```
type PP a = forall r r'. K7 (C (a -> r)) (C r) (C (a -> r')) (C r')
```

```
play :: K7 a b c d -> a -> b
```

```
    play csst = sideA csst
```

```
flip :: K7 a b c d -> K7 d c b a
```

```
    flip (K7 f g) = K7 g f
```

```
parse :: PP a -> String -> Maybe a
```

```
    parse csst = play csst (\_ _ x -> Just x) (const Nothing)
```

```
pretty :: PP a -> a -> Maybe String
```

```
    pretty csst = play (flip csst) (const Just) (\_ _ -> Nothing) ""
```



# Сравнение подходов

	Т. Rendel K. Ostermann	К. Matsuda M. Wang	М. Boespflug
Использование ширины вывода	Нет	Да	Нет
Что нужно реализовать	Синтаксический анализатор	Принтер	Синтаксический анализатор
Язык L (строк кода)	Около 250	Около 60	Около 230

# Сравнение производительности

	T. Rendel K. Ostermann	M. Boespflug
100 элементов	3.1127e-2	1.092e-3
1000 элементов	1.228305	7.031e-3
2000 элементов	5.772979	1.3437e-2

	T. Rendel K. Ostermann	M. Boespflug
100 элементов	5.282e-3	9.91e-4
1000 элементов	3.2422e-2	1.0082e-2
10000 элементов	0.452966	8.1065e-2

# Итоги

- Произведен обзор данных статей.
- На основе каждого подхода реализован синтаксический анализатор и принтер для учебного языка L.
- Были выявлены особенности, связанные с форматированием.
- Произведено сравнение производительности двух схожих подходов