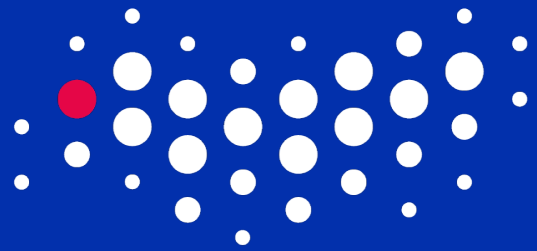
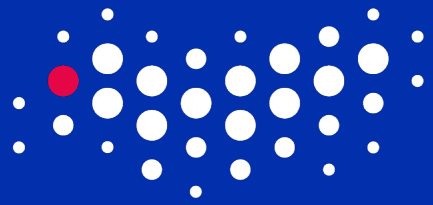


УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2017



УНИВЕРСИТЕТ ИТМО

Синтаксически-управляемый статический анализ

Выполнил Дмитрий Халанский
Научный руководитель: Лаздин Артур Вячеславович

Санкт-Петербург, 2017

УНИВЕРСИТЕТ ИТМО

Синтаксически-управляемый статический анализ

Выполнил Дмитрий Халанский

Научный руководитель: Лаздин Артур Вячеславович

Языково-ориентированная разработка

Зачем может понадобиться разрабатывать новый язык:

- Реализация текстового пользовательского интерфейса;
- Введение языковых примитивов, соответствующих предметной области;
- Проверка идей при разработке алгоритмов, оперирующих формальными языками:
  - Анализаторов;
  - Трансляторов;
  - Интерпретаторов...

Упрощённая обработка формальных языков

Обычно:

$$\text{``a := 2 + 3''} \xrightarrow{AST}$$
$$\texttt{(:= a (+ 2 3))} \xrightarrow{+\ 2\ 3 == 5}$$
$$\texttt{(:= a 5)} \xrightarrow{x{:=}y => x == y}$$
$$\texttt{a == 5}$$

Предлагается:

$$\text{``a := 2 + 3''} \xrightarrow{f(\text{``2''})=2,\ f(\text{``3''})=3,\ g(\text{``+''})=+}$$
$$\text{``a'' ``:='' 5} \xrightarrow{h(\text{``:=''}) = set-env}$$
$$\texttt{a == 5}$$

Сравнение с другими нотациями

	BNF	Wirth	Parsec	Perl6	Coq	bison	Marpa
Читаемость	+	+	+	-	-	+	-
Есть вычисления над термами	-	-	±	+	+	-	+
Грамматика обособлена от обработчика	+	+	-	-	-	+	-
Схожесть с BNF	+	+	+	-	-	+	+

Цели и задачи

Цель Разработка нотации для представления вычислений над языковыми конструкциями, а также статического анализатора в терминах полученной нотации для проверки её преимуществ и недостатков;

Задачи

- Выбор базового языка для представления грамматики;
- Выбор языка для представления вычислений;
- Комбинация этих языков и исключение недетерминизма при обработке;
- Разработка статического анализатора в терминах полученной нотации.

Выбор языка для грамматики

	BNF	ABNF	EBNF	Wirth
Наличие комментариев	-	+	+	+
Выразительность	-	++	+	+
Распространённость	+	+	+	-

Выбор языка для вычислений

	Lisp	Haskell	Python	C	Java
Популярность в среде ЯП	+	+	-	+	-
Простота реализации	+	+	+	-	-
Мало аннотаций типов	+	-	+	-	-
Поддержка ФП	+	+	+	-	-
Независимость от выравнивания	+	-	-	+	+

Пример текста

```
bin = "zero" : (identity 0)
    / "one"  : (identity 1)

top = bin{a} bin{b} :
        (if (= 0 (+ a b))
            (error "top" "zero zero")
            (* a b))
```

Передача параметров

Несколько правил с примерами возможных входных данных к ним:

```
top1 = bin LWSP bin
     ; param = '(0 "     " 1)
top2 = bin LWSP bin{a}
     ; param = '(1), a = 1
top3 = bin{a} LWSP ([bin]){b}
     ; param = '(1 ()), a = 1, b = '()
```

Среда

`top1 = bin{a:1} LWSP bin{b:2}` — сначала выполнится первый `bin`, затем второй.

`top2 = bin{a:1} LWSP bin{b:2} SP bin{c:3:1}` — выполнить первый `bin`; затем второй, проигнорировать изменения, внесённые им в среду; выполнить третий.

Интеграция со Scheme

Окружение, в котором выполняется каждая функция:

```
(let ([env internal_env] [internal_env '()])
  (bind-args
    (lambda (param) (body)))
  (if env-mutable (set! internal_env env)))
```

Ограничение на значения в среде

bad может выполнить произвольный, в том числе меняющий среду, код после такого прыжка, хотя и заявлено, что это чистое правило.

```
bad = "3" :
  ((env _произвольный_код_))

dirty = "4" :
  ((call/cc (lambda (x)
              (set! env x)
              '(identity 0))))
```

Статический анализатор

```
@a = rand bits 4; // Random 4-bit number
@d = inf 0;       // sum from 0 to @a
for @i to (@a) do
    @d = @d + inf @i
done
```

$\Rightarrow a \in [0; 15], d \in [0; 195]$

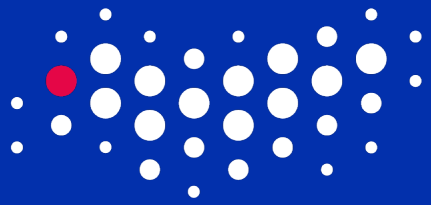
Преимущества и недостатки

Преимущества:

- Простота;
- Интуитивность;
- Гибкость.

Недостатки:

- Слабо подходит для нелинейного анализа, которому в самом деле требуется AST;
- Большие грамматики сложно читать и без привязки к вычислениям.



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

Санкт-Петербург, 2017