

# Самоприменимый специализатор для машинного кода

## предварительные результаты

Юрий Кравченко

руководитель Березун Даниил Андреевич

СПбАУ

2 апреля 2018 г.

# Специализатор

## Обозначения

$$\llbracket p \rrbracket_L[a_1, a_2, \dots, a_n] = b$$

# Специализатор

## Обозначения

$$\llbracket p \rrbracket_L[a_1, a_2, \dots, a_n] = b$$

## Определение

Программу *spec* назовём специализатором, если

$$\llbracket p \rrbracket_{L_1}[d_1, d_2] = d_3$$

$$\llbracket spec \rrbracket_{L_2}[p, d_1] = q$$

$$\llbracket q \rrbracket_{L_1}[d_2] = d_3$$

# Третья проекция Футамуры

## Утверждение

$\llbracket \text{spec} \rrbracket_L[\text{spec}, \text{spec}] = \text{compgen}$

$\llbracket \text{compgen} \rrbracket_L[\text{interpreter}] = \text{compiler}$

# В чём подвох?

# В чём подвох?

- ▶ Тривиальный спес

# В чём подвох?

- ▶ Тривиальный спрес
- ▶ Неразрешимость

# В чём подвох?

- ▶ Тривиальный спрес
- ▶ Неразрешимость
- ▶ Сложно реализовать



# Идея

# Идея

- ▶ Специализатор для машинного кода

# Идея

- ▶ Специализатор для машинного кода
- ▶ Специализатор на СИ

# Идея

- ▶ Специализатор для машинного кода
- ▶ Специализатор на СИ
- ▶ По аналогии со специализатором для Flow Chart

# Процесс разработки

# Процесс разработки

- ▶ Несколько версий архитектуры

# Процесс разработки

- ▶ Несколько версий архитектуры
- ▶ Поэтапная разработка

# Процесс разработки

- ▶ Несколько версий архитектуры
- ▶ Поэтапная разработка
- ▶ Реализованы самые популярные операторы



# Интерпретация

```
1 void sort(int len, int* a) {  
2     for (int i = 0; i < len; ++i) {  
3         for (int j = i + 1; j < len; ++j) {  
4             if (a[i] > a[j]) {  
5                 int k = a[i];  
6                 a[i] = a[j];  
7                 a[j] = k;  
8             }  
9         }  
10    }  
11 }
```

# Работа с памятью

```
1 char* eratosphen(int n) {  
2     char* a = my_malloc(n);  
3     for (int i = 0; i < n; ++i) {  
4         a[i] = 1;  
5     }  
6     a[0] = 0;  
7     a[1] = 0;  
8     for (int i = 2; i < n; ++i) {  
9         if (a[i]) {  
10             for (int j = i * i; j < n; j += i) {  
11                 a[j] = 0;  
12             }  
13         }  
14     }  
15     return a;  
16 }
```

# Простая специализация

```
1 int dict(int len, int* a, int* b, int c) {  
2     for (int i = 0; i < len; ++i) {  
3         if (a[i] == c) {  
4             if (b[i] == 0) {  
5                 return 1;  
6             }  
7             else {  
8                 return 2;  
9             }  
10        }  
11    }  
12    return 3;  
13 }
```

# Полный цикл обработки

```
1 int pow(int a, int b) {  
2     if (b == 0) {  
3         return 1;  
4     }  
5     return my_pow(a, b - 1) * a;  
6 }
```

Coming soon



Конец

<https://github.com/XJIE6/spec>