



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Факультет компьютерных наук
Департамент программной инженерии
Отчет по преддипломной практике

Программа поведенческого анализа вполне
структурированных систем переходов

Выполнил студент группы БПИ131
образовательной программы
09.03.04 «Программная инженерия»

Михайлов В. Е.

Научный руководитель:
Старший преподаватель
Дворянский Л. В.

Формальная верификация

Формальное доказательство соответствия или несоответствия формального предмета верификации его формальному описанию

Примеры объектов верификации:

- Исходные тексты программ
- Криптографические протоколы
- Протоколы передачи данных
- Логические схемы

Цель работы

Разработать программу для поведенческого анализа вполне структурированных систем переходов.

Задачи работы

1. Реализовать алгоритмы поведенческого анализа: Метод насыщения и Покрывающее дерево системы переходов
2. Разработать язык описания систем переходов WSTSL, основанный на языке SETL
3. Поставить эксперимент для изучения возможных путей совершенствования реализованных алгоритмов
4. Разработать техническую документацию

Глава 1. Вполне структурированные системы переходов (95%)

Глава 2. Алгоритмы поведенческого анализа

Метод насыщения (100%)

Покрывающее дерево системы переходов (85%)

Глава 3. Архитектура системы (90%)

Глава 4. Описание языка WSTSL (30%)

Глава 5. Эксперимент (60%)

Приложение А.

Техническое задание (90%)

Программа и методика испытаний (0%)

Руководство оператора (0%)

Текст программы (80%)

Вполне структурированные системы переходов

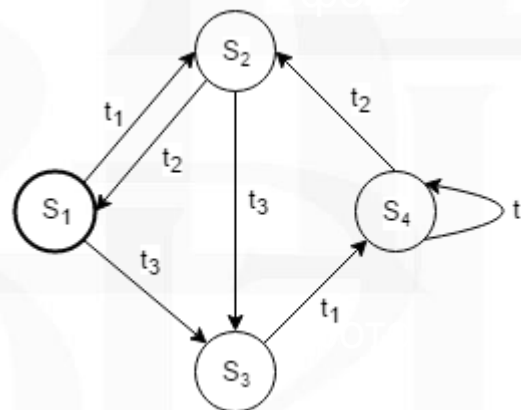
Вполне структурированной системой переходов называется система переходов $LTS = (S, T, \rightarrow, s_0)$, дополненная отношением квазипорядка $\leq \subseteq S \times S$, удовлетворяющая следующим двум условиям:

- 1) Отношение \leq является правильным квазипорядком;
- 2) Квазипорядок \leq совместим с отношением переходов \rightarrow , а именно, для любых состояний $s \leq q$ и перехода $s \xrightarrow{t} s'$ существует переход $q \xrightarrow{t} q'$ такой, что $s' \leq q'$.

Вполне структурированные системы переходов

Вполне структурированной системой переходов называется **система переходов** $LTS = (S, T, \rightarrow, s_0)$, дополненная отношением квазипорядка $\leq \subseteq S \times S$, удовлетворяющая следующим двум условиям:

- 1) Отношение \leq является правильным квазипорядком;
- 2) Квазипорядок \leq совместим с отношением переходов \rightarrow , а именно, для любых состояний $s \leq q$ и перехода $s \xrightarrow{t} s'$ существует переход $q \xrightarrow{t} q'$ такой, что $s' \leq q'$.



Вполне структурированные системы переходов

Вполне структурированной системой переходов называется система переходов $LTS = (S, T, \rightarrow, s_0)$, дополненная отношением **квазипорядка** $\leq \subseteq S \times S$, удовлетворяющая следующим двум условиям:

- 1) Отношение \leq является правильным квазипорядком;
- 2) Квазипорядок \leq совместим с отношением переходов \rightarrow , а именно, для любых состояний $s \leq q$ и перехода $s \xrightarrow{t} s'$ существует переход $q \xrightarrow{t} q'$ такой, что $s' \leq q'$.

Квазипорядок (предпорядок) \leq на множестве M :

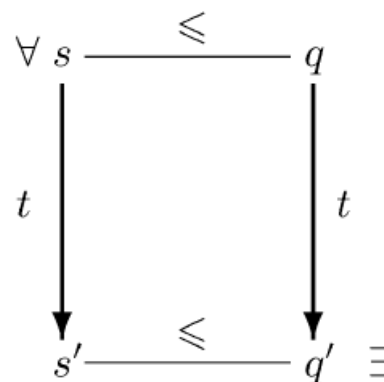
$$\forall a \in X: a \leq a$$

$$\forall a, b, c \in X: (a \leq b \wedge b \leq c) \Rightarrow (a \leq c)$$

Вполне структурированные системы переходов

Вполне структурированной системой переходов называется система переходов $LTS = (S, T, \rightarrow, s_0)$, дополненная отношением квазипорядка $\leq \subseteq S \times S$, удовлетворяющая следующим двум условиям:

- 1) Отношение \leq является правильным квазипорядком;
- 2) Квазипорядок \leq **совместим с отношением переходов \rightarrow** , а именно, для любых состояний $s \leq q$ и перехода $s \xrightarrow{t} s'$ существует переход $q \xrightarrow{t} q'$ такой, что $s' \leq q'$.



ВЫБОР МОДЕЛЕЙ, МЕТОДОВ И АЛГОРИТМОВ

- 1) Покрывающее дерево системы переходов
- 2) Метод насыщения

Покрывающее дерево системы переходов

- 1) Вершины дерева помечены состояниями системы переходов

Покрывающее дерево системы переходов

- 1) Вершины дерева помечены состояниями системы переходов
- 2) Каждая вершина является живой, либо мертвой

Покрывающее дерево системы переходов

- 1) Вершины дерева помечены состояниями системы переходов
- 2) Каждая вершина является живой, либо мертвой
- 3) Корень – живая вершина с пометкой s_0

Покрывающее дерево системы переходов

- 1) Вершины дерева помечены состояниями системы переходов
- 2) Каждая вершина является живой, либо мертвой
- 3) Корень – живая вершина с пометкой s_0
- 4) Мертвые вершины не имеют потомков

Покрывающее дерево системы переходов

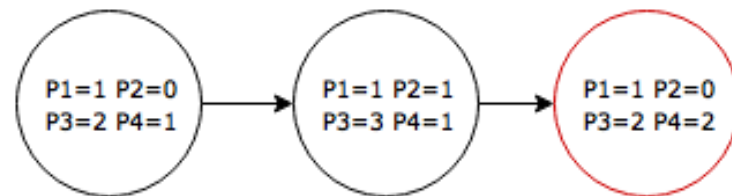
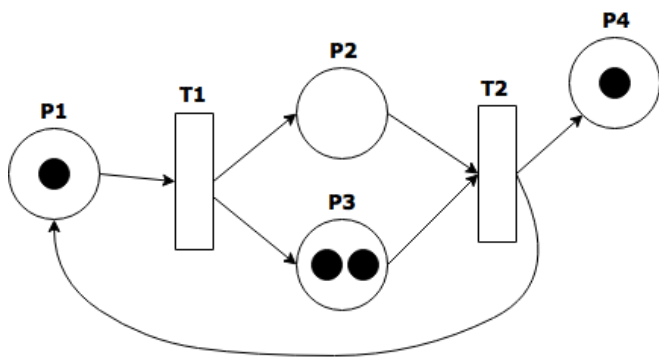
- 1) Вершины дерева помечены состояниями системы переходов
- 2) Каждая вершина является живой, либо мертвой
- 3) Корень – живая вершина с пометкой s_0
- 4) Мертвые вершины не имеют потомков
- 5) Живая вершина с пометкой s имеет по одному потомку на каждое состояние из $Succ(s)$

Покрывающее дерево системы переходов

- 1) Вершины дерева помечены состояниями системы переходов
- 2) Каждая вершина является живой, либо мертвой
- 3) Корень – живая вершина с пометкой s_0
- 4) Мертвые вершины не имеют потомков
- 5) Живая вершина с пометкой s имеет по одному потомку на каждое состояние из $Succ(s)$
- 6) Если на пути от корня до вершины с пометкой s' есть вершина с пометкой $s \leq s'$, то она мертвая

Покрывающее дерево системы переходов

- 1) Вершины дерева помечены состояниями системы переходов
- 2) Каждая вершина является живой, либо мертвой
- 3) Корень – живая вершина с пометкой s_0
- 4) Мертвые вершины не имеют потомков
- 5) Живая вершина с пометкой s имеет по одному потомку на каждое состояние из $Succ(s)$
- 6) Если на пути от корня до вершины с пометкой s' есть вершина с пометкой $s \leq s'$, то она мертвая



Метод насыщения

Задача покрытия: может ли быть достигнуто некоторое состояние s' из начального состояния s_0 , такое что оно $s' \geq s$, где s – заданное состояние, чье покрытие необходимо проверить?

Метод насыщения

Задача покрытия: может ли быть достигнуто некоторое состояние s' из начального состояния s_0 , такое что оно $s' \geq s$, где s – заданное состояние, чье покрытие необходимо проверить?

$Pred^*(I)$ – предел последовательности $I_0 \subseteq I_1 \subseteq \dots$, где

$$I_0 =^{def} I$$
$$I_{n+1} =^{def} I_n \cup Pred(I_n)$$

Задача покрытия: может ли быть достигнуто некоторое состояние s' из начального состояния s_0 , такое что оно $s' \geq s$, где s – заданное состояние, чье покрытие необходимо проверить?

$Pred^*(I)$ – предел последовательности $I_0 \subseteq I_1 \subseteq \dots$, где

$$I_0 =^{def} I$$
$$I_{n+1} =^{def} I_n \cup Pred(I_n)$$

Решение задачи покрытия: проверка $s_0 \in Pred^*(\uparrow s)$.

Задача покрытия: может ли быть достигнуто некоторое состояние s' из начального состояния s_0 , такое что оно $s' \geq s$, где s – заданное состояние, чье покрытие необходимо проверить?

$Pred^*(I)$ – предел последовательности $I_0 \subseteq I_1 \subseteq \dots$, где

$$I_0 =^{def} I$$
$$I_{n+1} =^{def} I_n \cup Pred(I_n)$$

Решение задачи покрытия: проверка $s_0 \in Pred^*(\uparrow s)$.

Последовательность множеств K_0, K_1, \dots , где

$$K_0 =^{def} I^b \text{ и } K_{n+1} =^{def} K_n \cup pb(K_n)$$

Задача покрытия: может ли быть достигнуто некоторое состояние s' из начального состояния s_0 , такое что оно $s' \geq s$, где s – заданное состояние, чье покрытие необходимо проверить?

$Pred^*(I)$ – предел последовательности $I_0 \subseteq I_1 \subseteq \dots$, где

$$I_0 =^{def} I$$
$$I_{n+1} =^{def} I_n \cup Pred(I_n)$$

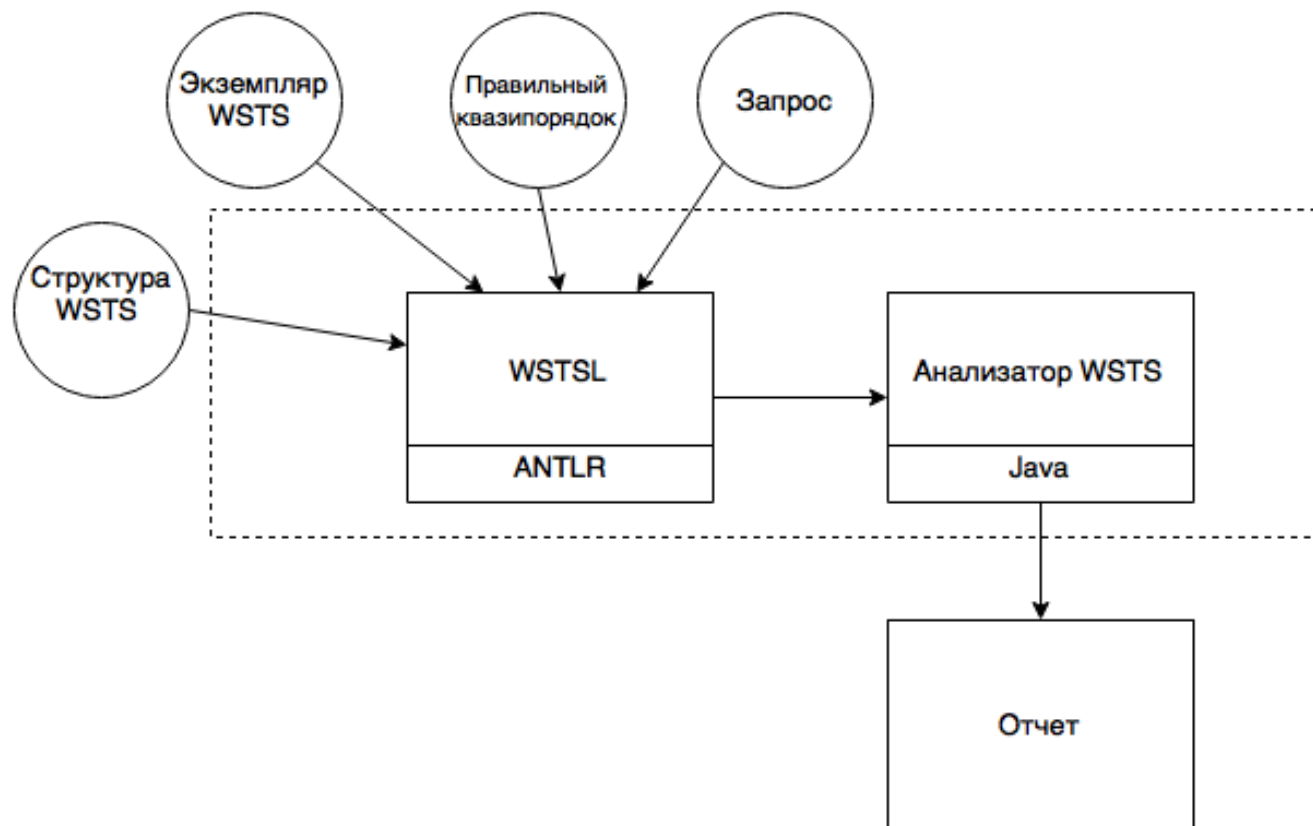
Решение задачи покрытия: проверка $s_0 \in Pred^*(\uparrow s)$.

Последовательность множеств K_0, K_1, \dots , где

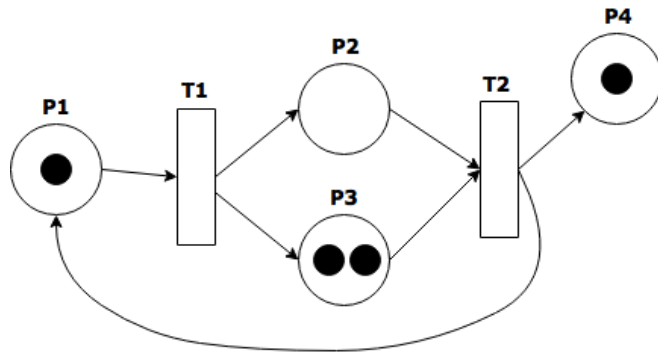
$$K_0 =^{def} I^b \text{ и } K_{n+1} =^{def} K_n \cup pb(K_n)$$

Может быть показано, что $\uparrow \cup K_i = Pred^*(I)$
 \Rightarrow **необходимо проверить** $s_0 \in \uparrow \min(\uparrow \cup K_i)$,

Архитектура системы



- 1) Основывается на языке SETL – ориентирован на работу с множествами
- 2) Похож на Python
- 3) Операторы
exists iterator | test
forall iterator | test
cpr
- 4) Встроенные операторы порядков (**embedsinto**, **parikh** и др.)



```

K0: [{P1=1,P2=1,P3=1,P4=2}]
K1: [{P1=0,P2=2,P3=2,P4=1},
     {P1=2,P2=0,P3=0,P4=2}]
K2: [{P1=1,P2=1,P3=1,P4=1}]
K3: [{P1=0,P2=2,P3=2,P4=0},
     {P1=2,P2=0,P3=0,P4=1}]
K4: [{P1=1,P2=1,P3=1,P4=0}]
K5: [{P1=2,P2=0,P3=0,P4=0}]
Union: [{P1=0,P2=2,P3=2,P4=0},
        {P1=0,P2=2,P3=2,P4=1},
        {P1=1,P2=1,P3=1,P4=0},
        {P1=1,P2=1,P3=1,P4=1},
        {P1=1,P2=1,P3=1,P4=2},
        {P1=2,P2=0,P3=0,P4=0},
        {P1=2,P2=0,P3=0,P4=1},
        {P1=2,P2=0,P3=0,P4=2}]
min(Union): [{P1=0,P2=2,P3=2,P4=0},
             {P1=1,P2=1,P3=1,P4=0},
             {P1=2,P2=0,P3=0,P4=0}]

```

The state $\{P1=1,P2=1,P3=1,P4=2\}$ is not covered

```

P1 = { "P1", "P2", "P3", "P4" };
T1 = { "T1", "T2" };
PT1 = { ["T1", "P1"], ["T2", "P2"], ["T2", "P3"] };
TP1 = { ["T1", "P2"], ["T1", "P3"], ["T2", "P1"], ["T2", "P4"] };

```

```
PN1 = [ P1, T1, PT1, TP1 ];
```

```

m0 = { <"P1", 1>, <"P2", 0>, <"P3", 2>, <"P4", 1> };
mc = { <"P1", 1>, <"P2", 1>, <"P3", 1>, <"P4", 2> };

```

```

func wgo(PN, s1, s2)
    return forall p in PN[0] | s1[p] <= s2[p];
end func;

```

```

func pred(wsts, s)
    P = wsts[0];
    T = wsts[1];
    PT = wsts[2];
    TP = wsts[3];
    predecessors = { };

    for t in T
        if forall tp in TP[t] | s[tp[1]] - 1 >= 0 then
            s1 = s;
            for pt in PT[t]
                s1[pt[1]] = s1[pt[1]] + 1;
            end for;
            for tp in TP[t]
                s1[tp[1]] = s1[tp[1]] - 1;
            end for;
            predecessors = predecessors with s1;
        end if;
    end for;


    return predecessors;
end func;

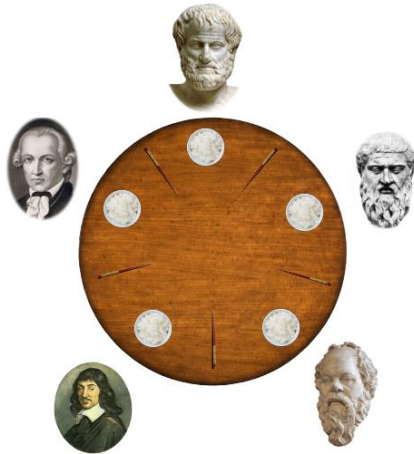
```

```
backwardanalysis(PN1, wgo, pred, m0, mc);
```

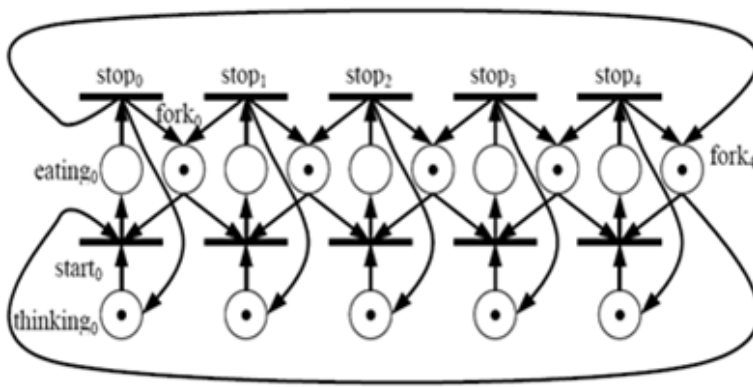

Покрытие тестами

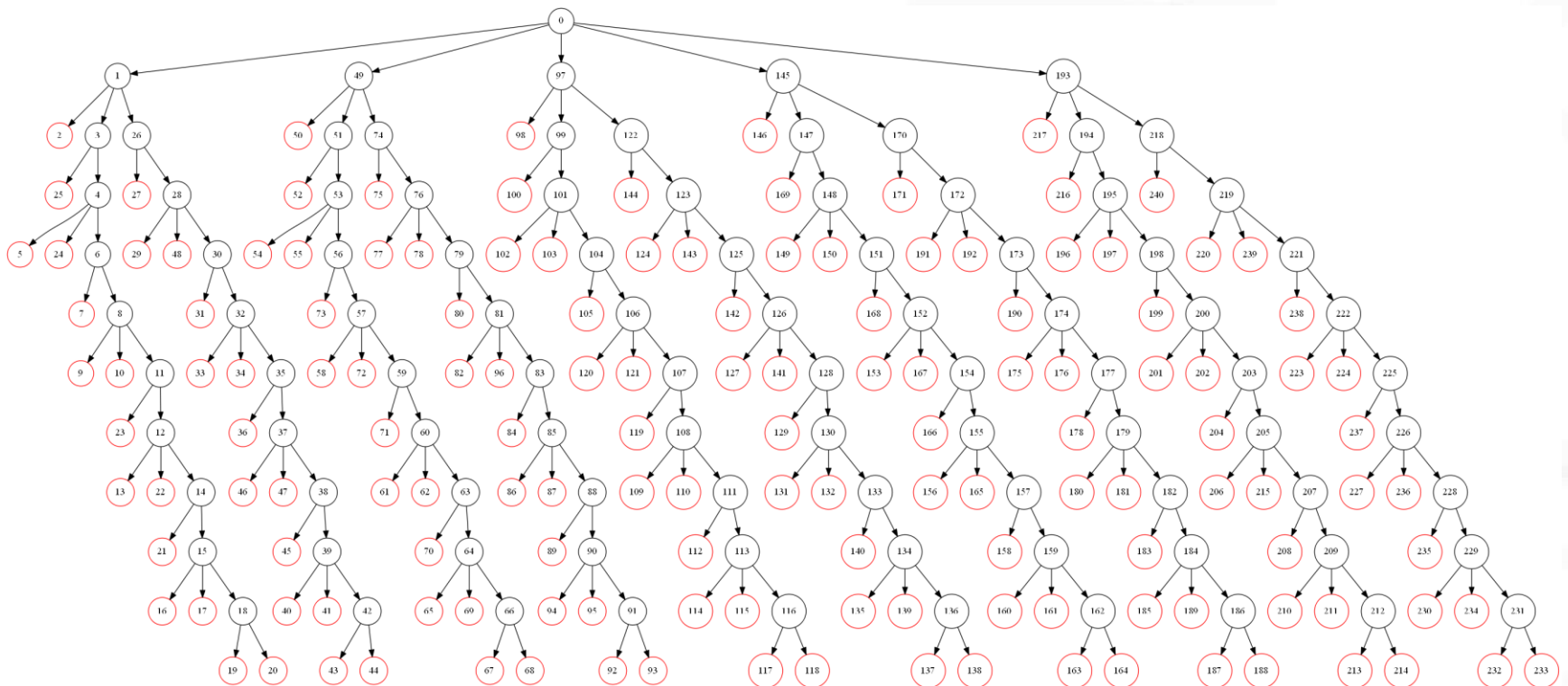
Done: 191 of 191 Failed: 24 (in 1s)

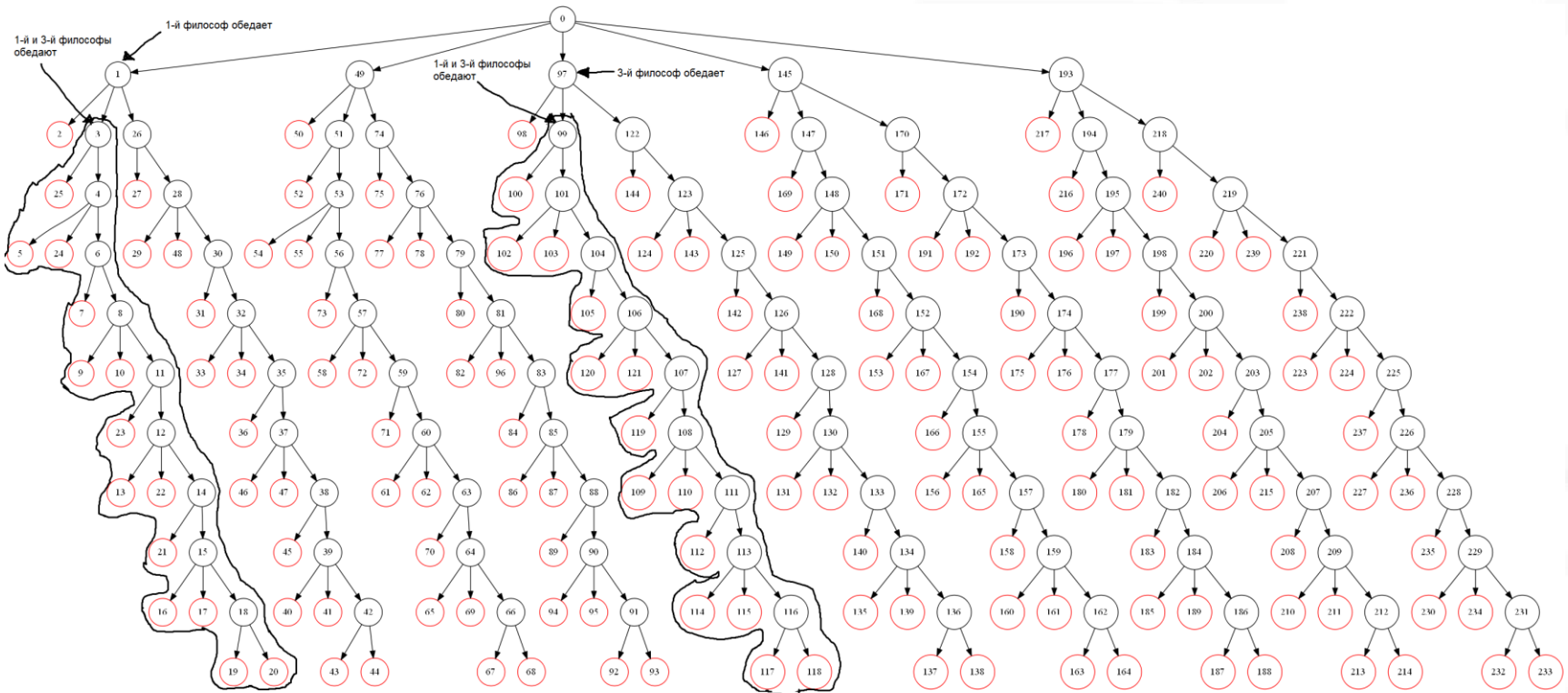
Element	...	Method, %	Line, %
 wstsl.Interpreter	...	73% (47/64)	66% (286/432)



	Run time (s)	Size of FRT
Phil5	0.08587	241
Phil6	1.87815	25711
Phil7	5221.64756	88062003

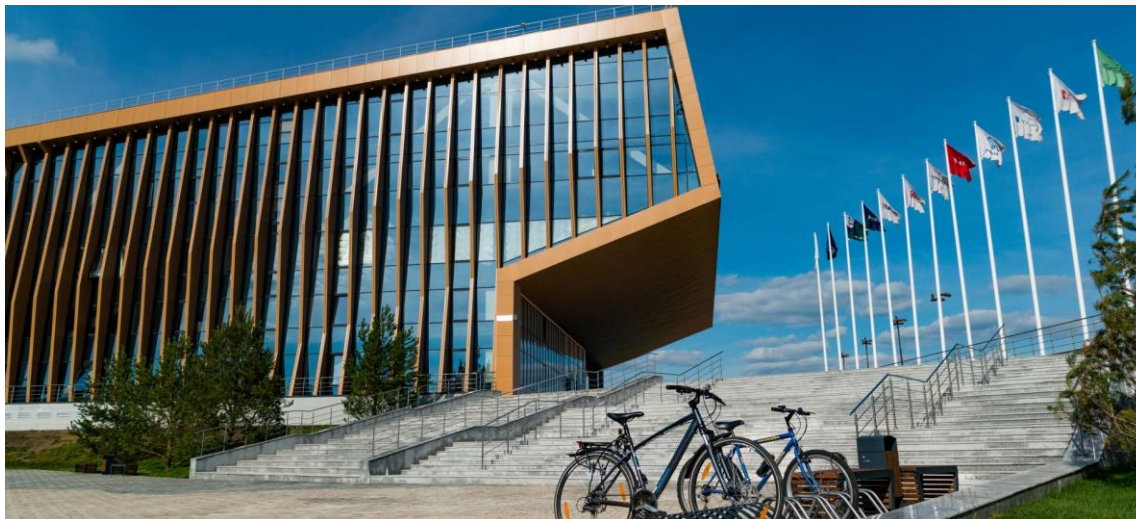
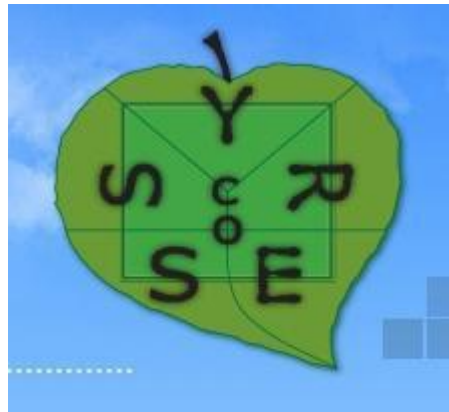








АПРОБАЦИЯ РАБОТЫ



ОСНОВНЫЕ РЕЗУЛЬТАТЫ практики

- 1) Алгоритмы поведенческого анализа (90%)
- 2) Интерпретатор WSTSL (80%)
- 3) Эксперимент (75%)
- 4) Текст ВКР (65%)
- 5) Документация по ЕСПД (20%)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. A. Finkel and P. Schnoebelen, "Well-structured transition systems everywhere!", Theoretical Computer Science, vol. 256, no. 1-2, pp. 63-92, 2001.
2. P. Abdulla, K. Čerāns, B. Jonsson and Y. Tsay, "Algorithmic Analysis of Programs with Well Quasi-ordered Domains", Information and Computation, vol. 160, no. 1-2, pp. 109-127, 2000.
3. Е. Кузьмин, В. Соколов, Вполне структурированные системы помеченных переходов, М.: ФИЗМАТЛИТ, 2005.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Спасибо за внимание!

Михайлов Владимир Евгеньевич,
vemikhaylov@edu.hse.ru

Москва - 2017