

Структурный анализ диаграмм и генерация кода в среде программирования роботов TRIK Studio

К.А.Батоев

Руководитель: ст.преп. Я.А.Кириленко

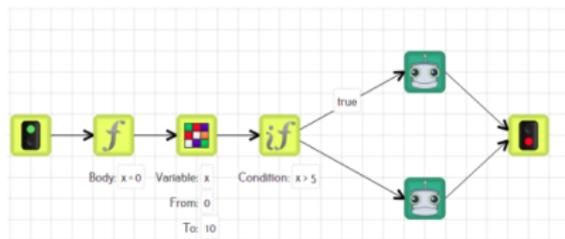
Консультант: асп. Д.А.Мордвинов

Кафедра системного программирования
344 группа

Санкт-Петербург, 2018

Способы создания диаграмм:

- ВИЗУАЛЬНЫЙ
- ТЕКСТОВЫЙ



```

1 var __interpretation_started_timestamp__;
2 var pi = 3.1415926535897931;
3
4 var x;
5
6 var main = function()
7 {
8     __interpretation_started_timestamp__ = Date.now();
9
10    x = 0;
11    x = script.random(0, 10);
12
13    if (x > 5) {
14        brick.smile();
15    } else {
16        brick.sadSmile();
17    }
18    return;
19 }

```

Не всегда удастся получить код по диаграмме:

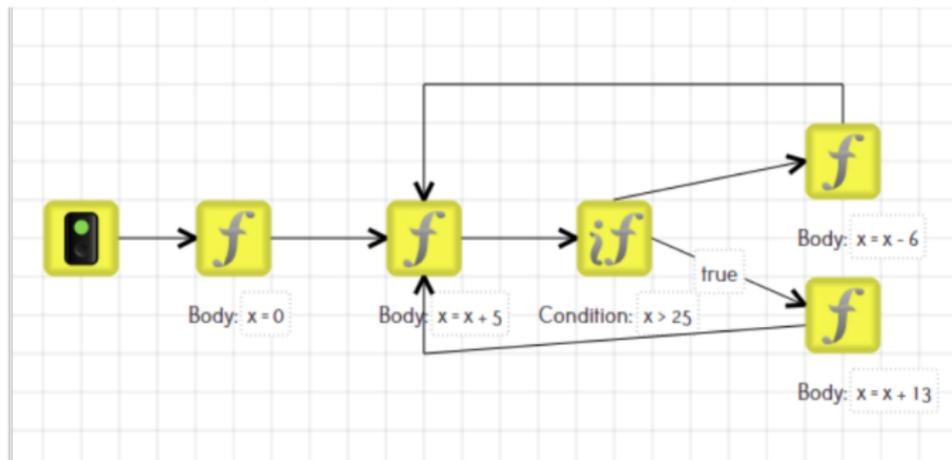


Рис.: Пример неструктурируемой диаграммы

- *Структуризация* — процесс получения абстрактного синтаксического дерева, в узлах которого находятся метки, обозначающие операции языков программирования высокого уровня.
- Полученное дерево будем называть *деревом управляющих структур*.

Целью данной курсовой работы является улучшение компоненты генерации кода в среде TRIK Studio. Для этого были поставлены следующие задачи:

- реализовать алгоритм структуризации графа потока управления
- провести апробацию алгоритма на диаграммах

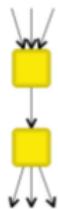
Анализ графа потока управления

- итеративный
- интервальный
- структурный

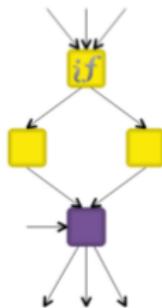
Для поиска циклических структур нужно вычислить следующие структуры:

- отношение доминирования в ориентированном графе
- обратные ребра

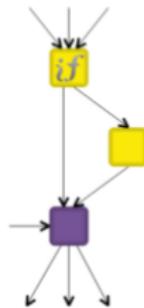
Шаблоны структурного анализа



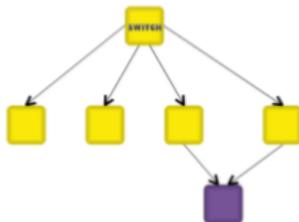
(a) Блок



(b) Если-То-Иначе



(c) Если-То

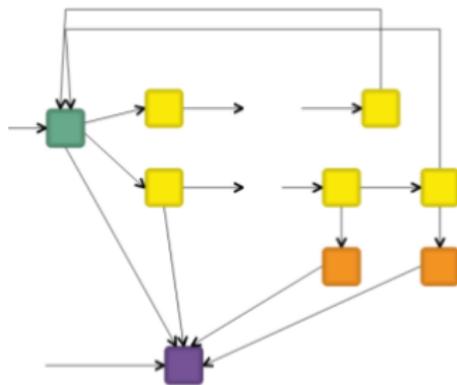


(d) Выбор



(e) Цикл Пока

Остальные шаблоны



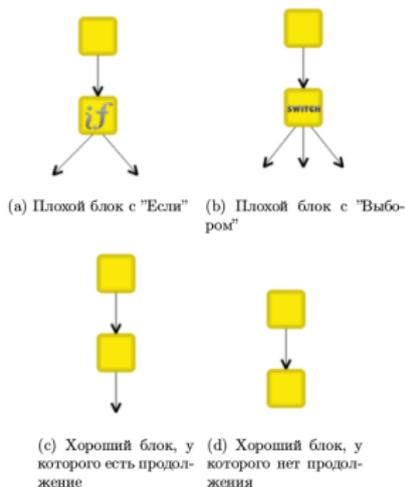
(f) Цикл пока с инструкциями прерывания



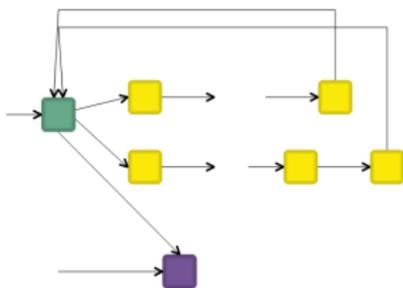
(g) Бесконечный цикл

- инициализировать структуры данных
- пока есть изменения в графе, пытаться сопоставлять вершинам шаблоны
- при успешном сопоставлении изменять граф

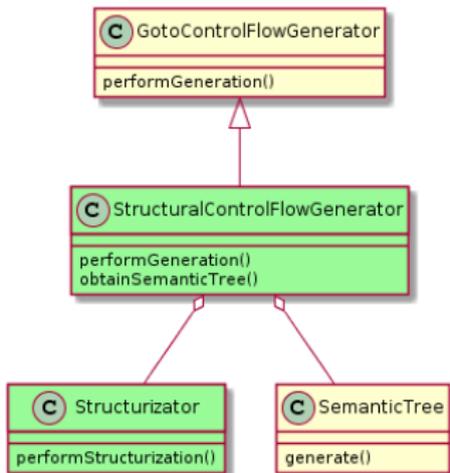
Для облегчения процесса последующей генерации был изменен шаблон «Блок»



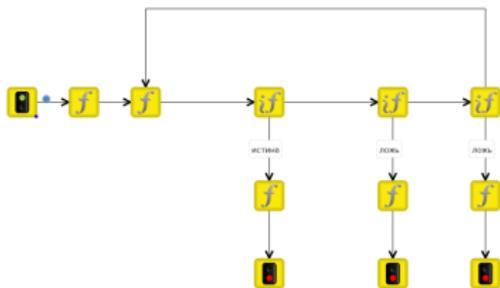
Редукция шаблона «Цикл Пока с прерываниями»



- конструирование дерева без знания информации о типе вершин и ребер
- делегирование процесса генерации самому дереву

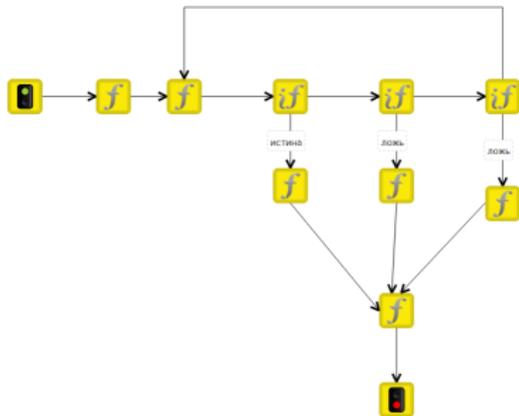


Алгоритм был протестирован на различных диаграммах, учитывающих все шаблоны структурного анализа. Опишем наиболее интересные случаи



```
1 var __interpretation_started_timestamp__;  
2 var pi = 3.1415926535897931;  
3  
4 var x;  
5  
6 var main = function()  
7 {  
8     __interpretation_started_timestamp__ = Date.now();  
9  
10    x = 100;  
11    while (true) {  
12        x = x + 5;  
13        if (x > 0) {  
14            x = 5;  
15            return;  
16            break;  
17        }  
18        if (!(x > 100)) {  
19            x = 100;  
20            return;  
21            break;  
22        }  
23        if (!(x > 0)) {  
24            x = x + 100;  
25            return;  
26            break;  
27        }  
28    }  
29 }  
30
```

Точка продолжения исполнения находится на третьем уровне



```
1 var __interpretation_started_timestamp__;  
2 var pi = 3.1415926535897931;  
3  
4 var x;  
5  
6 var main = function()  
7 {  
8     __interpretation_started_timestamp__ = Date.now();  
9  
10    x = 0;  
11    while (true) {  
12        x = x + 30;  
13        if (x > 0) {  
14            x = 10;  
15            break;  
16        }  
17        if (!(x < 50)) {  
18            x = 15;  
19            break;  
20        }  
21        if (!(x < 25)) {  
22            x = 20;  
23            break;  
24        }  
25    }  
26    x = x - 100;  
27    return;  
28 }  
29
```


- был реализован алгоритм структурного анализа в качестве алгоритма структуризации графа потока управления
- алгоритм был апробирован на диаграммах среды TRIK Studio

- интегрировать алгоритм структурного анализа в главную ветку среды TRIK Studio
- линеаризация графа потока управления