ИТМО
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Генерация управляющих автоматов на основе генетического программирования и верификации

Егоров Кирилл Викторович

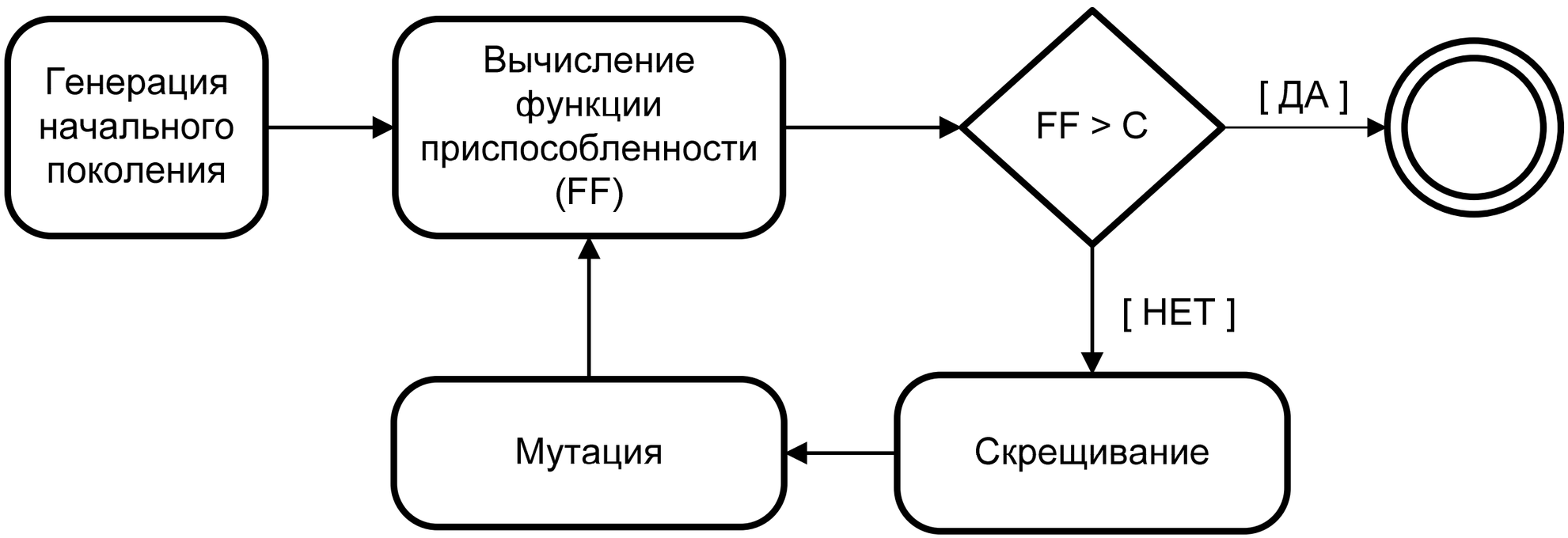
Диссертация на соискание ученой степени кандидата технических наук

Специальность 05.13.11.
Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Научный руководитель – д.т.н., проф. А.А. Шалыто

Актуальность темы (1)

- Автоматное программирование – парадигма программирования, в рамках которой поведение программ предлагается описывать **управляющими** конечными автоматами (автоматы).
- Построенные вручную автоматы могут быть не оптимальны, или их построение очень трудоемко. Поэтому автоматы предлагается генерировать.
- Один из вариантов генерации – поисковая оптимизация, в частности, генетическое программирование.



Актуальность темы (2)

- **Методы генерации автоматов на основе генетического программирования:**
  - С использованием моделирования (Поликарпова Н.И., Точилин В.Н.).
  - По тестам (Царев Ф.Н.).
- При моделировании для каждой задачи функция приспособленности строится заново.
- При построение по тестам функция приспособленности универсальна, но поведение автомата не гарантирует соответствие спецификации заданной формально, например, с помощью *LTL*-формул.
- Верификация позволяет доказать соответствие автомата и спецификации, поэтому предлагается использовать ее в ходе генетического программирования.

Совместное применение генетического программирования и верификации (1)

- В этом направлении известны только следующие работы:
- *Johnson C*. Genetic Programming with Fitness based on Model Checking. // Lecture Notes in Computer Science. Springer Berlin / Heidelberg. 2007. Volume 4445/2007, pp. 114 – 124.
- *Katz G., Peled D.* Model Checking-Based Genetic Programming with an Application to Mutual Exclusion / Proceedings of the 14th International Conference, Tools and Algorithms for the Construction and Analysis of Systems (TACAS-2008), Budapest. Springer Berlin Heidelberg. 2008, pp. 141 – 156.
- *Katz G., Peled D.* Genetic Programming and Model Checking: Synthesizing New Mutual Exclusion Algorithms / Proceedings of the 6th International Symposium, Automated Technology for Verification and Analysis (ATVA-2008), Seoul. Springer Berlin Heidelberg. 2008, pp. 33 – 47.
- *Katz G., Peled D.* Synthesizing Solutions to the Leader Election Problem using Model Checking and Genetic Programming / Proceedings of the 5th International Haifa Verification Conference (HVC-2009), Haifa. Springer Berlin Heidelberg. 2009, pp. 117 – 132.
- *Katz G., Peled D.* Code Mutation in Verification and Automatic Code Correction / Proceedings of the 16th International Conference, Tools and Algorithms for the Construction and Analysis of Systems (TACAS-2010), Paphos. Springer Berlin Heidelberg. 2010, pp. 435 – 450.
- *Katz G., Peled D.* MCGP: A Software Synthesis Tool Based on Model Checking and Genetic Programming / Proceedings of the 8th International Symposium, Automated Technology for Verification and Analysis (ATVA 2010), Singapore. Springer Berlin Heidelberg. 2010, pp. 359 – 364.

Совместное применение генетического программирования и верификации (2)

- Эти работы посвящены генерации программ, а не управляющих автоматов.
- В них вклад темпоральных формул в функцию приспособленности рассматривается как ноль или единица.
- Эти работы не используют верификацию в ходе выполнения операций скрещивания и мутации.
- Данные работы не используют ни тесты, ни контракты.
- Указанные недостатки ограничивают быстродействие указанных подходов.

Цель и задачи диссертационной работы

Цель работы – генерации автоматов на основе генетического программирования и верификации.

Для достижения этой цели в диссертации должны быть решены следующие задачи, в которых вместо тестов будут использоваться сценарии:

1. Предложить функцию приспособленности, учитывающую верификацию.
2. Разработать операции скрещивания и мутации, учитывающие верификацию.
3. Разработать алгоритм построения автоматов на основе генетического программирования с учетом контрактов.
4. Разработать верификатор, приспособленный для поддержки генерации автоматов на основе генетического программирования.
5. Разработать технологию и инструментальное средство для генерации автоматов с помощью генетического программирования и верификации.

Научная новизна

1. Предложена функция приспособленности, учитывающая выполнение темпоральной формулы на части автомата. В известных работах вклад каждой формулы был ноль или единица.
2. Операции скрещивания и мутации отличаются от известных тем, что в ходе их выполнения используется верификация.
3. Алгоритм генерации автоматов с учетом контрактов, которые предложено рассматривать как разновидность темпоральных формул.
4. Алгоритм генерации автоматов по темпоральным формулам и сценариям работы, который позволяет ускорить построение автоматов по сравнению с применением тестов.

Основные понятия. Темпоральные формулы

- Спецификация – набор темпоральных (временных) формул.
- В работе используются темпоральные формулы, записанные на языке логики линейного времени (*Linear Temporal Logic, LTL*).
- Пример:
  - *G(wasEvent(e1) -> X(wasAction(z1)))* – «если был совершен переход по событию *e1*, то следующим действием будет *z1*»;
  - *wasEvent(e2) U wasAction(z2)* – «переход по событию *e2* будет совершаться до тех пор, пока не будет вызвано действие *z2*».
- Контракты – *LTL*-формулы определенного вида:
  - предусловие $G(Xe \rightarrow p)$;
  - постусловие ($G(e \rightarrow Xp)$;
  - инвариант $G(e \rightarrow p)$.

Основные понятия. Тесты и сценарии

Тесты:

e1	e2	e3	e4	e5
z1, z2, z3, z4, z5, z6, z7				

Позитивный сценарий:

e1	e2	e3	e4	e5
z1	z2, z3	z4	z5, z6	z7

Негативный сценарий – последовательность переходов, которая не должна выполняться**:**

e1, e2, e3, e4, **e4**

Исходные данные для алгоритма генетического программирования

- Список событий $\{e_1, e_2, \ldots, e_v\}$.
- Список входных переменных $\{x_1, x_2, \ldots, x_w\}$.
- Список выходных воздействий $\{z_1, z_2, \ldots, z_t\}$.
- Максимальное число состояний k.
- Набор тестов или сценариев.
- Набор *LTL*-формул и контрактов.

I. Функция приспособленности (1)

$$\frac{\sum_{i=1}^{n}\left(1-\frac{ED\left(Output[i],Answer[i]\right)}{\max\left(\left|Output[i]\right|,\left|Answer[i]\right|\right)}\right)}{n}+\frac{C-T}{\max(n,m)\cdot C}-\frac{k_1}{k_2}+\frac{\sum_{i=1}^{m}\frac{t_i}{T}}{m}$$

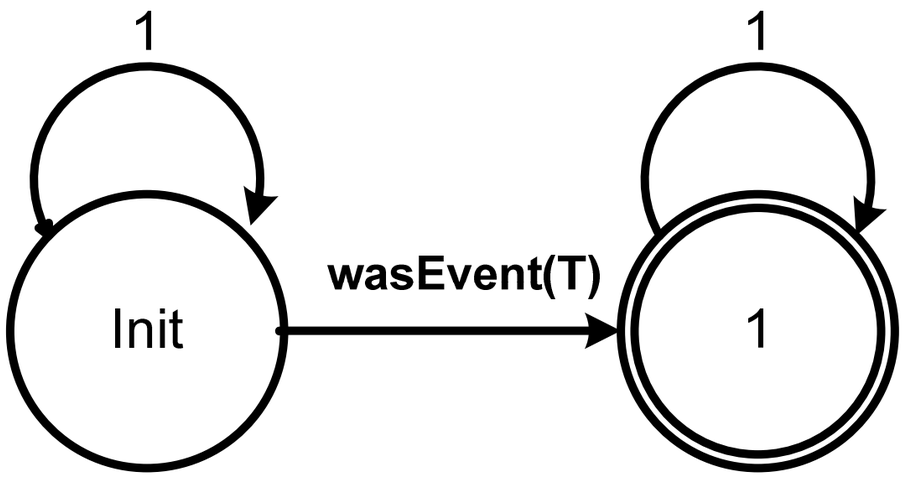
- Первое слагаемое учитывает вклад позитивных сценариев.
- Второе – вклад числа переходов.
- Третье – вклад негативных сценариев.
- Четвертое – вклад *LTL*-формул и контрактов.

I. Функция приспособленности (2)

$$\frac{\sum_{i=1}^{n}\left(1-\frac{ED\left(Output[i],Answer[i]\right)}{\max\left(|Output[i]|,|Answer[i]|\right)}\right)}{n}+\frac{C-T}{\max(n,m)\cdot C}-\frac{k_1}{k_2}+\frac{\sum_{i=1}^{m}\frac{t_i}{T}}{m}$$

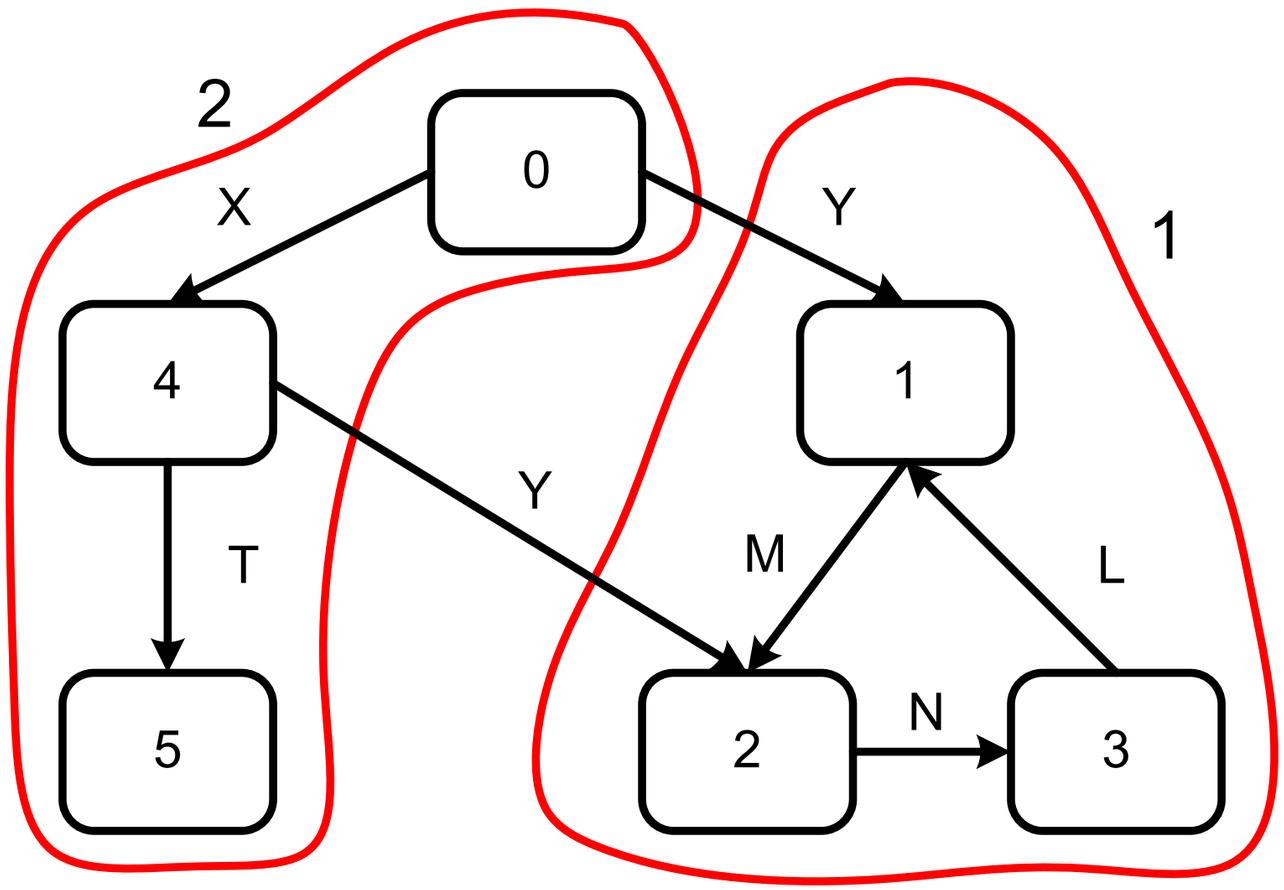
- *ED(Output[i], Answer[i])* – редакционное расстояние между реальными и ожидаемыми выходными воздействиями *i*-го сценария;
- n – число позитивных сценариев;
- k_1 – число выполненных негативных сценариев;
- k_2 – число негативных сценариев;
- m – число *LTL*-формул;
- T – число переходов в автомате;
- t_i – число переходов, для которых *i*-ая *LTL*-формула выполняется – «проверенные» переходы;
- C – число заведомо большее числа переходов.

I. Функция приспособленности (3)
Пример учета верификации в функции приспособленности



Автомат Бюхи, построенный по отрицанию формулы **G(!wasEvent(T)).**

Четвертая часть функции приспособленности



$$FF_{LTL} = \frac{3}{7}$$

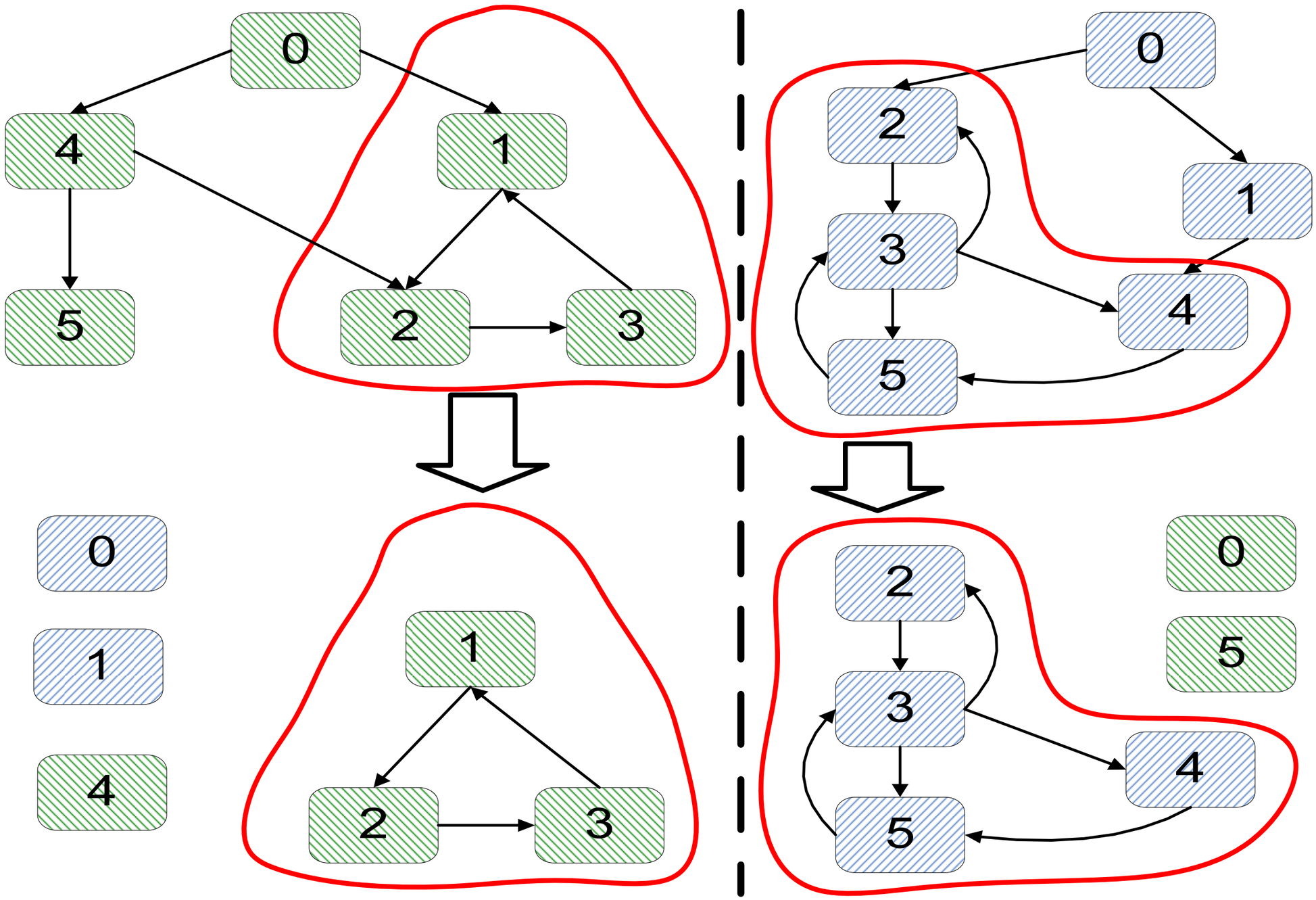
- 3 – число «проверенных» переходов из первого подмножества.
- 7 – число переходов в конечном автомате.

II. Учет верификации при мутации и скрещивании (1)

- С большей вероятностью **мутации** подвергаются переходы из контрпримера – пути автомата, на котором формула не выполняется. Операции мутации:
  - Удалить переход.
  - Изменить входное воздействие, число выходных или состояние в которое перейдет автомат.
- При **скрещивании** «проверенные» переходы переходят в новую особь без изменений. Операции скрещивания:
  - Для части автомата, образованного «проверенными» перехода, *LTL*-формула выполнятся.
  - В новой особи сохраняется пересечение множеств «проверенных» переходов для всех *LTL*-формул.

II. Учет верификации при мутации и скрещивании (2)

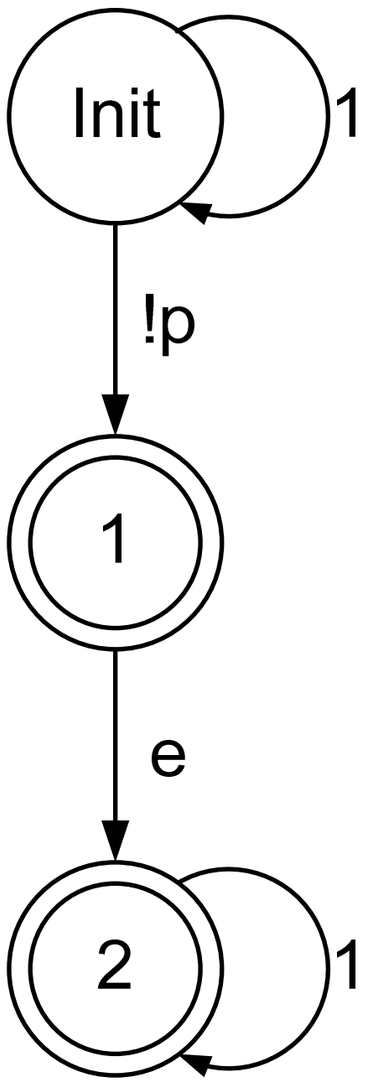
Пример скрещивания с учетом верификации



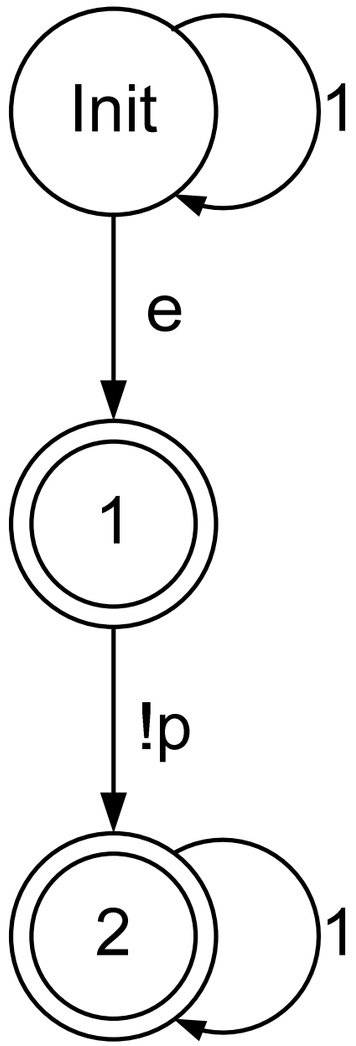
III. Автоматные контракты

Контракты – *LTL*-формулы определенного вида.

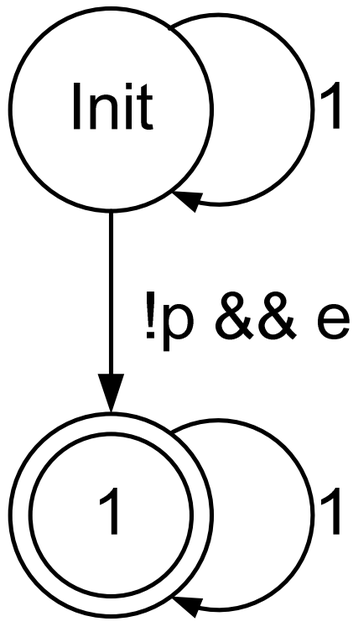
Предусловие
$G(Xe \rightarrow p)$



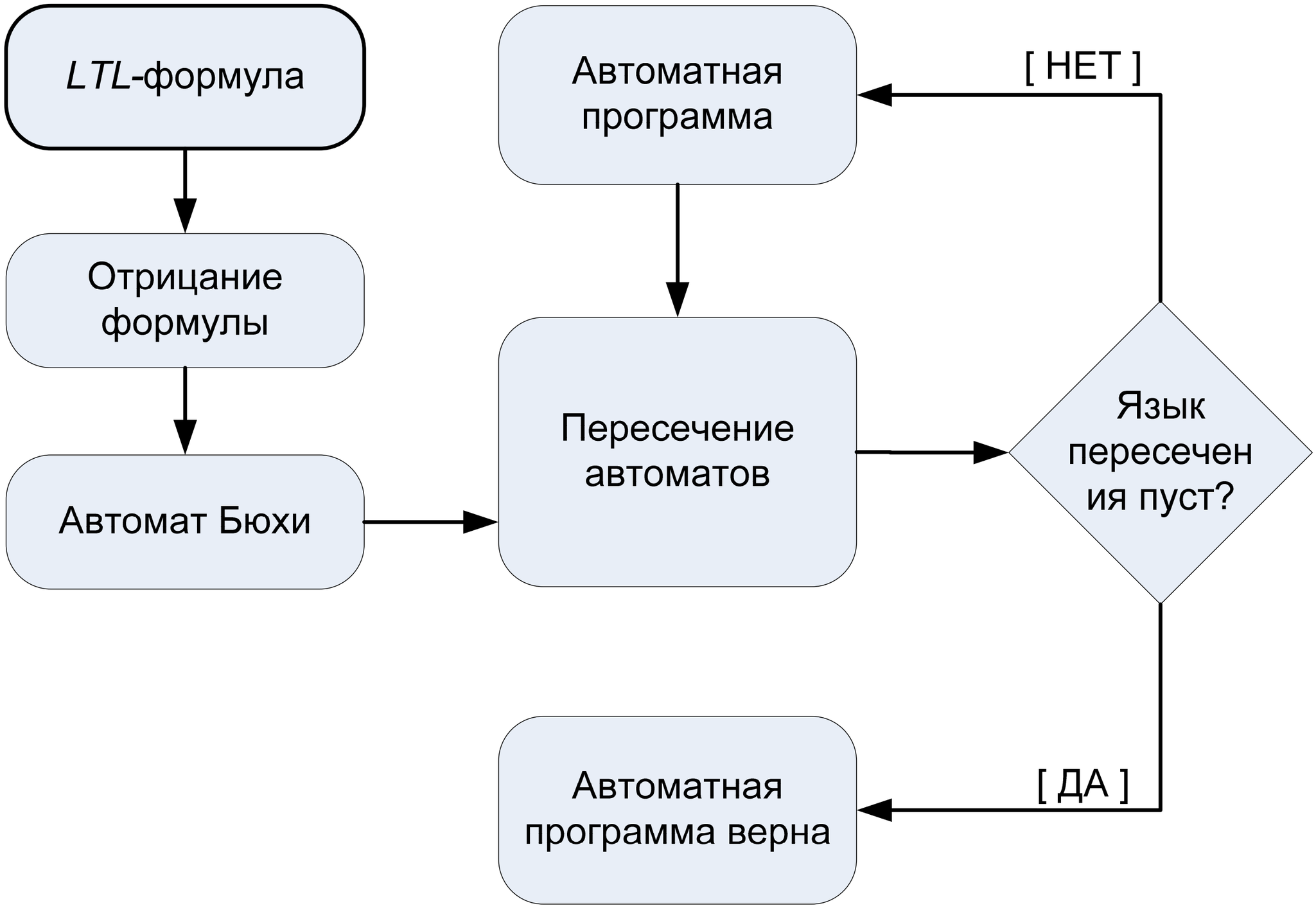
Постусловие
$G(e \rightarrow Xp)$



Инвариант
$G(e \rightarrow p)$

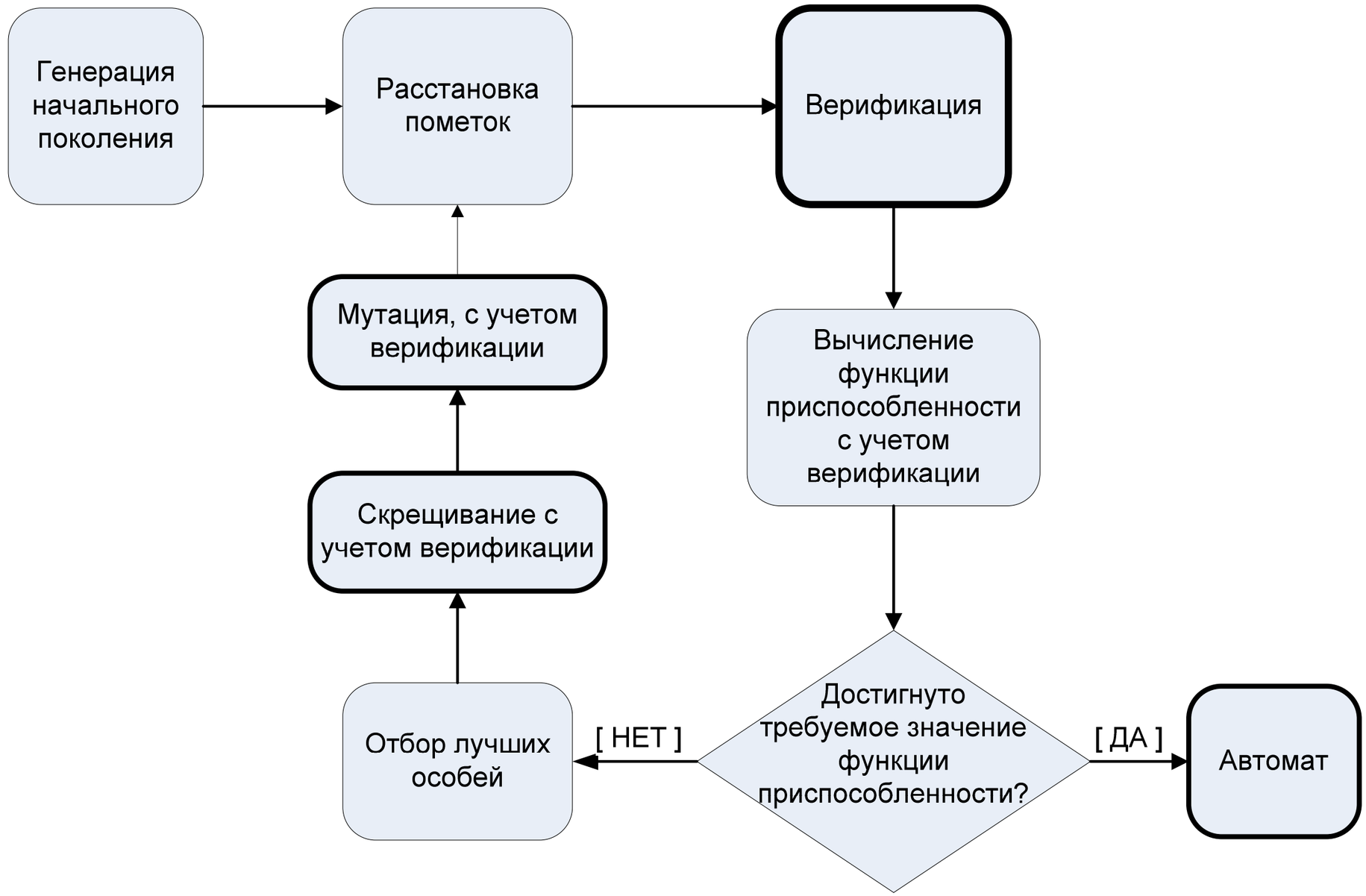


IV. Верификатор автоматных программ

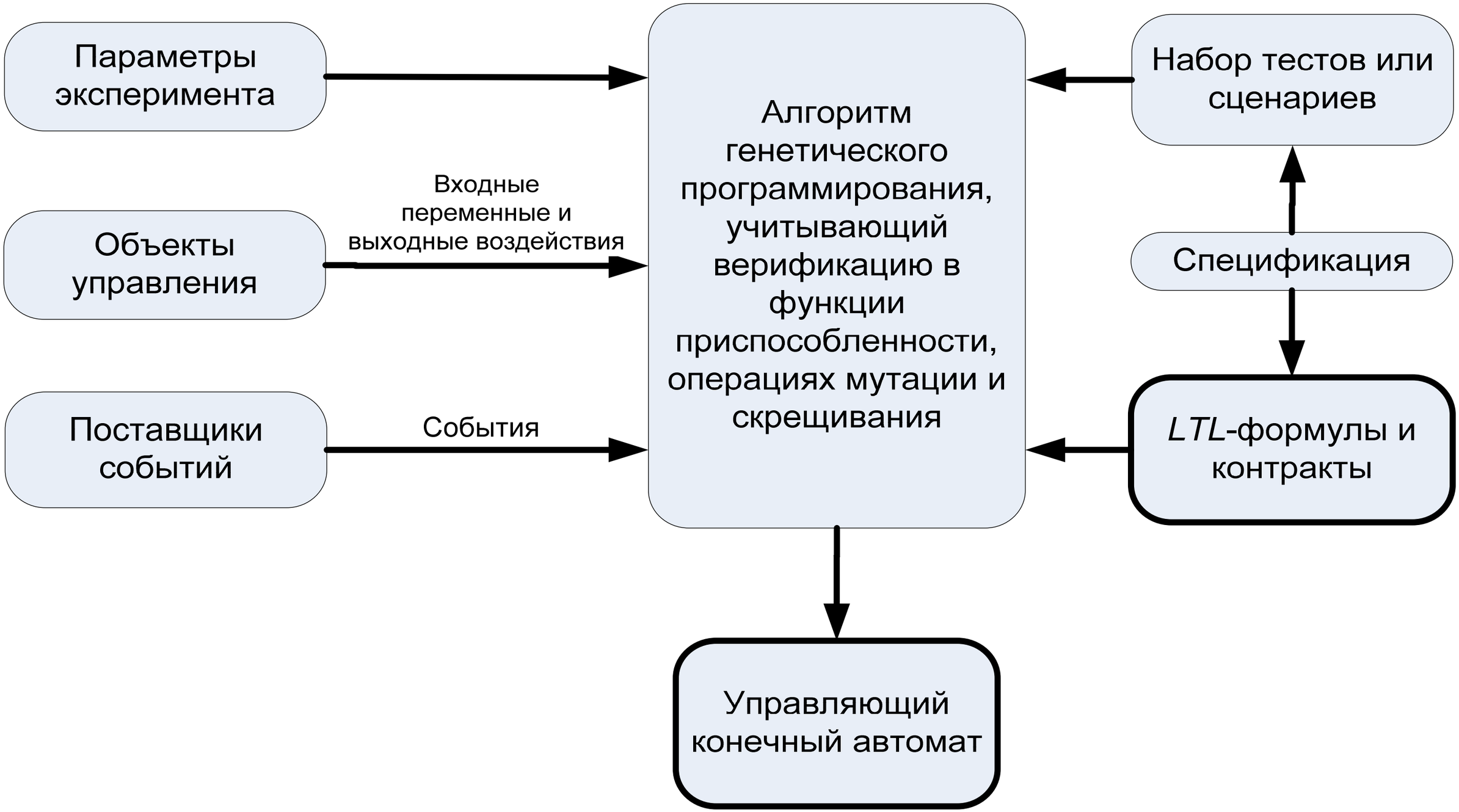


V. Технология генерации автоматов на основе генетического программирования и верификации (1)

Алгоритм генетического программирования, учитывающий верификацию в функции приспособленности, в операциях мутации и скрещивания.



V. Технология генерации автоматов на основе генетического программирования и верификации (2)

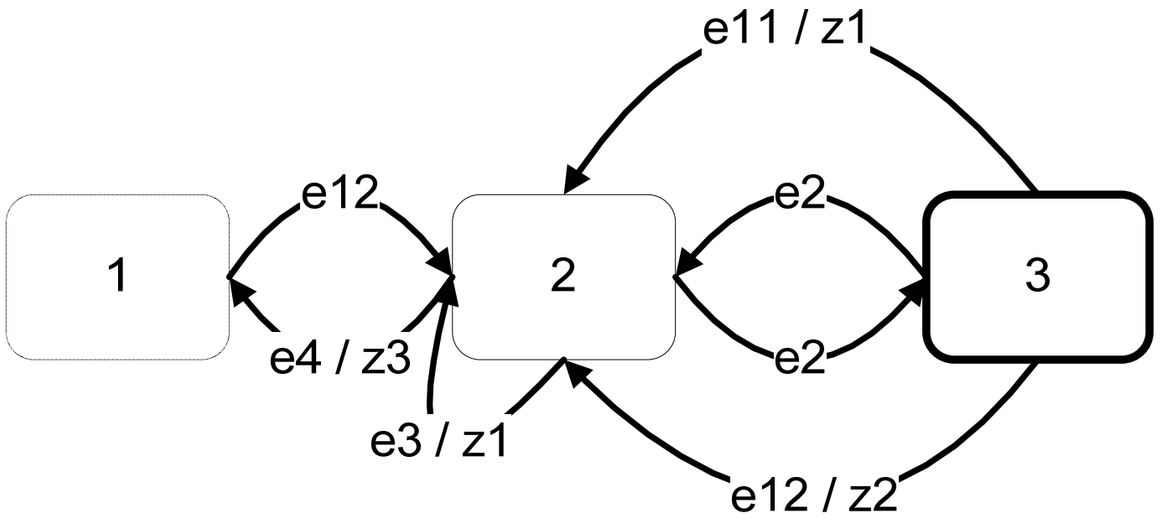


V. Инструментальное средство *GABP* для поддержки технологии

- Исходный код размещен в открытом доступе в сети Интернет по адресу: http://code.google.com/p/gabp/
- Реализует разработанные методы и технологию построения конечных автоматов
- Входные данные – описание тестов, сценариев, *LTL*-формул и контрактов в *XML*-формате
- Выходные данные – описание конечного автомата в *XML*-формате инструментального средства *UniMod*

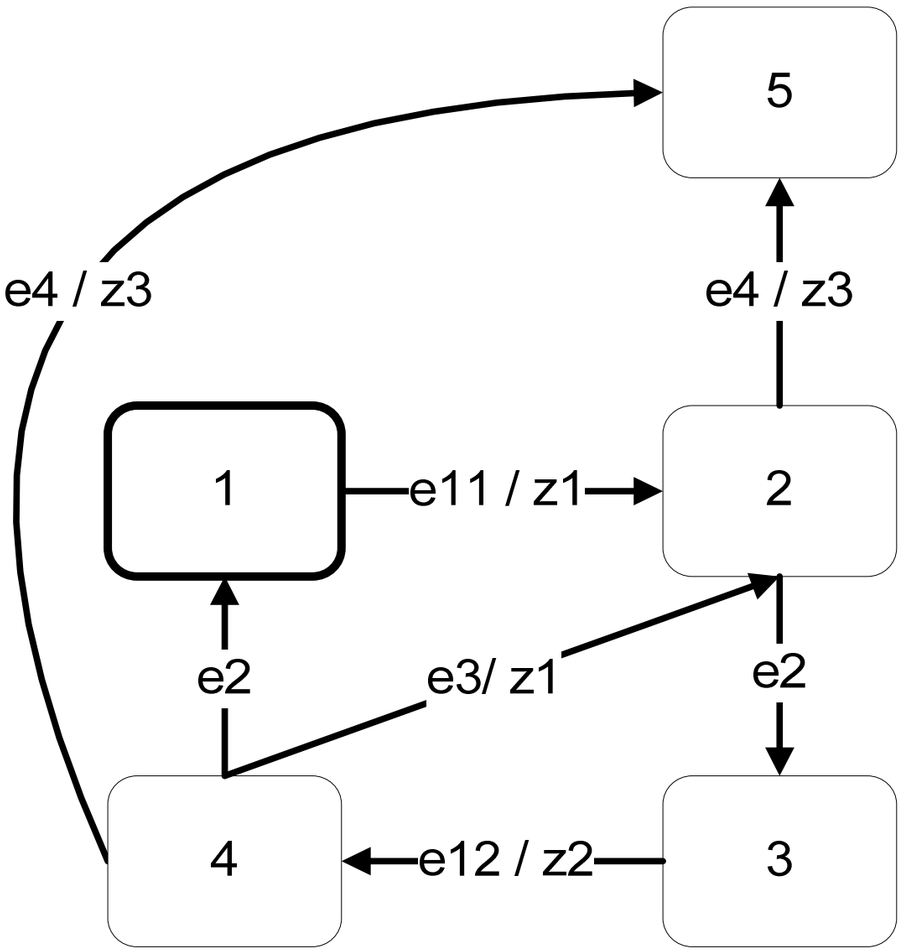
Экспериментальные исследования – автомат управления дверьми лифта (1)

Девять тестов



Сгенерированный автомат некорректный: после поломки дверей может быть отдана команда на их закрытие!

Девять тестов,11 *LTL*-формул, из которых девять – контракты



Сгенерированный автомат корректен.

Экспериментальные исследования – автомат управления дверьми лифта (2)

- Измерялось число вычислений функции приспособленности.
- 1000 экспериментов.
- **Лучший результат** показали сценарии совместно с *LTL*-формулами и контрактами.

	Тесты и *LTL*-формулы	Тесты, *LTL*-формулы и контракты	**Сценарии, *LTL*-формулы и контракты**
Среднее значение	8.372×10^5	7.109×10^5	1.616×10^5
Среднеквадратичное отклонение	7.57×10^5	6.32×10^5	1.102×10^5
Минимальное значение	6.331×10^4	6.153×10^4	3.808×10^4
Максимальное значение	5.912×10^6	4.589×10^6	8.162×10^5

Внедрение – модуль *Top Traffic Monitor* (1)

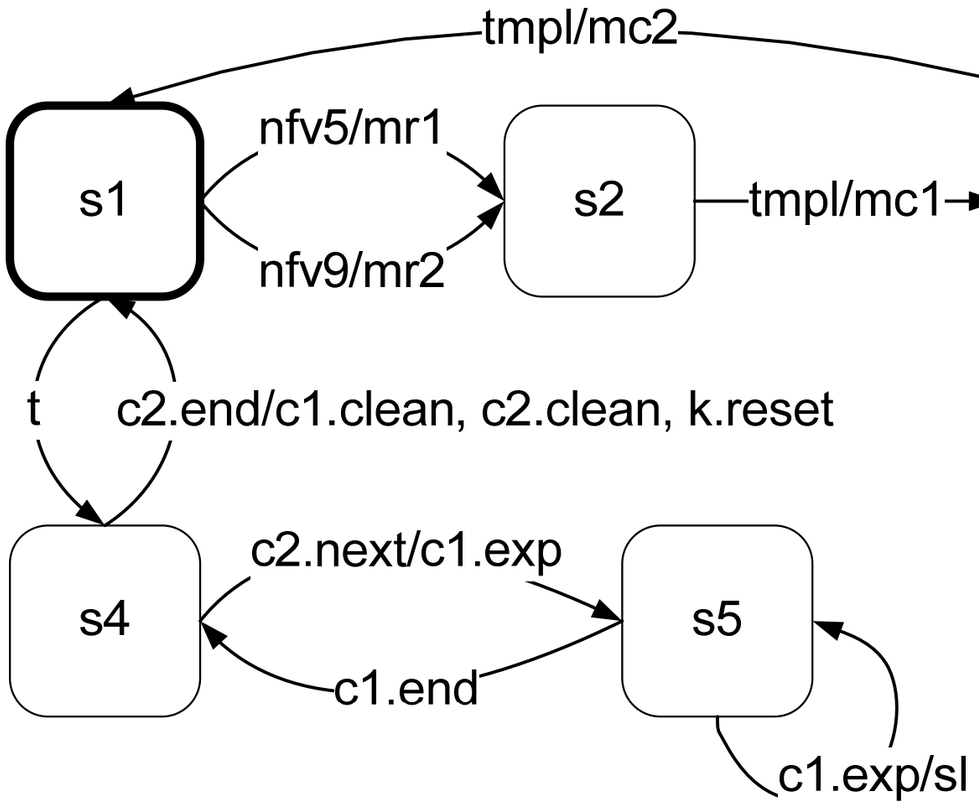
- Программный продукт для мониторинга сети и поиска потенциальных сетевых атак или угроз, разрабатываемый *ООО ЭВЕЛОПЕРС* (С.-Петербург).
- *Top Traffic Monitor* – модуль для поиска узлов сети с максимальным трафиком.
- Граф потока управления и данных (*Control and Data Flow Graph, CDFG*) – вершины выполняют операции, а ребра бывают двух типов:
  - *Control Flow* – последовательность выполнения операций.
  - *Data Flow* – связь между источниками и приемниками данных.

Внедрение – модуль *Top Traffic Monitor* (2)

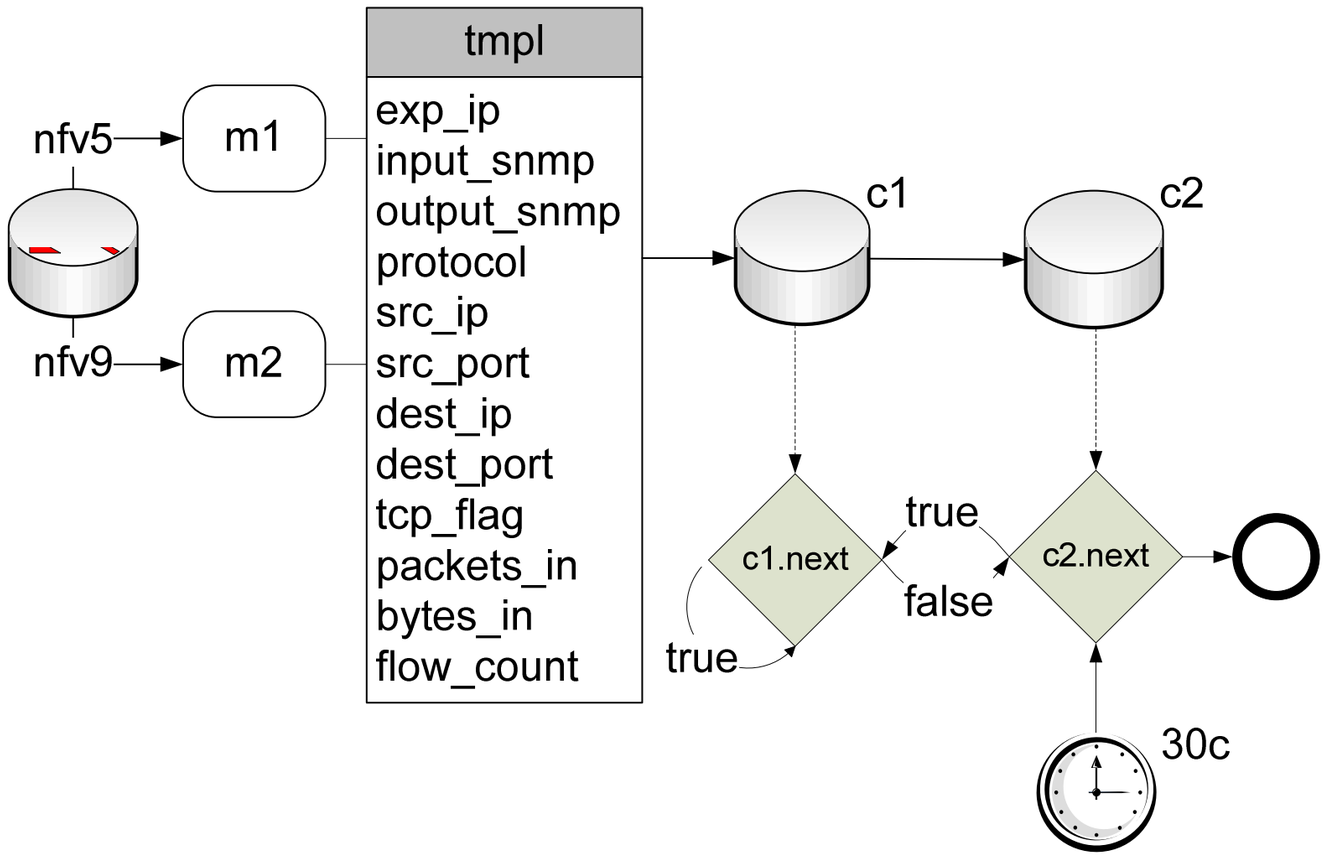
- 10 позитивных сценариев. Примеры:
  - nfv5/m1.mr1; tmpl/c1.mc1; tmpl/c2.mc2;
  - k1.t; c2.next/c1.exp; c1.end; c2.end/c1.clean, c2.clean, k1.reset.
- 27 негативных сценариев. Примеры:
  - nfv5, tmpl, c2.next;
  - t, c2.next, c1.next, c2.next.
- Три *LTL*-формулы. Пример:
  - *G(wasEvent(c2.next) => X(G(wasEvent(c1.next)) or U(wasEvent(c1.next), wasEvent(c1.end))))*
- Шесть контрактов. Пример:
  - *G(wasEvent(nfv5) => wasAction(m1.mr1))* – инвариант;
  - *G((wasEvent(c2.end) or wasAction(c2.mc2)) => X(wasEvent(nfv5) or wasEvent(nfv9) or wasEvent(k1.t)))* – постусловие.

Внедрение – модуль *Top Traffic Monitor* (3)

Сгенерированный управляющий автомат



Граф потока управления и данных, построенный по сгенерированному автомату



Построенный модуль был проверен отделом тестирования и работает у заказчиков уже около года.

Результаты работы (1)

- Предложена функция приспособленности, учитывающая верификацию.
- Предложены операции мутации и скрещивания, учитывающие верификацию.
- Разработан алгоритм генерации автоматов с учетом контрактов.
- Разработан алгоритм генерации автоматов по сценариям работы и темпоральным формулам.
- Разработан верификатор *AutomataVerificator*, приспособленный для поддержки генерации автоматов на основе генетического программирования.
- Разработана технология генерации автоматов с помощью генетического программирования и верификации.
- Создано инструментальное средство *GABP* для генерации автоматов с помощью генетического программирования и верификации.

Результаты работы (2)

- Результаты, полученные в диссертации, были внедрены при построении модуля *Top Traffic Monitor* для определения узлов сети с максимальным трафиком в программном продукте, выпускаемом компанией *ООО ЭВЕЛОПЕРС* (С.-Петербург).
- Результаты работы используются в учебном процессе на кафедре «Компьютерные технологии» НИУ ИТМО в курсе «Автоматное программирование».
- 11 публикаций, из которых три в журналах из перечня ВАК.
- Сделано семь докладов на конференциях, в том числе, один доклад на международной конференции *GECCO*:
  - *Egorov K., Tsarev F.* Finite State Machine Induction using Genetic Programming Based on Testing and Model Checking / Proceedings of the 2011 GECCO Conference Companion on Genetic and Evolutionary Computation. NY.: ACM. 2011, pp. 759 – 762.