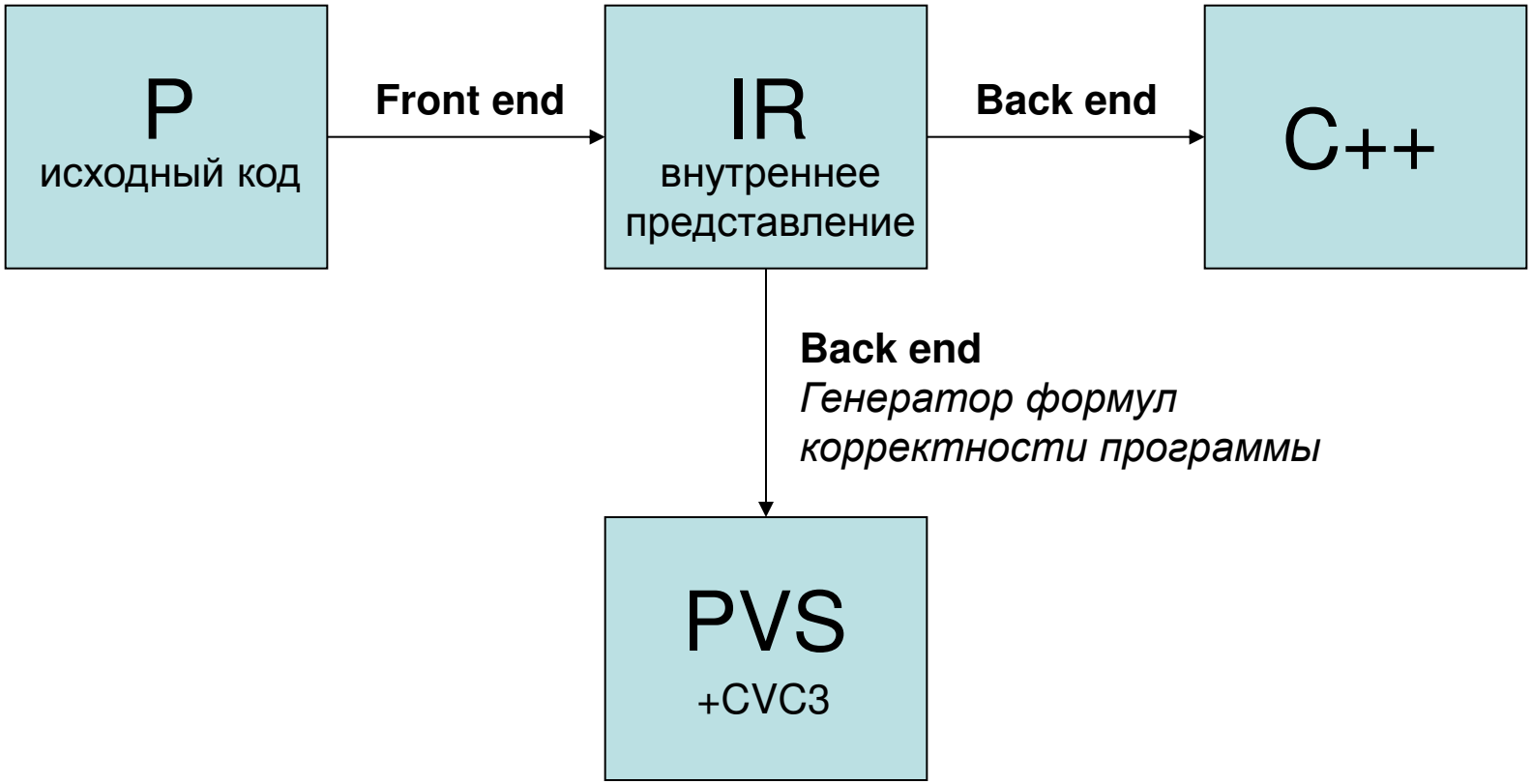
Генерация и доказательство формул корректности предикатных программ

Чушкин М.С
Научный руководитель:
Зав. лаб. ИСИ СО РАН, к.т.н.
Шелехов В.И.

Система предикатного программирования

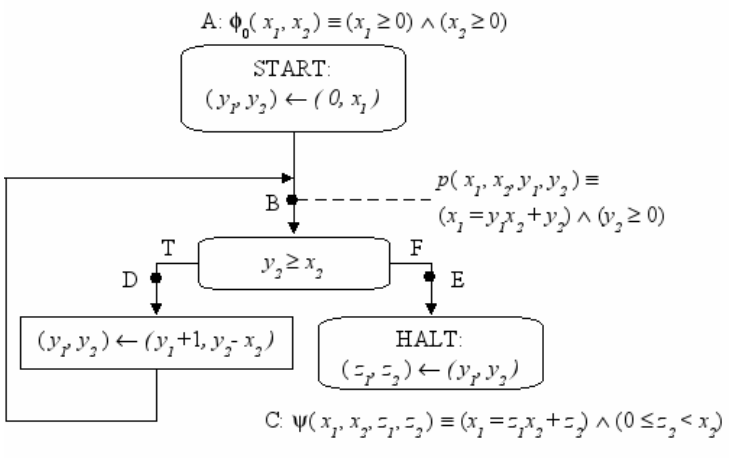


Постановка задачи

- Цель:
  - разработка и реализация системы верификации программ на языке P.
- Задачи:
  - реализация генератора формул корректности;
  - трансляция формул во внутреннее представление SMT решателя CVC3;
  - трансляция формул на язык спецификаций системы PVS;
  - апробация разработанной системы верификации.

Классические методы

- Метод Флойда
  - “Assigning meanings to programs”, 1967



- Метод Хоара
  - “An axiomatic basis for computer programming”, 1969

$$\frac{\{P\}\ S\ \{Q\}\ ,\ \{Q\}\ T\ \{R\}}{\{P\}\ S;T\ \{R\}}$$

$$\frac{\{P \wedge B\}\ S\ \{P\}}{\{P\}\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{done}\ \{\neg B \wedge P\}}$$

$$\frac{\{B \wedge P\}\ S\ \{Q\}\ ,\ \{\neg B \wedge P\}\ T\ \{Q\}}{\{P\}\ \textbf{if}\ B\ \textbf{then}\ S\ \textbf{else}\ T\ \textbf{endif}\ \{Q\}}$$

Определение предиката

```
A(X x: Y y)
pre P(x)
{
     S(x: y)
}
post Q(x, y)
measure m(x);
```

- A – *имя* предиката
- S – *оператор*
  - x, y – аргументы и результаты
- [P(x), Q(x, y)] – *спецификация*
  - P(x) – предусловие
  - Q(x, y) – постусловие
- m(x) – функция *меры*

Пример

```
НОД(nat a, b : nat c)
pre a >= 1 & b >= 1
{
  if (a = b)
     c = a
  else if (a < b)
     НОД(a, b - a : c)
  else
     НОД(a - b, b : c)
}
post gcd(c, a, b)
measure a + b;
```

Корректность программы

- L(S(x: y))
  - *логика оператора* S(x: y)
  - *сильнейший предикат, истинный при завершении исполнения оператора S(x: y)*

- Corr(S, P, Q)(x)
  - *Корректность оператора* S(x: y)
  - $P(x) \rightarrow [\ L(S(x: y)) \rightarrow Q(x, y)\ ] \ \&\ \exists y.\ L(S(x: y))$

- Corr*(A, P, Q)(x)
  - Корректность рекурсивного предиката A(x: y)
  - $Induct(A, P, Q)(t) \rightarrow Corr(A, P, Q)(t)$
  - $Induct(A, P, Q)(t)$
    $\cong \forall u\ (m(u) < m(t) \rightarrow Corr(A, P, Q)(u))$

Примеры правил вывода

Условный оператор:

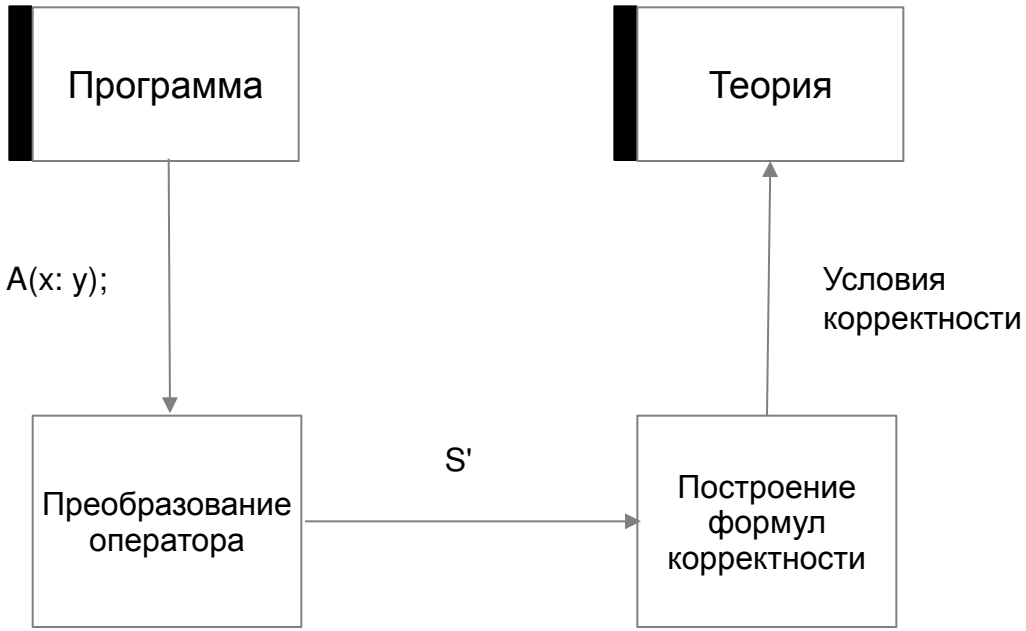
$$\frac{\mathrm{Corr}(B,\ P \,\&\, E,\ Q)(x);\quad \mathrm{Corr}(C,\ P \,\&\, \neg E,\ Q)(x)}{\mathrm{Corr}(\textbf{if}\ (E)\ B(x:y)\ \textbf{else}\ C(x:y),\ P,\ Q)(x)}$$

Оператор суперпозиции:

$$\frac{P(x) \Rightarrow \exists z.\ L(B(x:z));\quad \mathrm{Corr}(C,\ P \,\&\, L(B(x:z)),\ Q)(x)}{\mathrm{Corr}(B(x:z);\ C(z:y),\ P,\ Q)(x)}$$

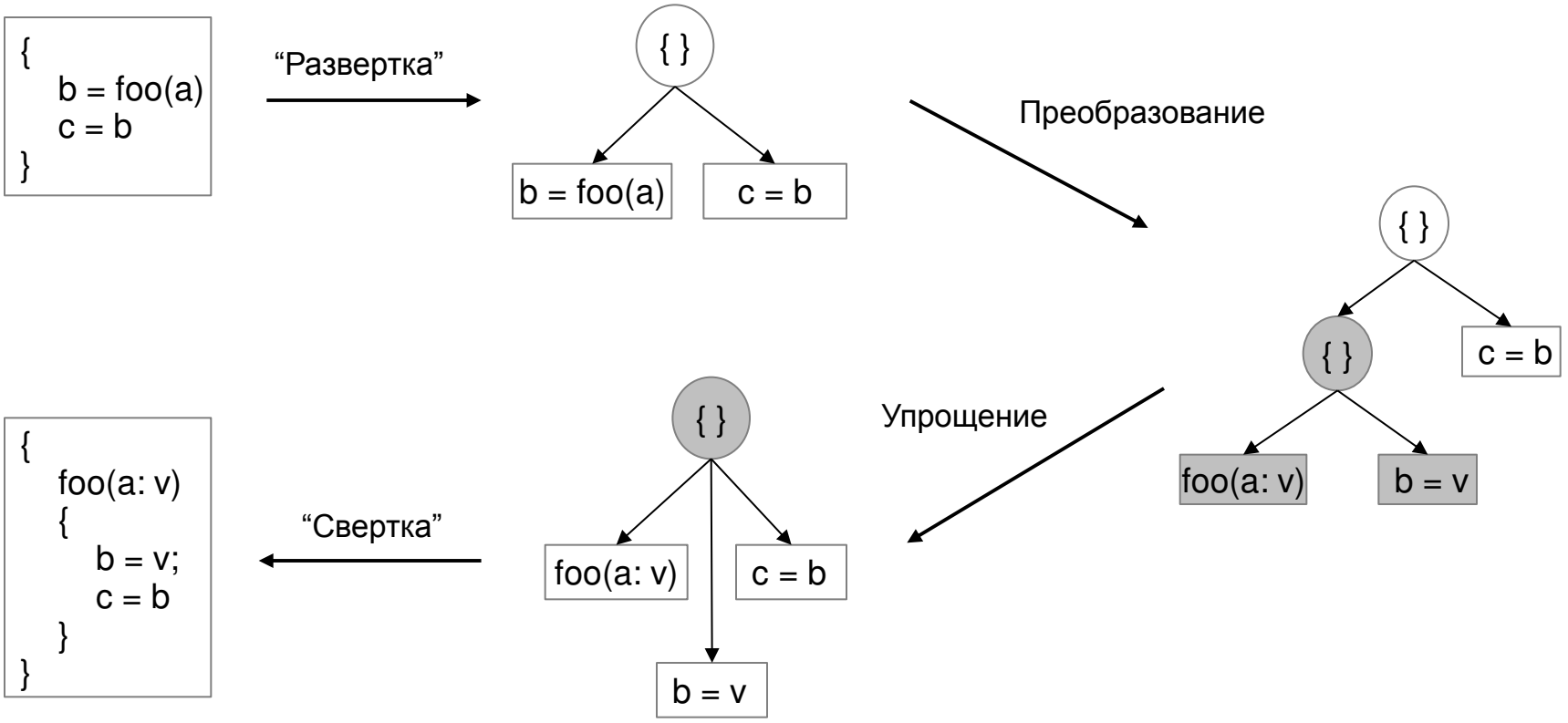
Генерация условий корректности

- Схема работы генератора



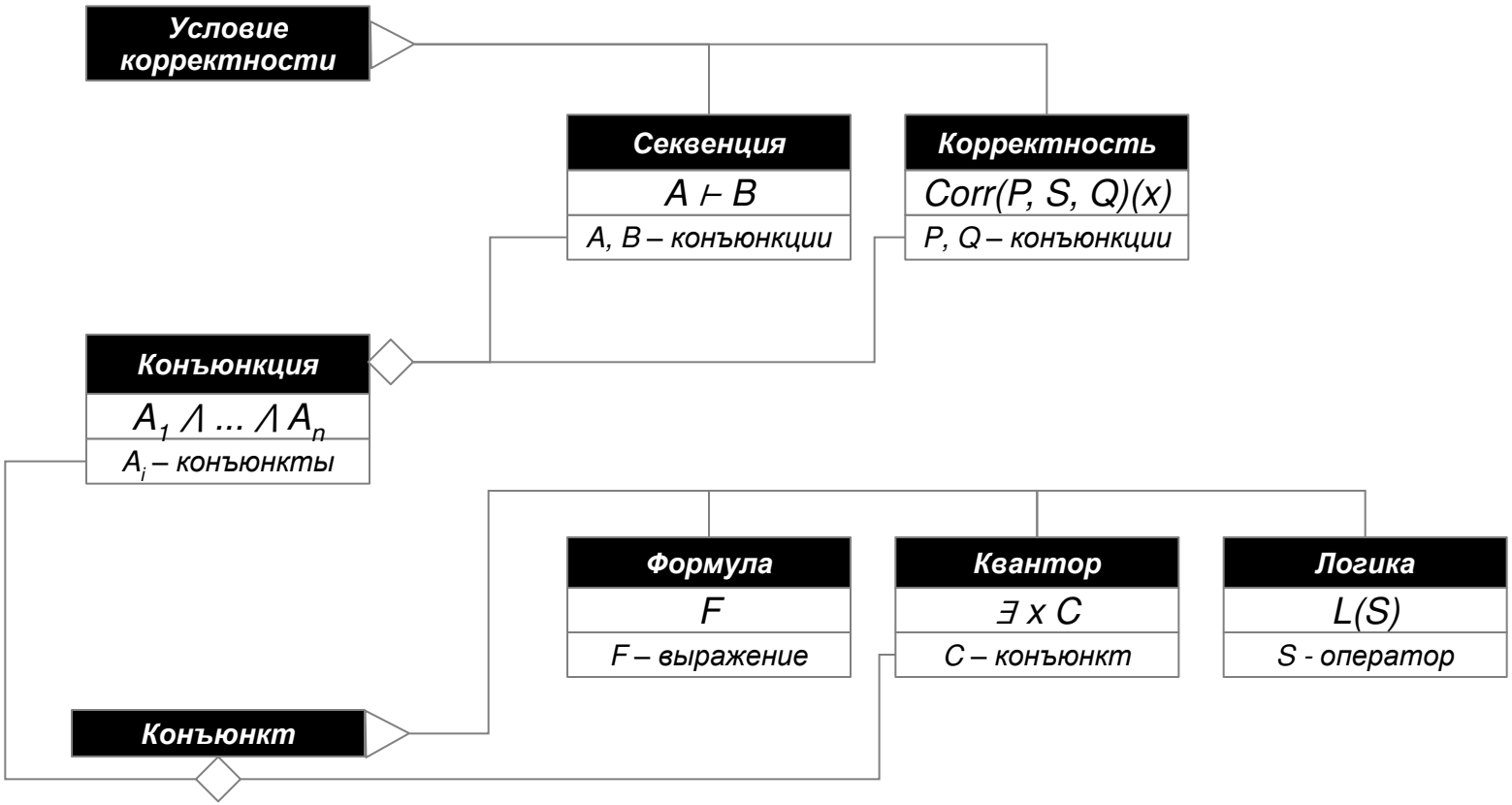
  - Преобразование оператора
  - Построение формул корректности

Преобразование оператора



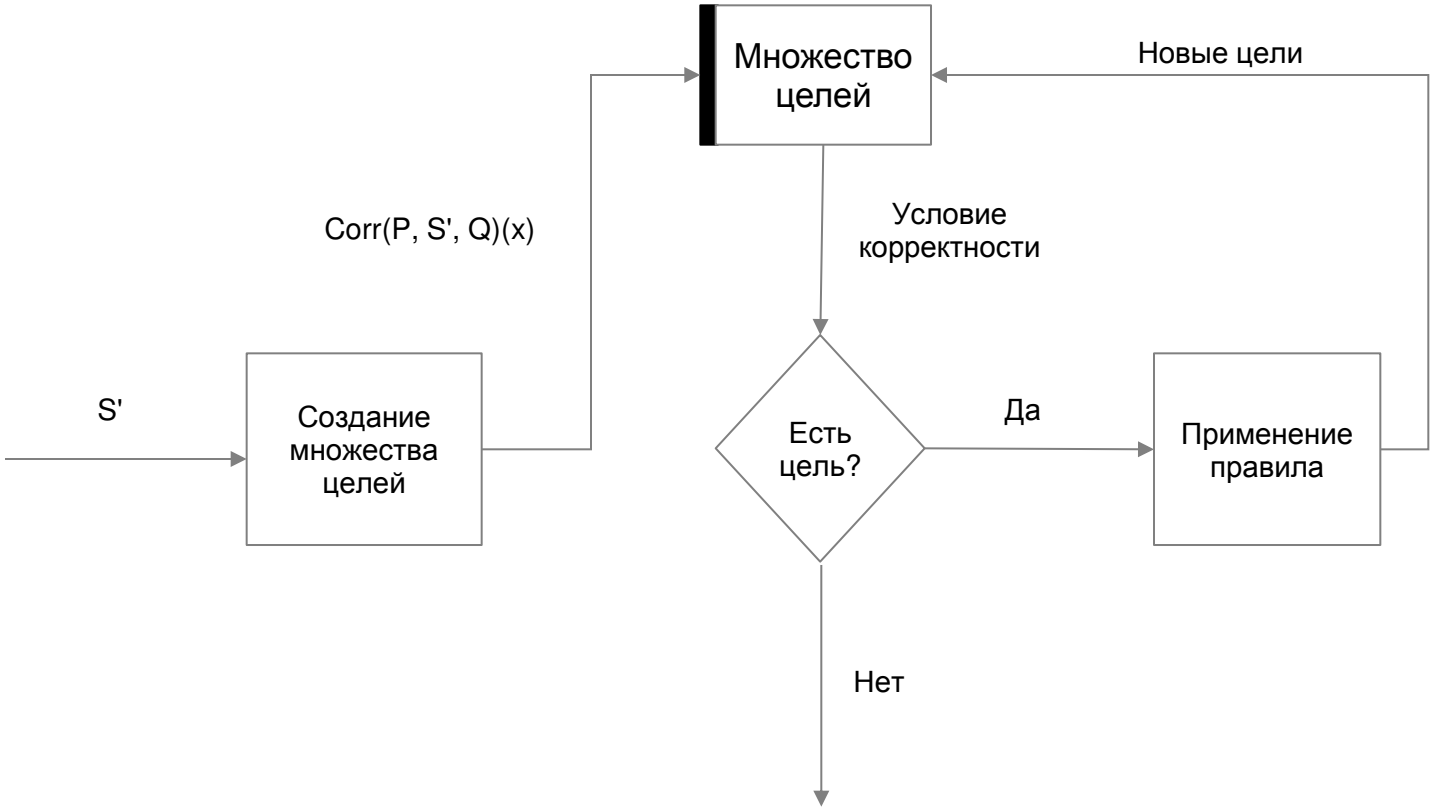
Построения формул корректности

- Иерархия классов внутри генератора



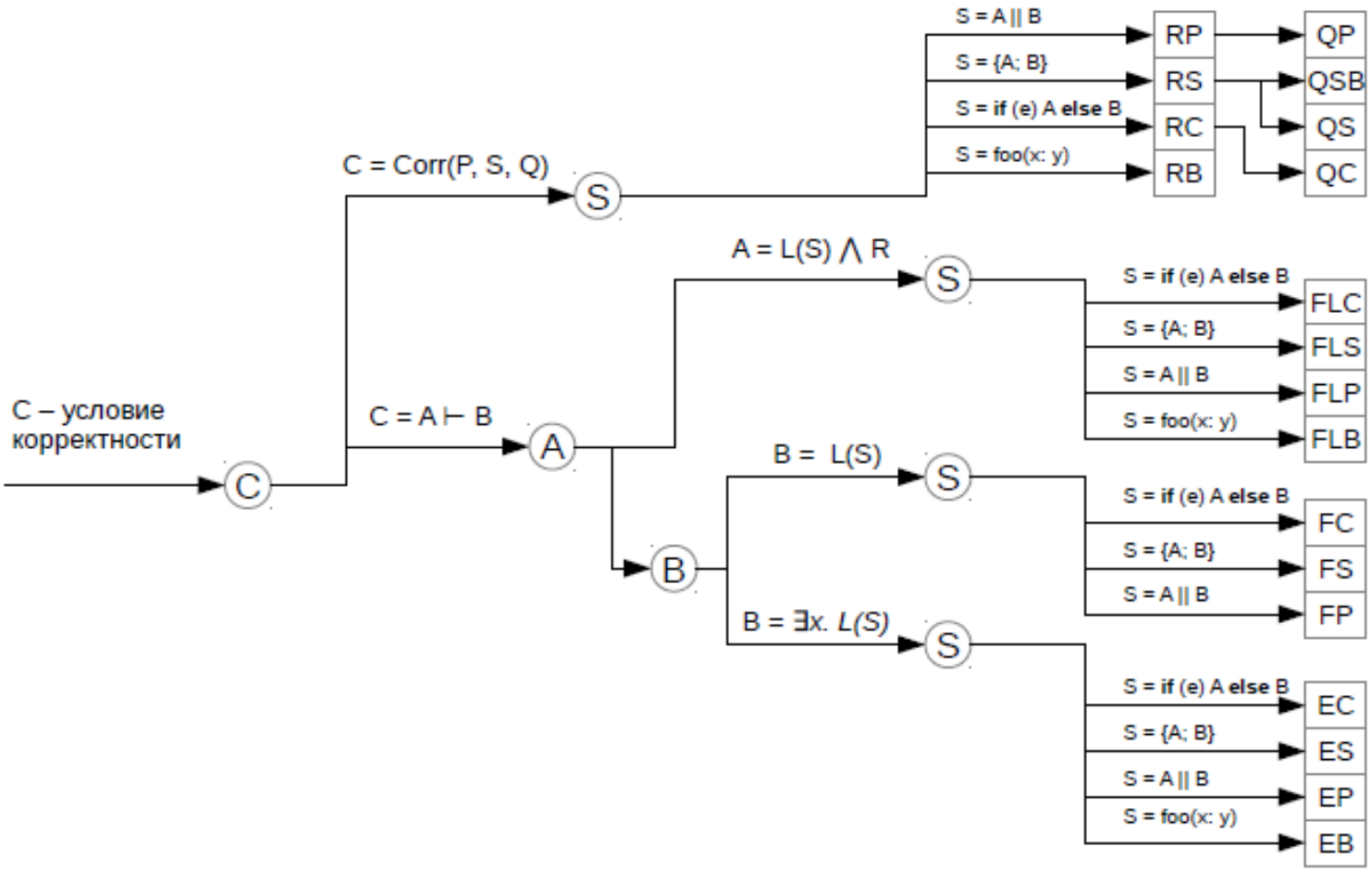
Построения формул корректности

- Схема построения формул корректности



Построения формул корректности

- Схема применения правил вывода



Корректность алгоритма

- Корректность правил (soundness)

  Проведено доказательство в системе PVS

  http://www.iis.nsk.su/persons/vshel/files/rules.zip

- Корректность реализации

  Проверялась тестированием

Пример

```
// Formulas
formula P(nat a, b) = a >= 1 & b >= 1;
formula Q(nat a, b, c) = gcd(c, a, b);
formula m(nat a, b : nat) = a + b;

// Lemmas
lemma forall nat a, b. P(a, b) & a = b => exists nat c. c = a;
lemma forall nat a, b, c. P(a, b) & a = b & c = a => Q(a, b, c);
lemma forall nat a, b. P(a, b) & a != b & a < b
                        => m(a, b - a) < m(a, b) & P(a, b - a);
lemma forall nat a, b. P(a, b) & a != b & a >= b
                        => m(a - b, b) < m(a, b) & P(a - b, b);
```

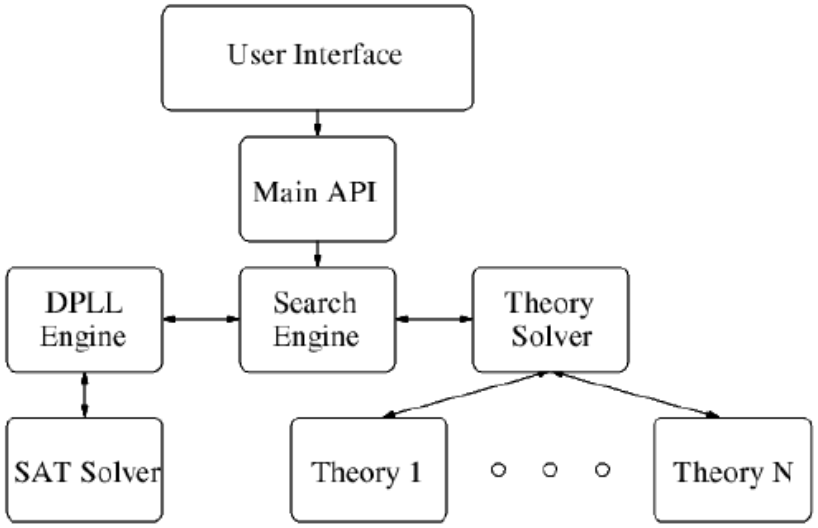
Трансляция на **CVC3**

CVC3

- *система автоматического доказательства для задачи **SMT***

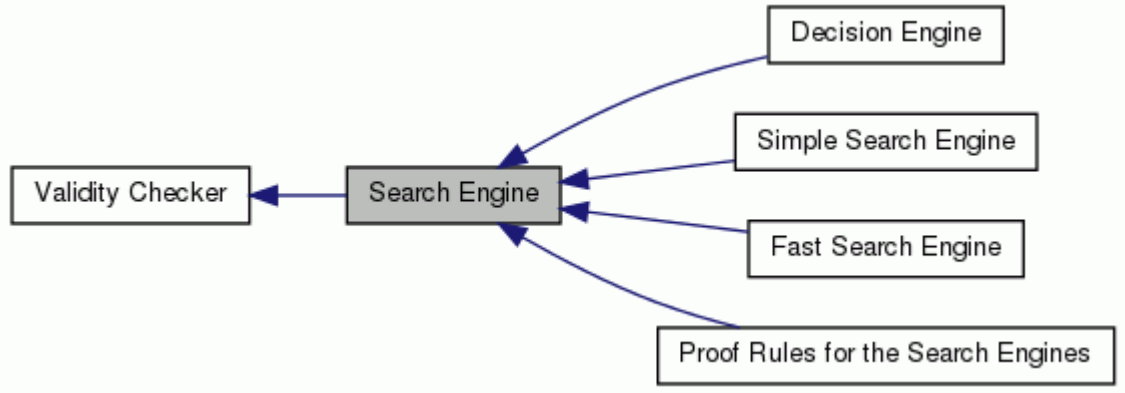
- Достоинства
  - Кванторные выражения
  - Подтипы
  - API

- Недостатки
  - Рекурсия
  - Параметрические типы
  - …



Трансляция на **CVC3**

- Задачи трансляции
  - Трансляция типов
  - Трансляция выражений
  - Анализ результата работы решателя



Пример

lemma valid forall nat c, **nat** a, **nat** b.
P_Умн(a, b) & a = 0 & c = 0 => Q_Умн(a, b, c);

lemma unknown forall nat a, **nat** b.
P_Умн(a, b) & a = 0 => (**exists** nat c. c = 0);

lemma valid forall nat c, **nat** a, **nat** b, **nat** d.
Q_Умн(a - 1, b, d) & P_Умн(a, b) & a != 0 & c = b + d
=> Q_Умн(a, b, c);

lemma unknown forall nat a, **nat** b, **nat** d.
Q_Умн(a - 1, b, d) & P_Умн(a, b) & a != 0
=> (**exists** nat c. c = b + d);

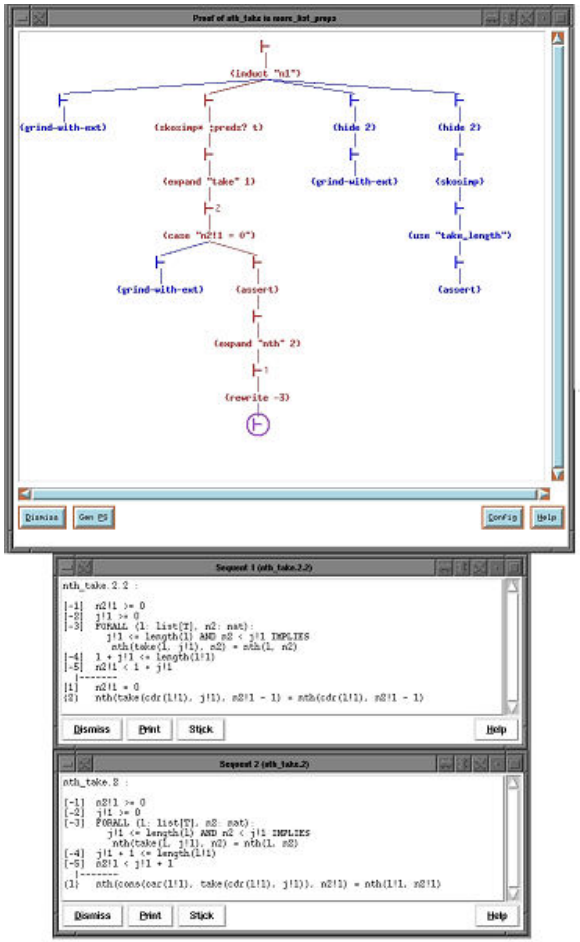
lemma valid forall nat a, **nat** b, **nat** e, **nat** f, **nat** g, **nat** h.
h = a - 1 & f = a - 1 & g = b & a >= 1 & e = b
=> g = e & h = f;

Трансляция на **PVS**

PVS

- *Интерактивный доказыватель теорем (theorem prover)*

- Преимущества
  - Обладает языком спецификаций высокого уровня
  - Модуль доказательства имеет развитую систему команд



Трансляция на **PVS**

- Задачи трансляции
  - Трансляция формул и лемм
    - *lemma* E_1 – лемма
  - Трансляция типов
  - Трансляция выражений
    - *a & b or c* – выражение
  - Трансляция идентификаторов

Пример

```
L1: LEMMA
    FORALL (number: nat, j: nat):
        ((1 /= number) AND (0 = (rem(number)(j))) AND P_MaxPrime(number))
            IMPLIES ((m_MaxPrime(number / j, j) < m_MaxPrime(number, j))
                        AND (0 /= j) AND P_MaxPrime(number / j))

L2: LEMMA
    FORALL (number: nat, j: nat):
        ((0 /= (rem(number)(j))) AND (1 /= number) AND P_MaxPrime(number))
            IMPLIES ((m_MaxPrime(number, 1 + j) < m_MaxPrime(number, j))
                        AND P_MaxPrime(number))

L3: LEMMA
    FORALL (number: nat, j: nat, result: nat):
        ((1 /= number) AND (0 = (rem(number)(j))) AND P_MaxPrime(number)
          AND Q_MaxPrime(number / j, j, result))
            IMPLIES Q_MaxPrime(number, j, result)
```

Анализ работы системы верификации

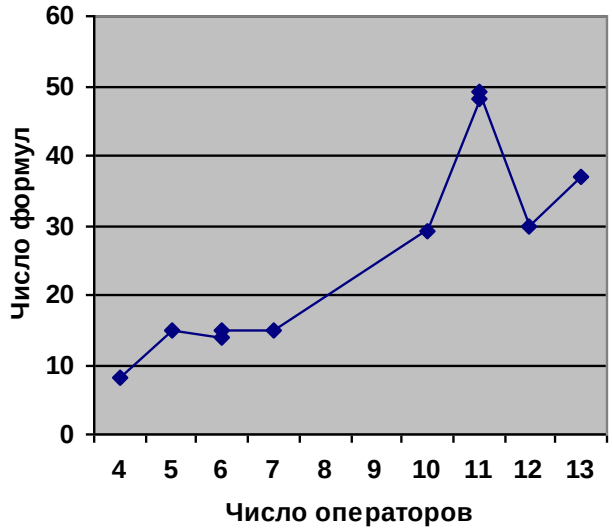
- *«Формальные методы в описании языков и систем программирования»*

- В тестировании участвовало ***10*** программ
- Было сгенерировано ***363*** формулы
  - ***103*** формулы семантики
  - ***260*** формул корректности
- ***14%*** формул оказались тривиальны

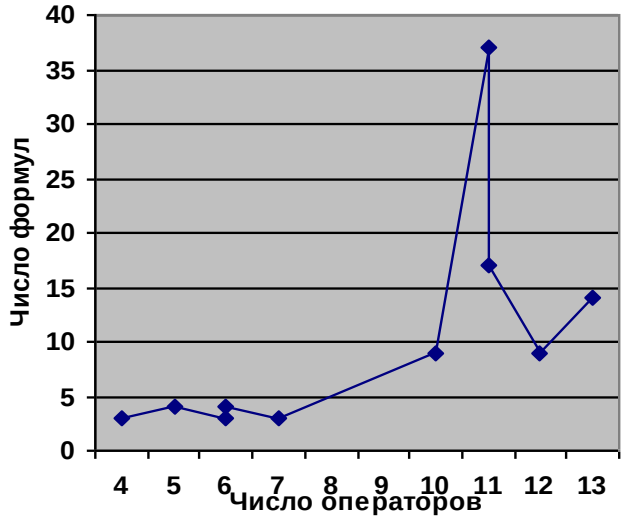
Анализ работы системы верификации

- Анализ работы генератора формул корректности

Формулы корректности:



Формулы семантики:



- Зависимость почти линейная
- Недостаточный объем выборки

Анализ работы системы верификации

- Анализ работы решателя **CVC3**

Семантика		Корректность		Итог	
CVC3	Всего	CVC3	Всего	CVC3	Всего
75	103	117	260	192	363
%					
0.72	1	0.45	1	0.53	1

- ***72%*** формул семантики прошли проверку в **CVC3**
- ***45%*** формул корректности прошли проверку в **CVC3**
- ***53%*** формул прошли проверку в **CVC3**

Результаты

- Построена система правил вывода формул корректности;
- Построена модель системы в **PVS**;
- Реализован генератор формул корректности;
- Реализована трансляция на **CVC3**;
- Реализована трансляция на **PVS**;
- Проведена апробация системы верификации.