Автоматическое исправление ошибок в программном коде

Гузель Гарифуллина, 471
Научный руководитель: к.т.н., доц. Т.А. Брыксин
Рецензент: ст. преп. Д.В. Луцив

Санкт-Петербург
2018 г.

Основные направления в исправлении ошибок

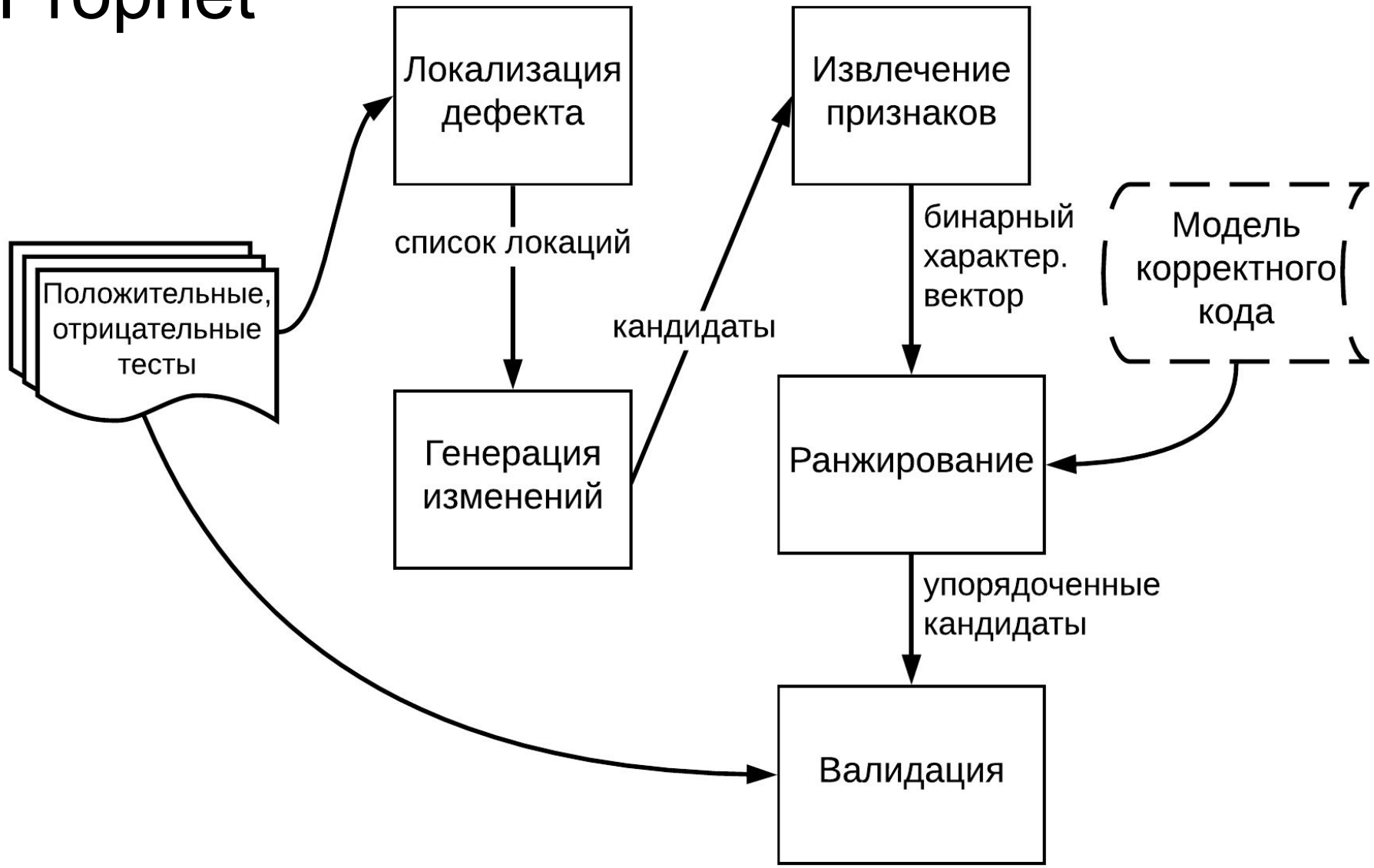
- Исправляющие определенный вид ошибок
- Для программ с формальными спецификациями
- Использующие символьное исполнение
- Generate-and-validate
  - GenProg
  - SPR
  - Prophet (Fan Long and Martin Rinard “*Automatic Patch Generation by Learning Correct Code(2016)*”)

Постановка задачи

Добавить поддержку новых классов ошибок в инструмент Prophet:

- Реализовать генерацию исправлений для новых классов ошибок
- Расширить модель корректного кода
- Провести экспериментальные исследования работы алгоритма

Prophet


Локализация дефекта
Извлечение признаков
список локаций
бинарный характер. вектор
Модель корректного кода
Положительные, отрицательные тесты
кандидаты
Генерация изменений
Ранжирование
упорядоченные кандидаты
Валидация

Генерация изменений

- Изменение условия (if tighten, loosen)
- Добавление условия
- Условное добавление управляющего утверждения (control flow statement)
- Добавление инициализации переменной
- Изменение значения переменной
- Изменение утверждения
- Добавление измененного утверждения

Модель корректного кода

- Признаки программных значений
  - Переменная или константа
  - Роль в исходной программе (внутри условия, параметр функции, ...)
  - Роль внутри исправления
- Признаки модификаций
  - Вид исправления
  - Виды утверждений, которые окружают подозрительную локацию

Пример работы Prophet

```
#define PATH_SIZE 60
char filename[PATH_SIZE];
FILE *fin = fopen(argv[1],
"r");
for (int i = 0;i <= PATH_SIZE;
i++) {
    char c = getc(fin);
    if (c == EOF) {
       filename[i] == '\0';
       break;
    }
    filename[i] = c;
}
fclose(fin);
```

```
for (int i = 0;i <= PATH_SIZE;
i++) {
    if (((i == 60)))
        break;
    char c = _IO_getc(fin);
    if (c == EOF) {
        filename[i] == '\0';
        break;
    }
    filename[i] = c;
}
...
```

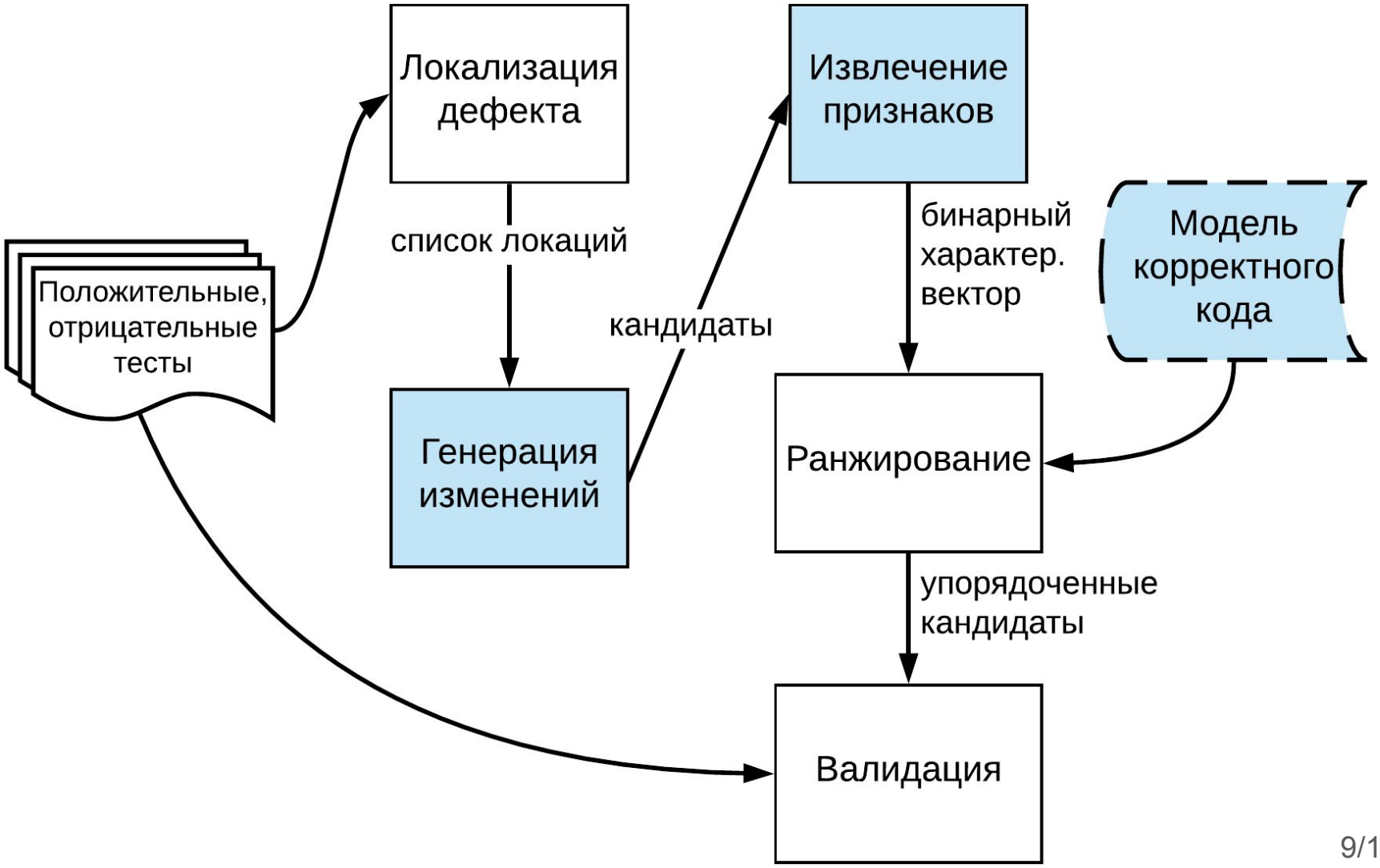
Расширение пространства генерируемых исправлений

Php-ecb9d80

```
JSON_G(error_code) = PHP_JSON_ERROR_NONE;
php_json_encode(&buf, parameter, options
TSRMLS_CC);
ZVAL_STRINGL(return_value, buf.c, buf.len, 1);
…

PHP_JSON_API void php_json_encode(smart_str *buf,
zval *val, int options TSRMLS_DC) {
    JSON_G(error_code) = PHP_JSON_ERROR_NONE;
    switch (Z_TYPE_P(val))
    {
        case IS_NULL:
          …
```

Реализация исправления



Генерация исправлений

- Первичный обход файла
- Фильтрация лишних кандидатов

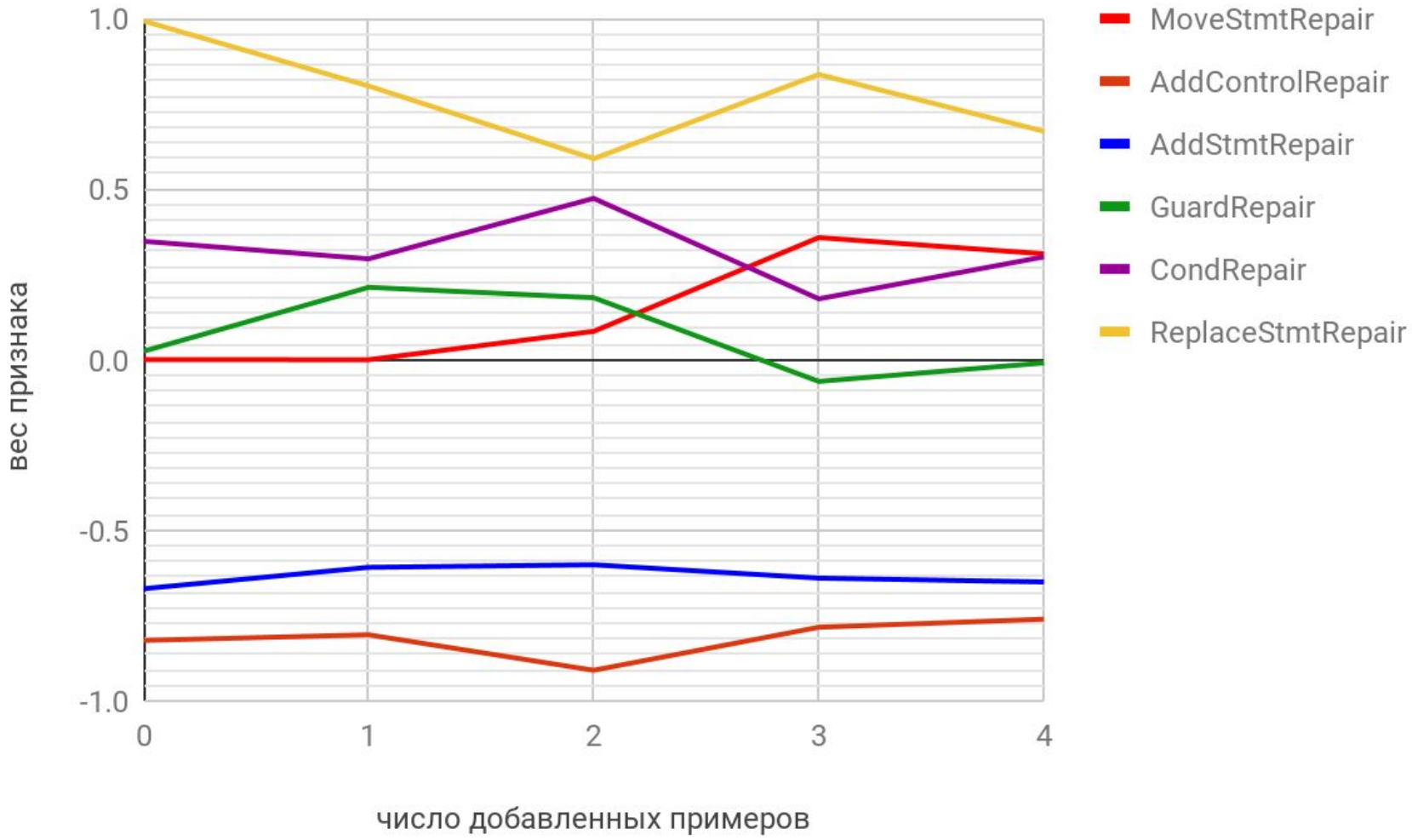
Новая модель корректного кода

- Признаки
  - MoveStmtRepair
  - 2 места
- Извлечение признаков
- Скрипт сравнивающий исходное и сгенерированное изменение
- Добавление новых примеров в датасет
- Обучение

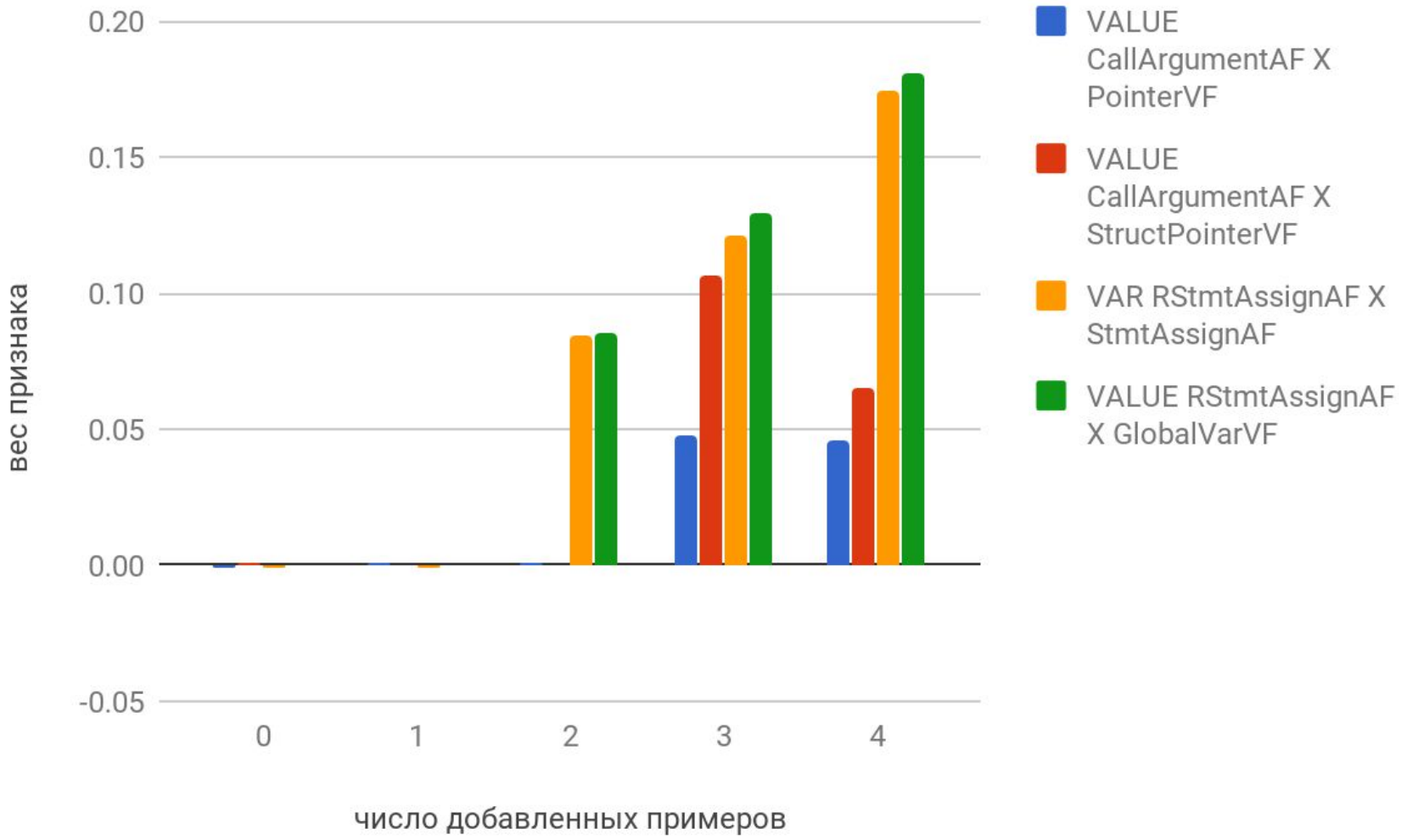
Виды исправлений в датасете

Программа / Вид исправ.	php	python	subversion	другие	всего
TightenCondition	72	38	103	103	316
Guard	22	18	41	57	138
ReplaceString	26	18	41	18	103
IfExit	27	15	8	21	71
AddAndReplace	24	15	10	19	68
Replace	10	7	34	12	63
SpecialGuard	4	3	3	6	16
AddInit	2	0	0	0	2
FunctionMutation	1	0	0	0	1
все	188	114	240	236	778

Изменение веса признаков вида исправления



Новые признаки (вторая локация)



В результате сгенерировалось

```
//prophet generated patch
(json_globals.error_code) = PHP_JSON_ERROR_NONE;
php_json_encode(&buf, parameter, options TSRMLS_CC);


 PHP_JSON_API void php_json_encode(smart_str *buf, zval
*val, int options TSRMLS_DC) {
        //prophet generated patch
        if (0)
           (json_globals.error_code) = PHP_JSON_ERROR_NONE;
        switch (Z_TYPE_P(val))
        {
                case IS_NULL:
                ...
```

Результаты

- Реализована генерация исправлений для новых классов ошибок, исправляемых вынесением первого утверждения функции перед вызывающим кодом
- Расширена и обучена новая модель корректного кода посредством добавления новых признаков в модель и примеров в тренировочный набор данных
- Проведена апробация решения, проведен анализ влияния новых тренировочных примеров на параметры модели корректного кода и выявлены самые значимые новые признаки модели

Новая модель корректного кода

$$F = R + n * M$$
$$M = 3 * R * A + 3 * A^2 + A * V$$

R – вид исправления;
M – все исходные признаки состояния и признаки модификаций без вида исправления;
n – максимальное число действий, поддерживаемое изменением
A -- атомные характеристики (вид оператора, вид утверждения, разыменование указателя, ...)
V -- признаки программных значений (константа, локальная переменная, ...)

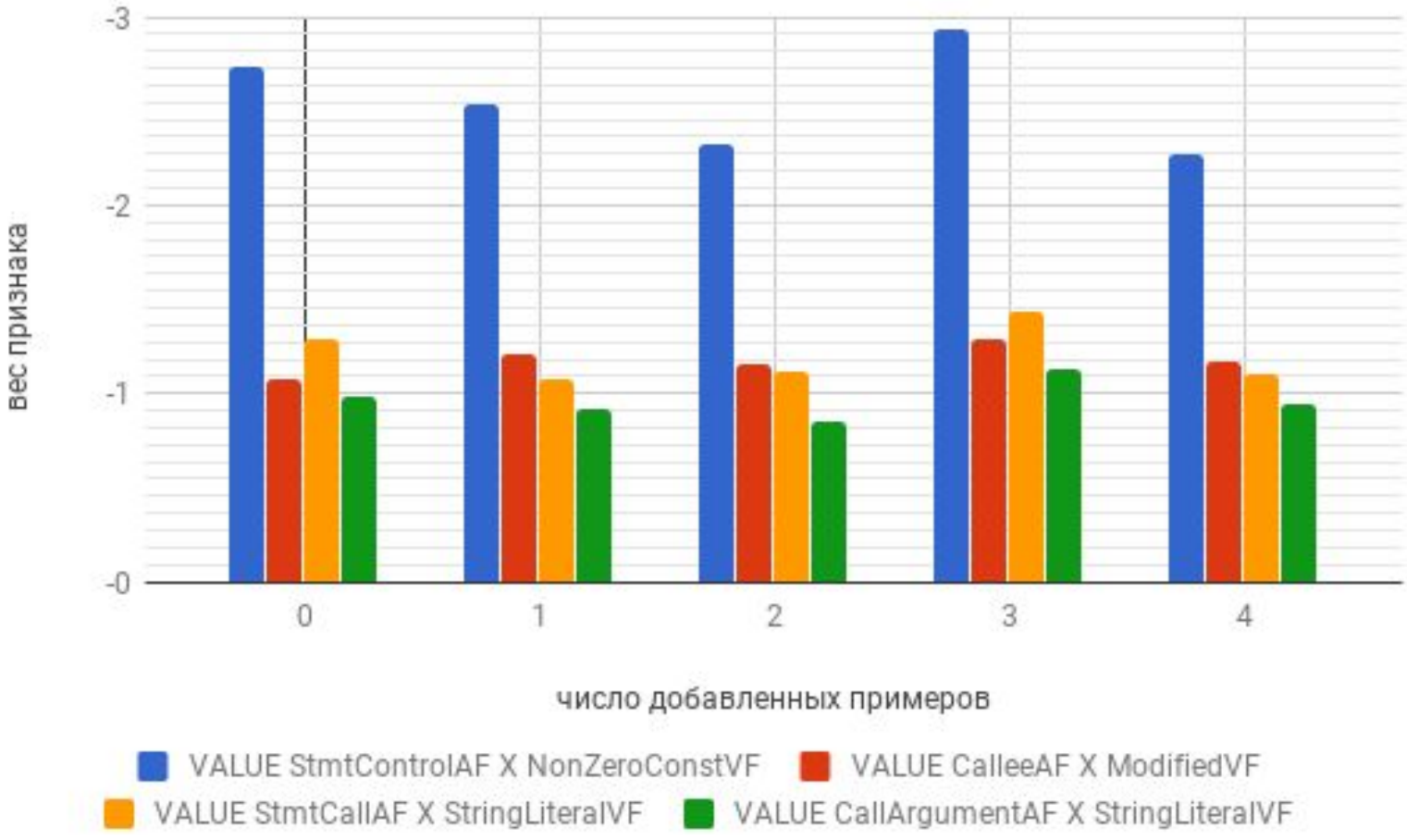
Результаты исправления ошибок

Дефект	Пространство поиска	Ранг до	Ранг после	Function Mutation
gzip-a1d3d4-f17cbd	47602	1929	1957	0
php-307914-307915	45389	1	1	0
php-308734-308761	14536	5376	4191	0
php-309111-309159	52908	7701 (-)	1138	0
php-309579-309580	51306	767	314	0
php-309892-309910	36940	462	632	0
php-310991-310999	87574	907	305	1
php-308525-308529	42587	-	2637	3

Изменение ранга исправления от числа добавленных примеров

Число примеров	Ранг схемы	Оценка
0	10354	-8.68
1	11547	-8.72
2	4935	-7.27
3	1512	-6.11
4	544	-5.53

Отрицательные признаки(первой локации)



Положительные признаки(первой локации)