오류 자동 수정 벤치마크 소개

오학주

고려대학교

APR Research @Korea Univ.

홍성준 (PhD'24)

이준희 (PhD'24)

소순범 (PhD'22)

송도원

오원석

- C / Java: MemFix [FSE 2018], SAVER [ICSE 2020], NPEX [ICSE 2022], SPRINT [PLDI 2024]
- OCaml: FixML [OOPSLA 2018], CAFE [FSE 2021]
- Solidity: SmartFix [FSE 2023]
- Python: PyTER [FSE 2022]

Benchmarks and Tools

- Benchmark: https://github.com/kupl/starlab-benchmarks

	Language	Bug Type	# of Bugs
Safety	C	Memory Leak	200
	Java	Null Pointer Exception	500
	Solidity	Integer Overflow	300
Functional	Java	Functional	100
	OCaml		300

- Tool: https://github.com/kupl/kaprese

kaprese

Collection of APR tools by SAL (Software Analysis Lab) at Korea University.

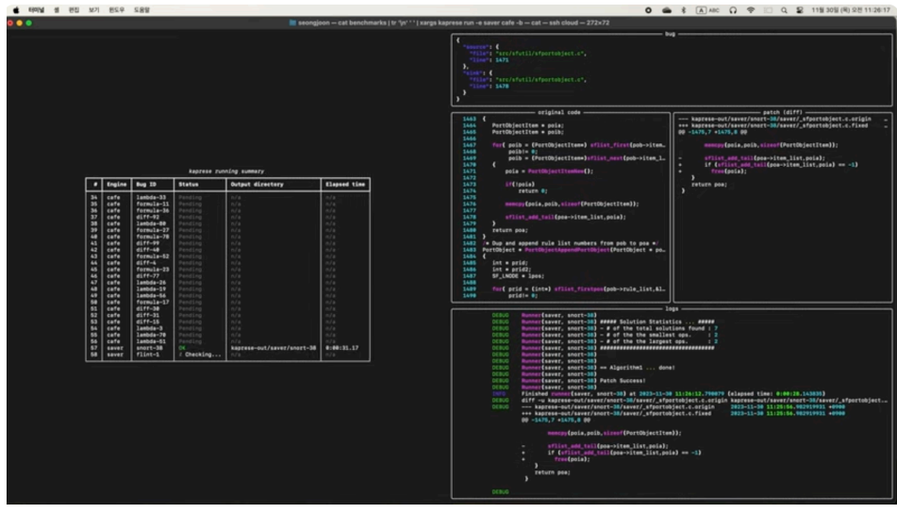
Quick Start

You can run `kaprese` by following the steps below.

Prerequisites

`kaprese` requires the following software to be installed:

- Python 3.12 or higher (required)
- Docker 24 or higher (recommended, we did not test it on other versions)



https://www.youtube.com/watch?v=_nphp8KBuUI

Memory Leak in C

```
1   int swTableColumn_add(swTable *table, ...) {
2     col = sw_malloc(sizeof(swTableColumn));
3     if (type == SW_TABLE_INT)
4         col->size = 1;
5     col->index = table->size;
6     return swHashMap_add(table->columns, ..., col);
7   }
8
9   int swHashMap_add(swHashMap *hmap, ..., void *data) {
10    node = sw_malloc(sizeof(swHashMap_node));
11    if (node == NULL)
12        return SW_ERR;
13    node->data = data;
14    swHashMap_node_add(hmap, ... node);
15    return SW_OK;
16  }
```

정상 실행 경로

Memory leak

Null Pointer Exceptions in Java

```
public StrBuilder appendFixedWidthPad(Object obj, int width, char padChar)
{
        String str = (obj == null ? getNullText() : obj.toString());
        int strLen = str.length();

                ...

        this.size += width;
        return this;
}
```

Integer Overflow in Smart Contracts

```
balance[from] = balance[to] = balance[msg.sender] = 0
value: 8fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fee  : 7000000000000000000000000000000000000000000000000000000000000001
```

```
1   function transferProxy (address from, address to, uint
          value, uint fee) public returns (bool) {
2       if (balance[from] < fee + value)
3           revert();
4       if (balance[to] + value < balance[to] ||
5           balance[msg.sender] + fee < balance[msg.sender])
6           revert();
7       balance[to] += value;
8       balance[msg.sender] += fee;
9       balance[from] -= value + fee;
10      return true;
11  }
```

false

0!

false

8fffff…ff

700…00

0!

Functional Bugs (Java, OCaml)

OCaml/diff-1/buggy/src.ml

```
exception EMPTYLIST

type aexp =
  | Const of int
  | Var of string
  | Sum of aexp list
  | Times of aexp list
  | Power of (string * int)

let rec diff ((aexp : aexp), (str : string)) : aexp =
  match aexp with
  | Const a -> Const 0
  | Var b -> if str = b then Const 1 else Const 0
  | Power (pstr, i) ->
      if str = pstr then Times [ Const i; Power (pstr, i
  | Times alst -> (
      match alst with
      | [] -> Const 0
      | [ h ] -> diff (h, str)
      | h :: t ->
          Sum [ Times (diff (h, str) :: t); Times [ h; d
  | Sum [] -> Const 0
  | Sum (h :: []) -> Const 0
  | Sum (h :: t) -> Sum [ diff (h, str); diff (Sum t, st
```

OCaml/diff-1/buggy/testcases

```
{
  (Const 1,"x") ; [("x",1)] => 0;
  (Var "y","x") ; [("y",1)] => 0;
  (Power ("x", 1), "y") ; [("x", 1)] => 0;
  (Power ("x", 1), "x") ; [("x", 2)] => 1;
  (Power ("x", 0), "x") ; [("x", 1)] => 0;
  (Power ("x", 1), "x") ; [("x", 1)] => 1;
  (Power ("x", 2),"y") ; [("x",5)] => 0;
  (Power ("y", 5), "y") ; [("y", 2)] => 80;
  (Times [Var "x"], "x") ; [("x", 3)] => 1;
  (Times [Var "x"; Var "x"], "x") ; [("x", 0)] => 0;
  (Times [Var "x"; Var "x"], "x") ; [("x", 1)] => 2;
  (Times [Const 1], "x") ; [("x", 1)] => 0;
  (Times ([Const 1; Var "x"]), "x") ; [("x", 1)] => 1;
  (Sum [Const 0; Var "x"],"x") ; [("x",10)] => 1;
  (Sum [Power ("x", 2); Times [Const 2; Var "x"]; Const 1
  (Sum [Power ("x", 2); Power ("x", 2); Const 1], "x") ;
  (Sum [Power ("x", 2); Power ("x", 2); Const 1], "y") ;
  (Sum [Const 2; Power ("x",2); Power ("x",3)], "x") ; ['
  (Times ([Const (-1); Const (-1); Var "x"]), "x") ; [("x
  (Times [Power ("x", 3); Power ("y", 2)], "x") ; [("x",
  (Sum [Times [Sum [Var "x"; Var "y"]; Times [Var "x"; Va
  (Times [Times [Sum [Var "x"; Var "y"]; Var "x"]; Var "x
  (Times [Const 2; Sum [Var "x"; Var "y"]; Power ("x", 3)
  (Times [Sum [Var "x"; Var "y"; Var "z"]; Power ("x", 2)
  (Times [Sum [Var "x"; Var "y"; Var "z"]; Power ("x", 2)
  (Sum ([Var ("x"); Var ("x")]), "x") ;[("x", 1)] => 2;
  (Sum ([Const (1)]), "x") ;[("x", 1)] => 0;
  (Times ([Var ("x")]), "x") ;[("x", 1)] => 1;
}
```

Data Format Example

safety/C/error-reports/memory-leak/snort-2.9.13_1.json


```
{
    "err_type": "MEMORY_LEAK",
    "source": {
        "filepath": "src/detection-plugins/sp_appid.c",
        "line": 213
    },
    "sink": {
        "filepath": "src/detection-plugins/sp_appid.c",
        "line": 223
    }
}
```

https://github.com/kupl/kaprese

How to Add Your Own

All benchmarks and engines are registered to the global registry (by default, registry is located in `~/.kaprese`). You can add your own benchmarks and engines by registering them with a small python code.

Add a Benchmark

A benchmark is an instance of `kaprese.core.benchmark.Benchmark`. You can initialize a benchmark with the following 4 parameters:

- `name`: name of the benchmark (must be unique).
- `image`: docker image name, e.g., `ghcr.io/kupl/starlab-benchmarks/c:flint-1`.
- `language_command` or `_language`: command to find the language of the benchmark, if you set `_language`, `language_command` is ignored. the command will be run inside the docker container from `image`.
- `workdir_command` or `_workdir`: command to find the working directory of the benchmark, if you set `_workdir`, `workdir_command` is ignored. the command will be run inside the docker container from `image`. Then by calling `register` method of `Benchmark`, you can register the benchmark to the global registry.

Add an Engine

An engine is an instance of `kaprese.core.engine.Engine`. You can initialize an engine with the following 6 parameters:

- `name`: name of the engine (must be unique).
- `supported_languages`: list of supported languages, e.g., `["c", "java"]`.