Symbiotic-Witch 2

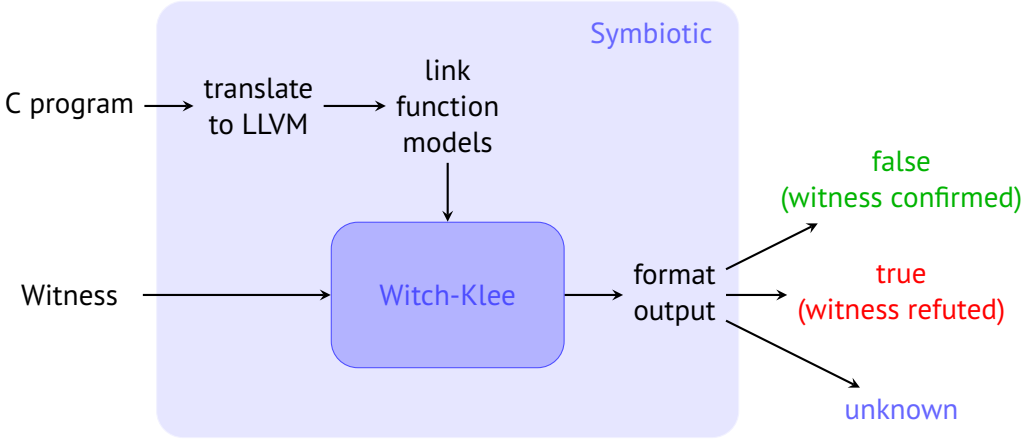
More Efficient Algorithm and Witness Refutation

***Paulína Ayaziová*, Jan Strejček**

Masaryk University

TACAS 2023 - April 24, 2023

Symbiotic-Witch



Validation approach

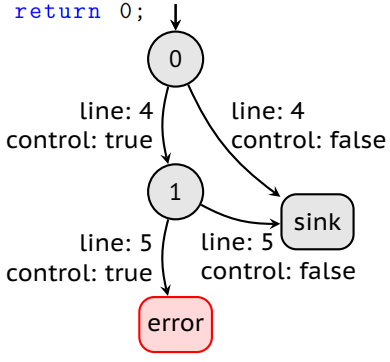
- Execute the program symbolically and simultaneously traverse the witness
- Source-code guards are used to map the witness edges to program instructions during symbolic execution
- State-space guards are used to concretise symbolic values
- Confirm the witness if the specified error is found with the witness traversal in an error node
- Report "unknown" if the exploration ends without confirming the error and
  - the witness uses unsupported source-code guards, or
  - we have concretised to a value not provided by the witness
- Refute the witness otherwise

What's new?

- More precise interpretation of witness semantics
  - Smaller sets of tracked witness nodes
  - Use of sink nodes
- Different handling of return values
- Witness refutation

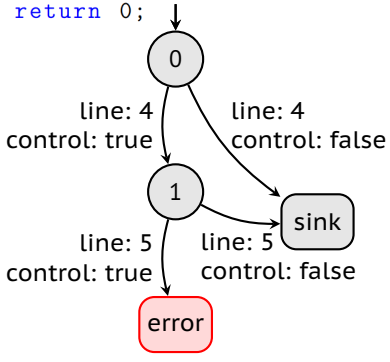
Example

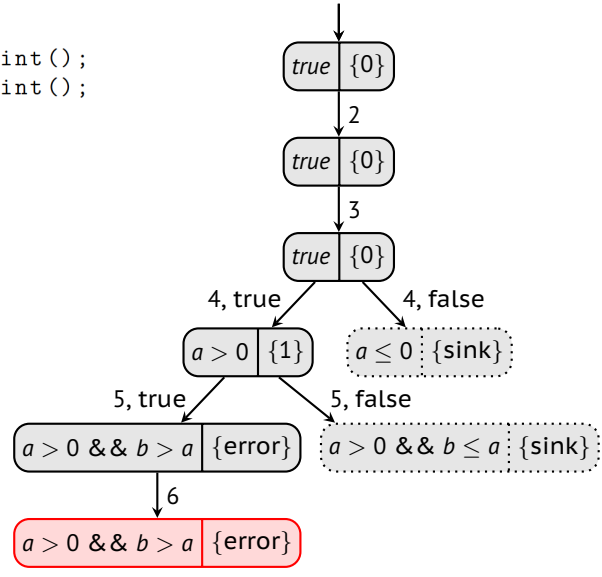
```
1  int main() {
2    int a = __VERIFIER_nondet_int();
3    int b = __VERIFIER_nondet_int();
4    if (a > 0) {
5      if (b > a) {
6        reach_error();
7      }
8    }
9    return 0;
10 }
```



Example

```
1  int main() {
2    int a = __VERIFIER_nondet_int();
3    int b = __VERIFIER_nondet_int();
4    if (a > 0) {
5      if (b > a) {
6        reach_error();
7      }
8    }
9    return 0;
10 }
```





Results summary

- 38,644 correct results
  - 35,536 correctly confirmed witnesses
  - 5,804 correctly refuted witnesses
- 10 incorrect results
- 1st place in Software Systems
- 3rd place overall