04.Concolic execution

List

Concolic testing

- **Concrete Execution과 Symbolic Execution의 장점을 이용한 실행 방식입니다.(Conc~~rete~~ + ~~Symb~~olic = Concolic)**
  - Symbolic execution은 코드 적용 범위를 최대화하기 위해 구체적인 입력값(Concrete value)이 필요합니다.
    - 구체적인 입력 값을 생성하기 위해 Symbolic execution은 제약 논리 프로그래밍(constraint logic programming)을 기반으로하는 자동화된 정리 증명계(automated theorem prover)또는 constraint solver를 함께 사용합니다.
  - 주요 초점은 프로그램의 정확성을 확인하는 것이 아니라 소프트웨어에서 버그를 찾는 것입니다.
  - Concolic 테스트에 적합한 SMT(Satisfiability modulo theories) Solver는 Z3, STP, Z3str2, Boolector, 등이 있습니다.

Example

- **다음 코드를 이용해 Concolic execution을 확인 할 수 있습니다.**
- 해당 프로그램은 사용자로 부터 입력 받은 값으로 인해 3가지의 실행 경로를 확인 할 수 있습니다.
- x, y는 미지수 이며, 해당 값을 기호 값으로 선언합니다.
  - $y = \lambda$
  - $x = \chi$
- Solver를 이용해 다음과 같이 수식을 만족하는 값을 찾을 수 있습니다.

Find path and concrete value

경로	조건	Solved
• z의 값이 1000이 아닐 경우	$((\chi * 2) \neq 1000)$	χ : 500 이외의 값
• z의 값이 1000이고 y의 값이 z보다 작거나 같을 경우	$((\chi * 2) = 1000 , \lambda \leq (\chi * 2))$	χ : 500, λ : 1000과 같거나 작은 값
• z의 값이 1000이고 y의 값이 z보다 클 경우	$((\chi * 2) = 1000, \lambda > (\chi * 2))$	χ : 500, λ : 1000 보다 큰 값

ConcolicTest.c

```
#include <stdio.h>

void main(){
    int x,y,z;

    scanf("%d",&x);
    scanf("%d",&y);

    z = x * 2;

    if(z == 1000){
        if(y > z){
            printf("Nice!\n");
        }else{
            printf("Wrong!\n");
        }
    }
}
```

Algorithm

- **Concolic 테스트 알고리즘은 다음과 같이 동작 합니다.**
    1. 사용자 입력값을 저장하는 변수들은 symbolic execution이 진행되는 동안 기호 변수로 처리 됩니다.
        a. 다른 모든 변수들은 구체적인 값으로 처리됩니다.
    2. 우선 임의의 입력을 선택해 프로그램을 실행합니다.
    3. 실행 과정에서 Concolic 수행에 필요한 심볼들을 추출합니다.
    4. 추출된 심볼을 바탕으로 Symbolic 조건을 생성합니다.
    5. Solver를 이용해 Symbolic 조건에 만족하는 새로운 입력 값을 생성하여 프로그램 재실행
        a. 만족하는 입력값이 없으면 다음 실행 경로를 시도합니다.

Limitations

- **프로그램이 같은 입력을 넣어도 경우에 따라 다른 결과가 나오는 경우 의도한 경로와 다른 경로를 따를 수 있습니다.**
    - 이로 인해 검색이 종료되지 않거나 부실한 범위로 진행될 수 있습니다.
- **같은 입력에 대하여 같은 출력이 되는 프로그램에서도 여러 가지 이유로 인해 크거나 무한한 경로 트리에서 가장 유익한 부분을 찾지 못하는 등 범위가 작아질 수 있습니다.**
    - 빈약한 적용 범위
    - 부정확한 기호 표현
    - 불완전한 법칙 증명
- **암호화 기초요소와 같이 변수의 상태를 완전히 썩는 프로그램의 경우 실질적으로 해결할 수 없는 매우 큰 상직적 표현을 만들어 냅니다.**

Tools

- **Tool Concolic execution .**
    - Angr
    - Triton
    - Ponce
    - .

Related info

- https://en.wikipedia.org/wiki/Concolic_testing
- http://shell-storm.org/talks/SSTIC2015_English_slide_detailed_version_Triton_Concolic_Execution_FrameWork_FSaudel_JSalwan.pdf