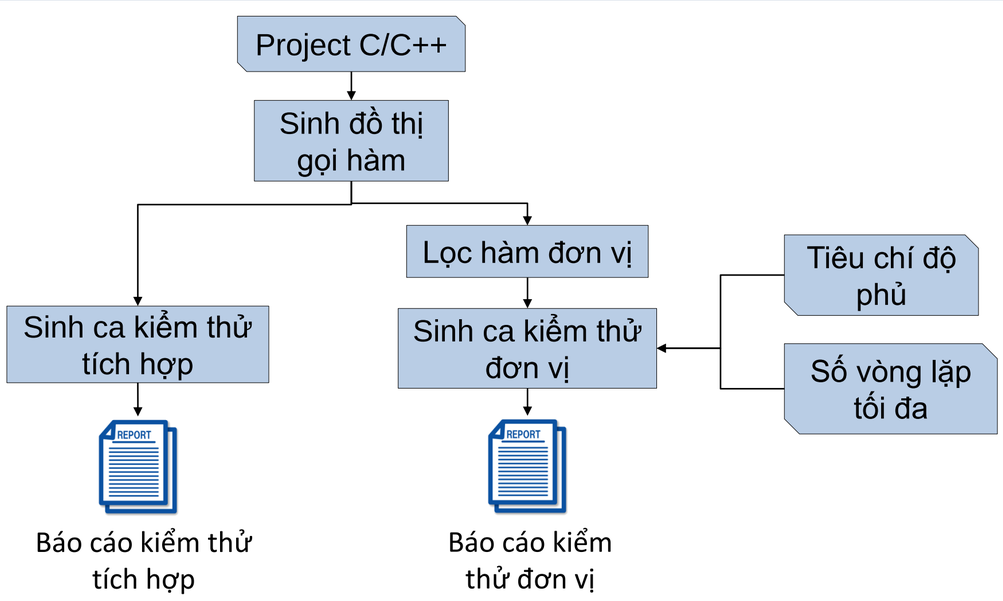
PHƯƠNG PHÁP VÀ CÔNG CỤ KIỂM THỬ TỰ ĐỘNG CHO CÁC ỨNG DỤNG C/C++

Sinh viên: Dương Tuấn Anh, Sầm Đức Vũ, GV hướng dẫn: PGS. TS. Phạm Ngọc Hùng
Khoa Công Nghệ Thông Tin, Trường Đại Học Công Nghệ, Đại Học Quốc Gia Hà Nội

Giới thiệu

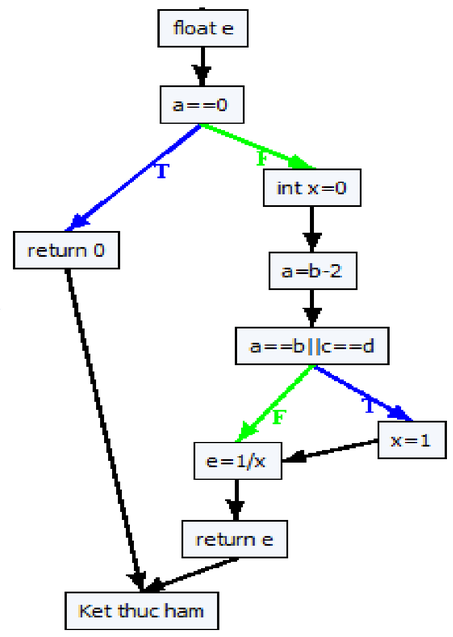
- C/C++ là ngôn ngữ phổ biến xây dựng PM hệ thống, PM điều khiển
- Chi phí kiểm thử tốn hơn rất nhiều với chi phí lập trình
- Trong kiểm thử có nhiều kĩ thuật, một kĩ thuật đảm bảo chất lượng PM là kiểm thử hộp trắng luồng điều khiển
- **Đầu vào**
  + Một project chứa một hoặc nhiều tập tin mã nguồn C/C++
  +Tiêu chí kiểm thử: phủ câu lệnh / phủ nhánh / phủ điều kiện con
- **Đầu ra**: Báo cáo kết quả kiểm thử

Tổng quan PP



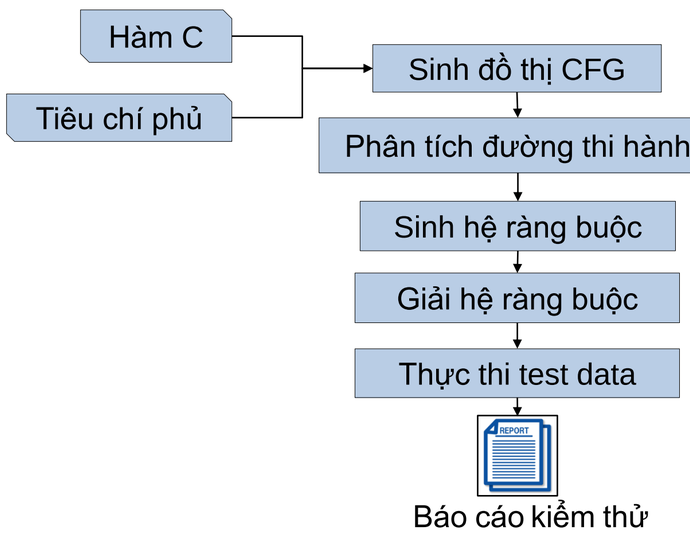
Đề xuất thuật toán sinh CFG

```
float foo(int a, int b,
          int c, int d){
    float e;
    if (a==0)
        return 0;
    int x = 0;
    a = b - 2;
    if (a == b||c == d)
        x = 1;
    e = 1/x;
    return e;
}
```

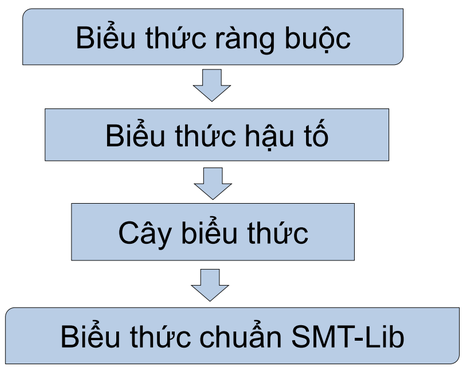


Một trong ba độ phủ
- Phủ câu lệnh ✓
- Phủ nhánh
- Phủ điều kiện con

Đề xuất PP ca kiểm thử đơn vị

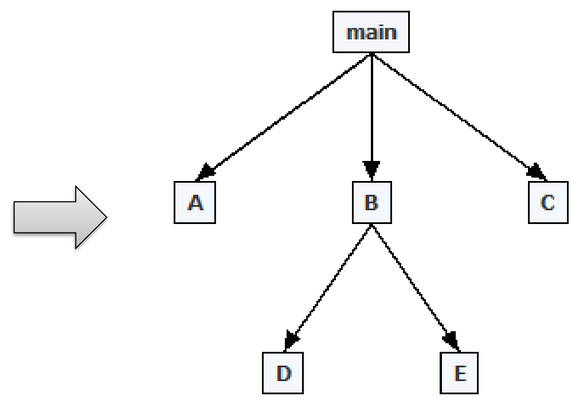


Đề xuất PP giải hệ hiệu quả

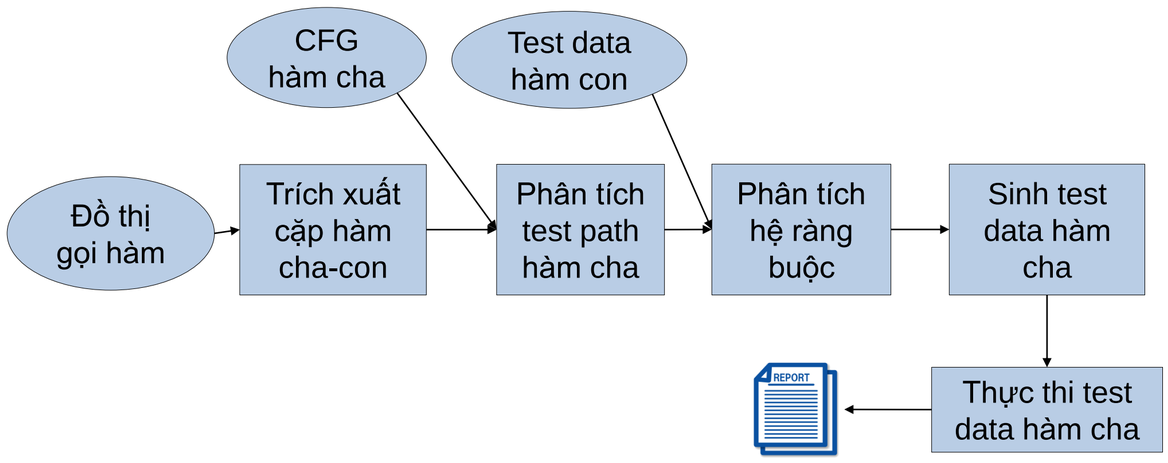


Đề xuất thuật toán sinh đồ thị gọi hàm

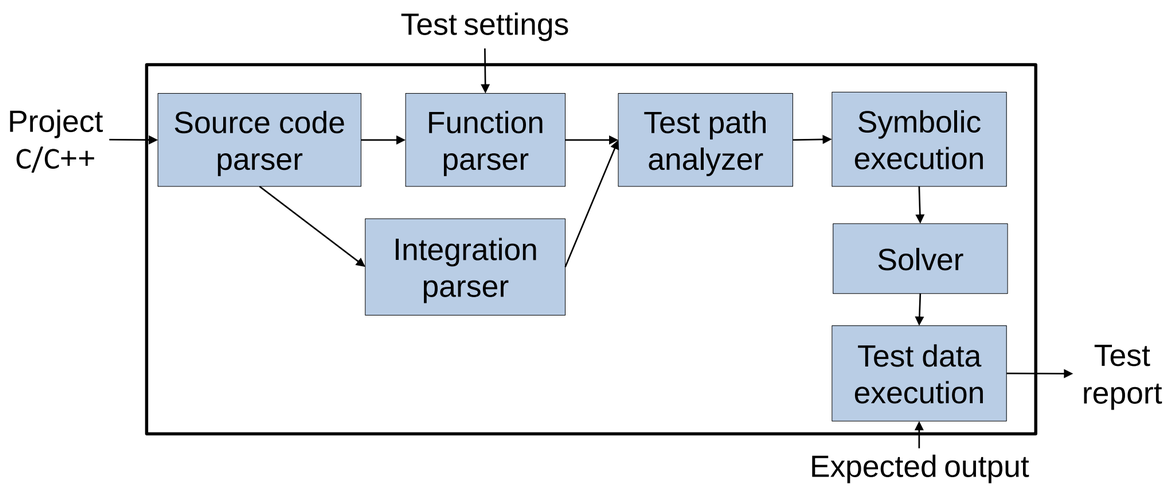
```
int main(){
    A(); B(); C();
}

int A() {}
int B() {
    D(); E();
}
float C() {}
float D() {}
int E() {}
```



Nghiên cứu PP sinh ca kiểm thử tích hợp

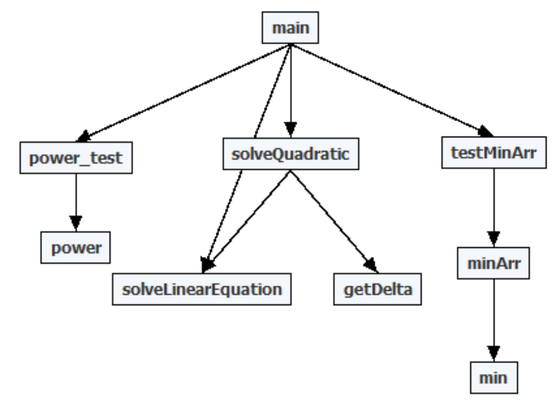


Kiến trúc công cụ



Thực nghiệm và đánh giá

Hàm C	CFT4CUnit		PathCrawler	
	Số lỗi	Số đường kiểm thử	Số lỗi	Số đường kiểm thử
Foo	3	5	-	-
Grade	1	6	1	12
Average	4	5	0	21
ComplexIndex	1	3	-	-



ID	Cặp hàm tích hợp	Input	Ouput
1	minArr - min	n = 2, a = {-34, -15}	-34
2	testMinArr - minArr	a = {-50}, n = 1, b = -50	1
3	solveQuadratic - solveLinearEquation	a = 0, b = 10, c = 47	-4
4	solveQuadratic - getDelta	a = -42, b = -38, c = 17	-51
5	power_test - power	base = -18, power = -11. result = 1	1

- Tổng thời gian sinh tập ca kiểm thử được cải thiện
- Thời gian giải một hệ ràng buộc nhanh hơn và hiệu quả hơn do kết hợp kĩ thuật sinh ngẫu nhiên và tận dụng thế mạnh SMT-Solver
- Số lượng ca kiểm thử ít dễ dàng cho quản lý
- Công cụ có ý nghĩa trong môi trường kiểm thử phần mềm và trong môi trường giảng dạy. Công cụ đã công bố mã nguồn cho các nhóm sinh viên khác phát triển tiếp

Tham khảo
1. Nicky Williams, Bruno Marre, Patricia Mouy, Muriel. Roger: ”PathCrawler: automatic generation of path tests by combining static and dynamic analysis”. In Proc. 5th European Dependable Computing Conference (EDCC-5)
2. Manish Mishra, Shashi Mishra, Rabins Porwal:”Basic Principle for testcase Generation Automatically”, VSRD-IJCSIT, Vol. 2 (9), 2012, 772-781
3. Sangeeta Tanwer and Dr. Dharmender Kumar :“Automatic testcase Generation of C Program Using CFG”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 8, July 2010
4. Zheng Wang :”Test Data Generation for Derived Types in C Program”, 2009 Third IEEE International Symposium on Theoretical Aspects of Software Engineering
5. G.C. Necula, S. McPeak, S.P . Rahul and W. Weimer, CIL: “Intermediate Language and Tools for Analysis and Transformation of C Programs”, Proc. Conference on Compiler Construction, 2002
6. Arthur H. Watson, Thomas J. McCabe: “Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric”, NIST Special Publication 500-235