**2016measuring**

**Tiêu đề:**

**Measuring code behavioral similarity for programming**

**and software engineering education**

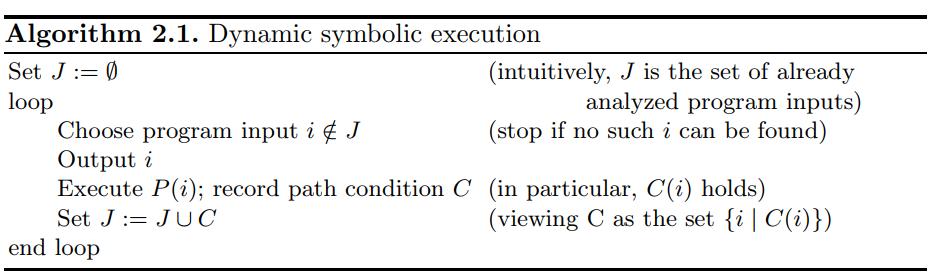
**Nội dung:**

- Xác định độ tương đồng hành vi của các chương trình

**\* Để xác định chính xác độ tương đồng:**

+ Tính tỷ lệ đầu vào của chương trình và tỷ lệ đầu ra của cả hai chương trình trên cùng một miền giá trị đầu vào.

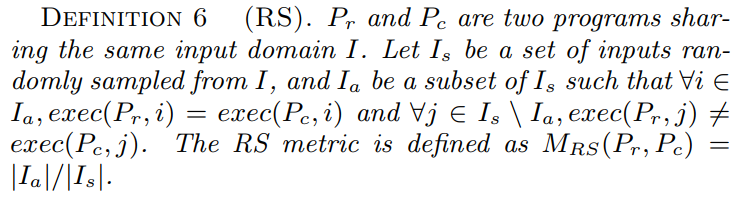
+ Dynamic symbolic execucement (DSE) (IntelliTest trong Visual Studio 2015) là một kỹ thuật thu thập các ràng buộc từ các nhánh của chương trình và phủ nhận một phần các ràng buộc để tạo ra dữ liệu đầu vào của chương trình và có độ phủ cao



+ Ba kỹ thuật đo lường độ tương đồng hành vi:

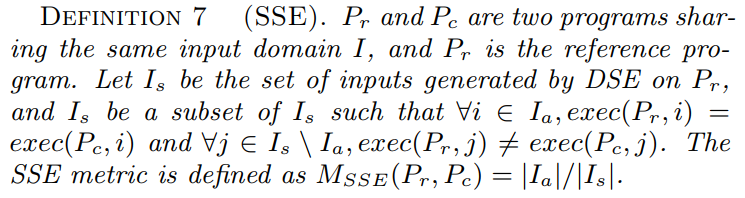
**a. Random Sampling (RS):** Sử dụng phép thử nghiệm ngẫu nhiên để tạo giá trị đầu vào trên cả 2 chương trình. Sau đó chạy 2 chương trình riêng biệt với giá trị đầu vào và so sánh kết quả đầu ra của cả 2 chương trình. Tỷ lệ trùng khớp kết quả đầu ra của 2 chương trình là kết quả của phép thử RS.

Các kết quả đánh giá cho thấy, nói chung, RS cung cấp sự xấp xỉ chính xác cao đối với sự tương đồng hành vi, nhưng nó có thể không phân biệt được những khác biệt về hành vi nhỏ



**b. Single program Symbol Execution (SSE):** Sử dụng một chương trình làm chương trình tham chiếu với chương trình cần xác định và sử dụng DSE để tạo các đầu vào kiểm tra để nắm bắt các hành vi của chương trình tham chiếu. Sau đó chạy chương trình mẫu theo phân tích trên các đầu vào thử nghiệm và tính toán tỷ lệ đầu vào đối với các đầu vào được tạo ra là giá trị SSE.

- Chương trình tham chiếu không hoàn toàn chính xác như chương trình được phân tích và đầu vào thử nghiệm có thể không không thực hiện một số hành vi.

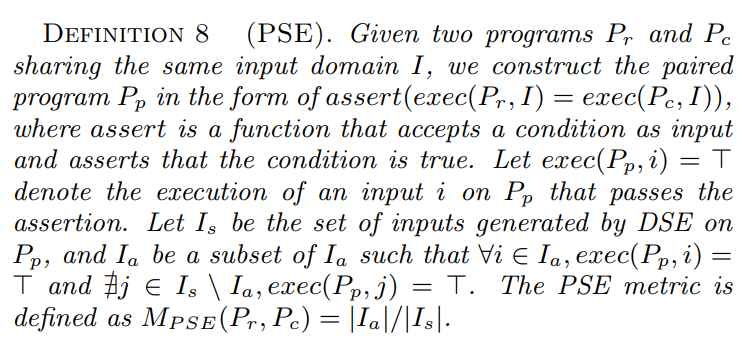


[25] K. Taneja and T. Xie. DiffGen: Automated regression unit-test generation. In *Proc. ASE*, pages 407-410, 2008.

**c. Paired program Symbolic Execution (PSE):** chỉ ra hạn chế của SSE nên kết hợp 2 kỹ thuật RS và SSE.

- Xây dựng 1 chương trình kết hợp cả hai chương trình cùng chạy trên 1 miền giá trị đầu vào và khẳng định đầu ra là như nhau. Sau đó chuyển qua DSE tạo ra các giá trị đầu vào kiểm tra trên chương trình kết hợp và tính tỷ lệ đầu vào kiểm tra, vượt qua khẳng định trên các đầu vào được tạo ra chính là giá trị PSE.

- PSE có hiệu quả cao trong việc phát hiện sự khác biệt hành vi nhỏ như vậy nhưng không cung cấp như là xấp xỉ tốt như RS không



[29] T. Wang, X. Su, Y. Wang, and P. Ma. Semantic similarity-based grading of student programs. *Inf. Softw. Technol.*, 49(2):99-107, 2007.

**godefroid2005dart**

Dynamic symbolic execucement (DSE)

**Tiêu đề:**

DART: Directed Automated Random Testing

**Nội dung:**

Giới thiệu công cụ tự động kiểm tra phần mềm DART, sử dụng 3 kỹ thuật:

1. Tự động trích giao diện màn hình của chương trình, phân tích cú pháp bằng mã nguồn tĩnh

2. Tự động tạo ra trình điều khiển thử nghiệm cho giao diện và thực hiện thử nghiệm ngẫu nhiên để mô phỏng môi trường chung nhất chương trình có thể hoạt động

3. Phân tích động cách thức hoạt động của chương trình dưới sự kiểm tra ngẫu nhiên và tự động tạo ra các đầu vào ngẫu nhiên và các đầu vào mới ngẫu nhiên thể hệ thống trực tiếp thực thi theo các đường dẫn của chương trình thay thế

Trong quá trình kiểm tra, DART sẽ phát hiện các lỗi cơ bản như sự cố đứng chương trình, các điều kiện không đúng và không dừng chương trình...

**sen2005cute**

Dynamic symbolic execucement (DSE)

**Tiêu đề:**

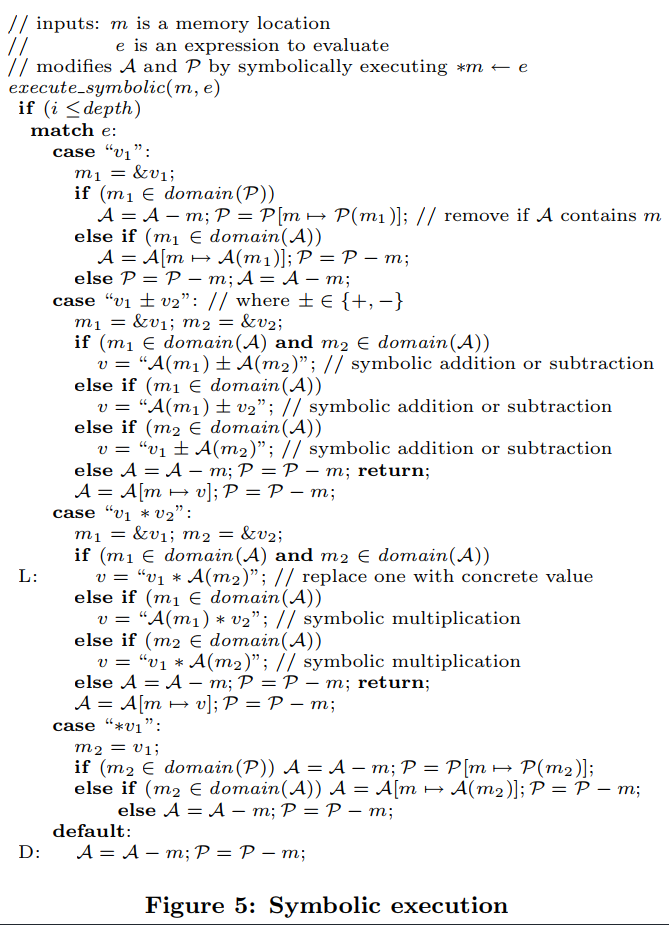
CUTE: A Concolic Unit Testing Engine for C

**Nội dung:**

Giới thiệu công cụ CUTE: tự động kiểm tra các thành phần trong chương.

Một chương trình được phân tách thành tập các hàm và được kiểm tra bằng cách tạo ra các giá trị đầu vào cho mỗi hàm. Mỗi giá trị đầu vào có thể chứa địa chỉ con trỏ, theo giõi hành vi của các hàm.

Một chương trình khi thực thi có thể biểu diễn dưới dạng cây, một nhánh chương trình được đánh dấu là một nút. Cách tiếp cận cổ điển là duyệt cây theo chiều sâu (nhưng nó khó cho những chương trình lớn, phức tạp)



**tillmann2008pex**

**Tiêu đề:**

**Pex–White Box Test Generation for .NET**

**Nôi dung:**

**su2016code**

**Tiêu đề:**

Detecting Similarly Behaving Software

(Phát hiện hành vi tương tự của phần mềm)

Nội dung:

**alur2013automated**

1. **Tiêu đề:**

Automated Grading of DFA Constructions  
(Phân loại tự động của cấu trúc DFA)

1. **Nội dung**

* Bài báo cung cấp một giải pháp phân loại tự động vấn đề tính toán và lý thuyết để hỏi học sinh cấu trúc lý thuyết hữu hạn tự động từ ngôn ngữ đó. Tập trung vào việc chỉ định khu vực có câu trả lời bị lỗi (không chính xác).
* Mỗi câu trả lời của học sinh được so sánh với kỹ thuật DFA sử dụng 3 kỹ thuật lập để bắt các lớp lỗi khác nhau.
  + 1. Bắt lỗi cú pháp
  + 2. Xem xét sự đối xứng giữa các ngôn ngữ
  + 3. Năm bắt việc bắt lỗi đọc mô tả vấn đề
* Sử dụng thuật toán và chuyển đổi ngôn ngữ mô tả MOSEL vào DFA và ngược lại

**binkley1992using**

**Tiêu đề:**

Using Semantic Differencing to Reduce the Cost of Regression Testing

(Sử dụng những kịch bản khác nhau để giảm chi phí kiểm tra hồi quy)

**Nội dung:**

Bài báo trình bày thuật toán làm giảm chi phí kiểm tra quy hồi bằng việc giảm số lượng các trường hợp kiểm thử phải chạy lại bằng cách giảm kích thước của chương trình chạy

