HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN HỌC

An toàn hệ điều hành

Đánh giá an toàn

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: An toàn thông tin - Khoa CNTT1

Nội dung

- Các đặc trưng của đặc tả an toàn
- Các kỹ thuật kiểm chứng đặc tả an toàn
- Các phương pháp phân rã dữ liệu và chương trình
- Các kỹ thuật kiểm chứng mã chương trình

Các đặc trưng của đặc tả an toàn

Các đặc trưng của đặc tả an toàn

- Các đặc tả an toàn chỉ hữu ích cho hệ thống cần phải đảm bảo mức độ an toàn/an ninh cao nhất còn mô hình an toàn có khả năng ứng dụng rộng rãi hơn
- Mục đích của đặc tả an toàn là diễn tả các hành vi chức năng của hệ thống theo cách thức chính xác, không lập lờ và phù hợp với việc xử lý của máy tính
  - Yêu cầu phù hợp với xử lý máy tính là để giảm thiểu lỗi do con người gây ra và giúp cho việc phân tích phần cứng hay phần mềm xây dựng có thỏa mãn các đặc tả hay không

Các đặc trưng của đặc tả an toàn

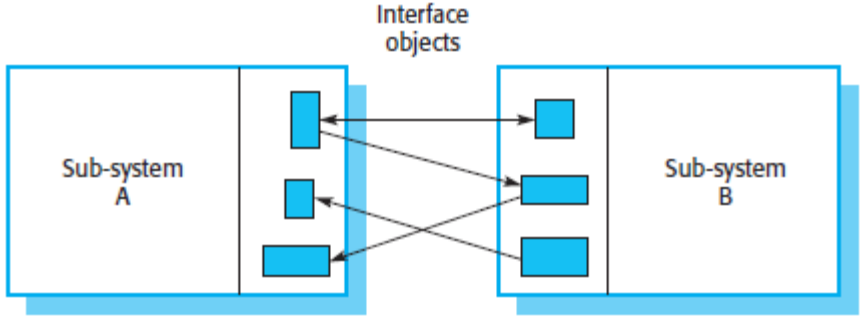
- Các đặc tả chính tắc có thể dùng để chứng minh các thuộc tính về thiết kế của hệ thống đặc biệt là việc phù hợp với các đặc tả của mô hình an toàn.
- Việc kiểm chứng chứng minh việc triển khai tuân thủ hay phù hợp với các đặc tả chính tắc. Việc chứng minh chính tắc đầy đủ cho hệ thống lớn thực sự là thách thức cho dù về mặt lý thuyết chứng minh đã được nghiên cứu rõ ràng.
  - Dù vậy việc kiểm chứng chính tắc một phần giúp người dùng đánh giá mức độ tương ứng giữa đặc tả và triển khai.

Các đặc trưng của đặc tả an toàn

- Các đặc tả chính tắc trông giống như chương trình máy tính thông thường với biểu thức lô-gíc và tinh toán. Tuy nhiên, các ký hiệu có ngữ nghĩa phong phú hơn ngôn ngữ máy tính cho phép biểu diễn các phép toán lô-gíc và quan hệ

Các đặc trưng của đặc tả an toàn

- Các đặc tả chính tắc bao gồm đặc tả giao tiếp và hành vi
  - Đặc tả giao tiếp: giúp cho việc phân rã các hệ thống lớn thành các hệ thống con. Các giao tiếp thường được mô tả bằng tập các đối tượng hay thành phần cho biết dữ liệu và các thao tác được truy nhập thông qua giao tiếp



Các đặc trưng của đặc tả an toàn

- Đặc tả hành vi: mô tả các trạng thái có thể của hệ thống và các thao tác làm thay đổi trạng thái. Nói cách khác các hành vi của hệ thống có thể được bằng cách xây dựng cách thức các hành vi này làm thay đổi trạng thái của hệ thống như thế nào.

Các kỹ thuật kiểm chứng đặc tả an toàn

- Chứng minh các đặc tả rất phức tạp và có thể có lỗi do vậy cần các công cụ tự động. Các công cụ này được gọi là công cụ chứng minh định lý (*theorem prover*)
  - Các hệ thống chứng minh và tích hợp đặc tả sinh ra một cách tự động các định lý dựa trên các tiên đề, hàm, bất biến, các hạn chế và các thành phần khác của đặc tả
  - Cần có sự trợ giúp từ phía người dùng trong việc sinh ra các tiên đề, ràng buộc hay bất biến

Các kỹ thuật kiểm chứng đặc tả an toàn

Các kỹ thuật kiểm chứng đặc tả an toàn

- Kiểm chứng dựa trên mô hình (model based) bằng cách mô tả các hành vi hệ thống có thể theo cách thức chính xác và rõ ràng về mặt toán học. Các mô hình hệ thống được kiểm nghiệm tất cả các trạng thái bằng thuật toán.
  - Việc này cho phép phát hiện sớm các lỗi như thiếu đầy đủ, mơ hồ, không nhất quán trong giai đoạn phân tích thiết kế
  - Kỹ thuật này là cơ sở kỹ thuật từ kiểm nghiệm toàn bộ (kiểm chứng mô hình – model checking) hay kiểm nghiệm các tình huống giới hạn (mô phỏng) hay kiểm nghiệm thực tế (test)

Các kỹ thuật kiểm chứng đặc tả an toàn

- Kiểm chứng mô hình là kỹ thuật mà xem xét tất cả các trạng thái hệ thống có thể theo kiểu vét cạn. Như vậy có thể chứng minh được mô hình hệ thống cho trước thực sự thỏa mãn một thuộc tính nhất định.

Các công cụ

- Spin : được dùng để lập mô hình phần mềm song song hay tiến trình dị bộ. Spin chủ yếu nhắm đến kiểm chứng chính tắc các thuật toán máy tính .
- Uppaal : được dùng để lập mô hình hệ thống thời gian thực. Uppaal sử dụng ngôn ngữ riêng cho việc mô tả các mô hình cũng như các thuộc tính.
- SMV, NuSMV : được dùng để lập mô hình phần cứng (lô-gíc số) tuy nhiên cũng có thể dùng được cho lĩnh vực khác.

Các công cụ

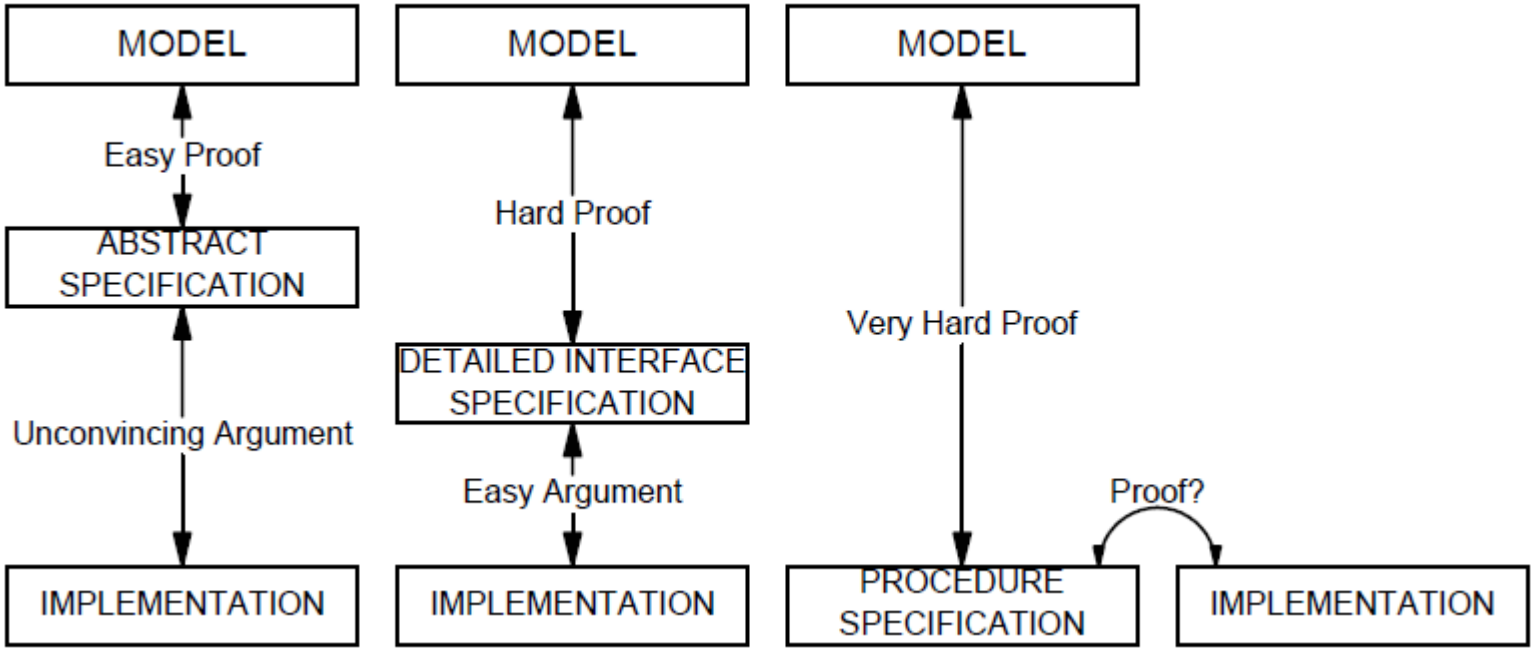
- FDR : được dùng để lập mô hình hệ thống dị bộ
- Alloy : được dùng để phân tích tính nhất quán của các cấu trúc dữ liệu. Alloy sử dụng lô-gíc bậc nhất để chuyển các đặc tả thành các biểu thức Boolean và phân tích dựa trên bộ phân tích SAT.
- Simulink Design Verifier : được dùng để kiểm chứng mô hình được sinh ra từ Simulink, một công cụ mô phỏng dựa trên luồng dữ liệu và máy trạng thái.

Các phương pháp phân rã dữ liệu và chương trình

Phân rã

- Công việc đặc tả cung cấp thông tin ở hai mức
  - Mức trừu tượng gần giống với việc lập mô hình
  - Mức chi tiết mô tả các thao tác hay hoạt động của hệ thống như mô tả các giao tiếp
- Việc chứng minh việc triển hai phù hợp mô hình đề ra có thể vô cùng khó khăn tuy nhiên việc chứng minh mã chương trình phù hợp với đặc tả có thể được chấp nhận như việc chứng minh một phần.

Chứng minh tương ứng giữa triển khai và mô tả



Các kỹ thuật phân rã

- Phân rã cấu trúc dữ liệu (Data structure refinement)
- Phân rã thuật toán (Algorithm refinement)

Phân rã cấu trúc dữ liệu

- Kỹ thuật phân rã (tinh chỉnh) dữ liệu sử dụng nhiều mức trừu tượng với mức độ chi tiết khác nhau
  - Mỗi lớp đặc tả là máy trạng thái mô tả hoàn chỉnh hệ thống
  - Lớp trên cùng trừu tượng nhất và kết hợp nhiều kiểu dữ liệu, biến và các hàm vào trong một vài hàm đơn giản
  - Các lớp kế tiếp bổ sung các chi tiết bằng các phân rã các hàm khái quát thành các đối tượng và hàm cụ thể.
- Kỹ thuật này không cho biết cách thức thiết kế hệ thống bên trong. Việc kiểm chứng cần sử dụng các kỹ thuật công nghệ phần mềm truyền thống như kiểm tra mã nguồn và kiểm thử.

Phân rã thuật toán

- Kỹ thuật phân rã thuật toán cho phép mô tả một phần cấu trúc nội tại của hệ thống. Kỹ thuật này coi hệ thống như một chuỗi các máy trạng thái có phân lớp.
  - Mỗi máy trạng thái sử dụng các chức năng do lớp trên cung cấp
  - Việc triển khai các chức năng (function) bao gồm một chương trình khái quát sử dụng các chức năng có trong máy trạng thái ở bên dưới.
  - Lớp thấp nhất cung cấp các chức năng nguyên thủy nhất cả hệ thống mà không thể phân rã thêm được nữa

Phân rã thuật toán

Layer	Formal Specifications	Abstract Programs
	interface to system ↓	
N	top-level machine (interface specification) func A func B	**proc** A_N **call** A_{N-1} **call** C_{N-1} **return** **proc** B_N **call** B_{N-1} **call** A_{N-1} **return**
$N-1$	intermediate machine func A func B func C	**proc** A_{N-1} **call** A_{N-2} **call** C_{N-2} **return** **proc** B_{N-1} **call** B_{N-2} **call** A_{N-2} **call** A_{N-2} **return** **proc** C_{N-1} **call** C_{N-2} **return**
$N-2$	intermediate machine	**proc** A_{N-1} **proc** B_{N-1} **proc** C_{N-1}
. . .	. . .	. . . .
1	intermediate machine	**proc** A_1 **proc** B_1
0	primitive machine	**proc** A_0 **proc** B_0

Phân rã thuật toán

- Việc chứng minh đặc tả sử dụng kỹ thuật này trước hết cần chứng minh các đặc tả mức cao nhất tương ứng với mô hình xây dựng.
- Tiếp theo chứng minh chương trình khái quát của lớp cao nhất phù hợp với đặc tả của nó khi biết các đặc tả các chức năng của lớp kế tiếp
- Nhược điểm chính của kỹ thuật này là việc khó thực hiện các chứng minh thuật toán khái quát. Một số bài toán chứng minh thuộc lớp bài toán khó (intractable)

Ví dụ

Abstract Machine	Data Structures	Functions
Machine 2	Files Directories	Create/delete files/directories Read/write files Access control functions
Machine 1	Files File descriptors	Create/delete files Read/write files
Machine 0	Disk blocks	Read/write disk blocks