

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGÔ THỊ LAN HOA

**SINH CA KIỂM THỬ THEO CẤP CHO
MÔ HÌNH ĐẶC TRƯNG CÓ RÀNG BUỘC SỐ**

CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH

MÃ SỐ: 60.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN ĐÌNH HÓA

HÀ NỘI - 2016

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. Nguyễn Đình Hóa

Phản biện 1:

Phản biện 2:

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại
Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ... giờ ngày tháng năm

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

MỞ ĐẦU

Tính cấp thiết của đề tài

Tổng quan về vấn đề nghiên cứu

Mục đích, đối tượng, phạm vi và phương pháp nghiên cứu

Cấu trúc luận văn

Nội dung của luận văn được trình bày trong ba phần chính như sau:

1. Phần mở đầu

2. Phần nội dung: bao gồm ba chương

Chương 1: Tổng quan về Khai phá dữ liệu trong bài toán dự báo

Chương 2: Khai phá dữ liệu và các công cụ học máy

Chương 3: Dự báo tiến trình học tập của sinh viên Đại học Thủy lợi

3. Phần kết luận

CHƯƠNG I. SINH CA KIỂM THỬ DỰA TRÊN CÂY ĐẶC TRƯNG

1.1 Sinh ca kiểm thử tự động

1.1.1 Khái niệm ca kiểm thử

1.2 Sinh ca kiểm thử hộp đen

1.2.1 Một số kỹ thuật tiêu biểu

Một số kỹ thuật kiểm thử hộp đen tiêu biểu:

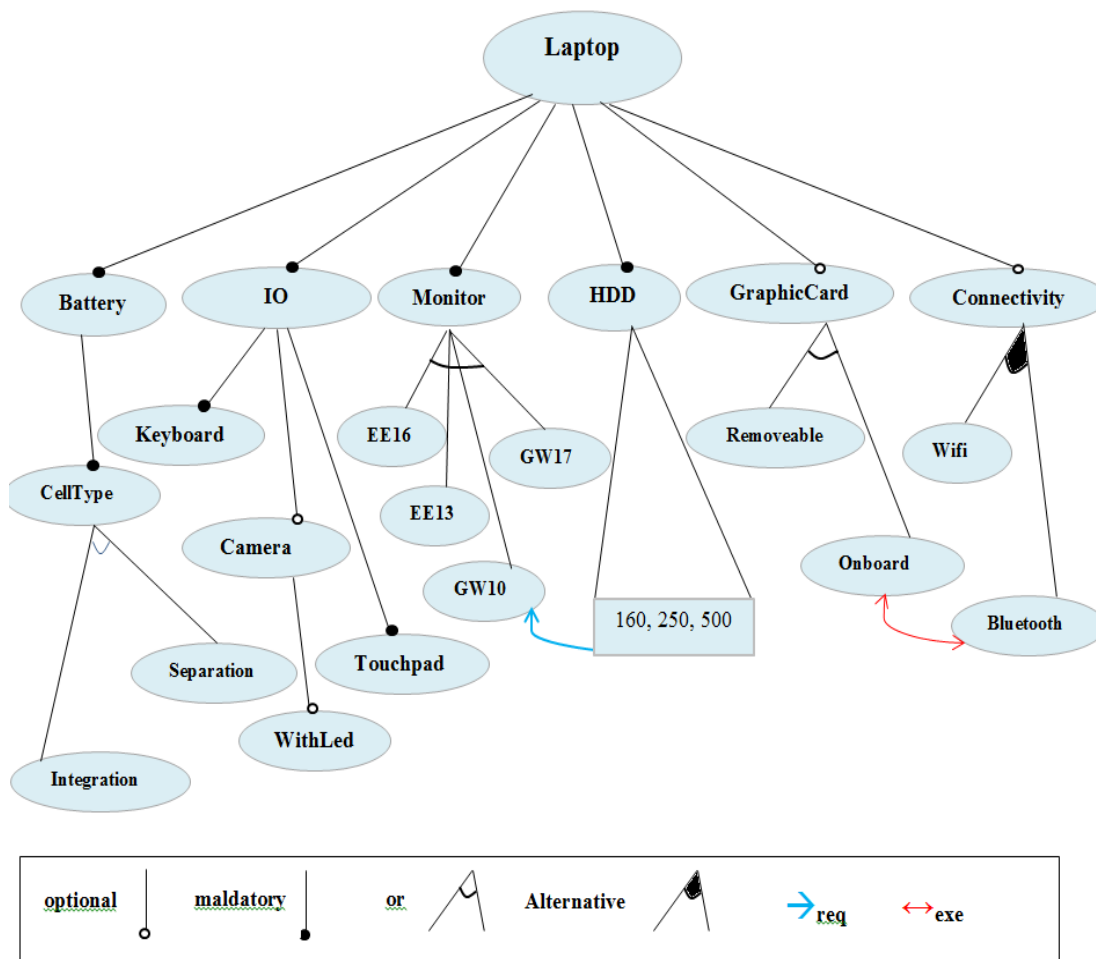
- *Phân lớp tương đương*
- *Phân tích giá trị biên*
- *Kiểm thử theo cặp*

1.2.2 Kỹ thuật kiểm thử theo cặp

Rõ ràng, số ca kiểm thử được sinh ra bởi kỹ thuật kiểm thử theo cặp ít hơn rất nhiều so với kiểm thử tất cả các trường hợp. Như vậy, kiểm thử theo cặp là kỹ thuật có thể giúp tiết kiệm được thời gian và công sức mà vẫn đảm bảo được chất lượng phần mềm.

1.3 Cây đặc trưng

Dòng sản phẩm phần mềm thường được mô hình hóa bằng cây đặc trưng. Hình 1.3 là một ví dụ về cây đặc trưng được lấy từ trang [<http://www.splot-research.org>] về dòng máy tính của hãng Dell, học viên đã lược bỏ bớt các đặc trưng cho đơn giản và phù hợp với ví dụ của luận văn.



Hình I-1 Cây đặc trưng cho dòng sản phẩm phần mềm Laptop

Cây đặc trưng được biểu diễn dưới dạng phân cấp các đặc trưng và được tạo bởi 2 yếu tố:

Mối quan hệ: mô tả mối quan hệ giữa các đặc trưng của cây.

Kiểm ràng buộc: mô tả các kiểm ràng buộc giữa các đặc trưng.

Có 4 loại đặc trưng là:

- 1, Đặc trưng **Mandatory (Bắt buộc)**
- 2, Đặc trưng **Optional (Tùy chọn)**
- 3, Nhóm đặc trưng **Alternative (Thay thế)**
- 4, Nhóm đặc trưng **Or (Hoặc)**

Có 2 loại ràng buộc là:

- 1, Ràng buộc **Requires (Yêu cầu)**
- 2, Ràng buộc **Excludes (Loại trừ)**

1.4 Cây đặc trưng trong kiểm thử phần mềm

Trong [4], tác giả đã đưa ra một phương pháp dựa trên kỹ thuật FOT (Feature Oriented Testing). Phương pháp này nhằm hỗ trợ người kiểm thử thiết kế các mô hình kiểm thử hiệu quả hơn.

a, Khái niệm

b, Sinh ca kiểm thử theo cặp cho cây đặc trưng

c, Đánh giá

1.5 Kết luận chương

Chương này đã nêu lên được các khái niệm về kiểm thử cũng như các phương pháp sinh ca kiểm thử nói chung và kiểm thử hộp đen nói riêng. Đồng thời, đưa ra khái niệm về phương pháp sinh ca kiểm thử theo cặp nhằm giảm bớt số lượng các ca kiểm thử mà vẫn đảm bảo chất lượng phần mềm. Cuối cùng, chương này đưa đến khái niệm về cây đặc trưng, phương pháp biểu diễn cây đặc trưng và phương thức ứng dụng cây đặc trưng vào kiểm thử.

CHƯƠNG II. CÂY ĐẶC TRƯNG CÓ RÀNG BUỘC SỐ VÀ SINH CA KIỂM THỬ THEO CẶP CHO CÂY ĐẶC TRƯNG MỞ RỘNG

2.1 Cây đặc trưng có ràng buộc số

2.1.1 Hạn chế của cây đặc trưng

Cây đặc trưng chỉ áp dụng cho các ràng buộc logic. Khi cần xử lý các giá trị số, FTT với đặc trưng số ở mục trước có một số khó khăn:

- Cần tính bằng tay để giải các ràng buộc số để tìm ra giá trị cụ thể.
- Một số đầu vào ca kiểm thử FTT là không khả thi vì CTC (cross-tree-constraint) hiện tại không thể biểu diễn được đầy đủ các ràng buộc, đặc biệt là ràng buộc số.

2.1.2 Mở rộng cây đặc trưng với ràng buộc số

Để giải quyết 2 vấn đề nêu trên, đồng thời khắc phục được nhược điểm như luận văn đã trình bày ở phần 2.4, [9] đã đề xuất một cây đặc trưng mới cho phép biểu diễn các đặc trưng và các ràng buộc số. Nhờ vậy, ta có thể biểu diễn cả giá trị boolean (như của FTT) và các giá trị số.

2.2 Sinh ca kiểm thử theo cặp cho cây đặc trưng có ràng buộc số

2.2.1 Hướng tiếp cận

Cụ thể để sinh ca kiểm thử cho mô hình đặc trưng ta thực hiện theo các bước sau:

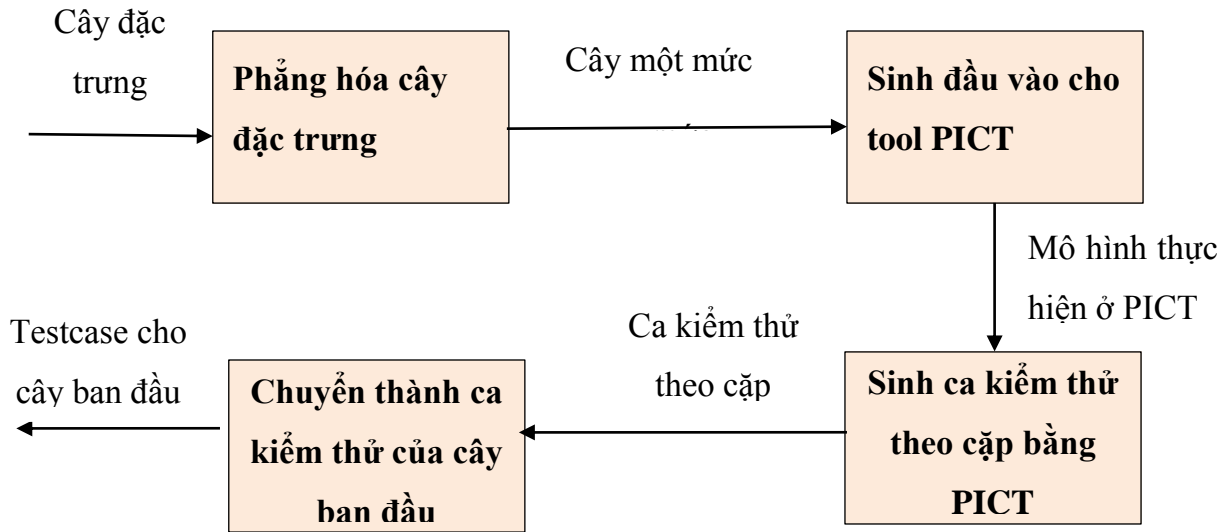
Bước 1. Phẳng hóa cây mô hình đặc trưng sử dụng thuật toán phẳng hóa cây.

Bước 2. Từ mô hình đặc trưng đã phẳng hóa sinh ra đầu vào cho tool PICT (Pairwise Independent Combinatorial Testing tool).

Bước 3. Dùng tool PICT để sinh ra ca kiểm thử.

Bước 4. Chuyển ca kiểm thử được sinh bằng tool PICT thành ca kiểm thử thực tế.

Sơ đồ sau biểu diễn quá trình sinh ca kiểm thử cho mô hình đặc trưng:



Hình II-1 Quá trình sinh ca kiểm thử cho mô hình đặc trưng

2.2.2 Phẳng hóa cây đặc trưng có ràng buộc số

a, Thuật toán phẳng hóa

Thuật toán phẳng hóa được thực hiện theo 2 bước sau:

Bước 1: Mọi đặc trưng với các quan hệ và ràng buộc liên quan đều được kéo lên tới khi nó được đặt trực tiếp bên dưới nút gốc. Mọi đặc trưng sau đó được coi như những tham số.

Bước 2: Sau đó, thuật toán gán cho mọi tham số những giá trị tương ứng của nó.

b, Các luật phẳng hóa

Cây đặc trưng của chúng ta gồm 4 loại quan hệ. Đối với mỗi sự kết hợp của các quan hệ cha con thì một luật chuyển đổi được yêu cầu. Các luật này được áp dụng lặp đi lặp lại cho tới khi ta thu được cây đặc trưng 1 mức. Vì vậy cần $4 \times 4 = 16$ luật.

I. Kí hiệu:

|+| p : p là Mandatory

|-| p : p là Optional

|*| p₁|*| p₂|*| ...|*| p_n : (p₁, ..., p_n) là Alternative group, gọi p_i là alternative

|/| p₁|/| p₂|/| ...|/| p_n : (p₁, ..., p_n) là Or group, gọi p_i là or

Ghi chú:

Coi biến boolean có giá trị {0,1}: { 0 = true, 1 = false}

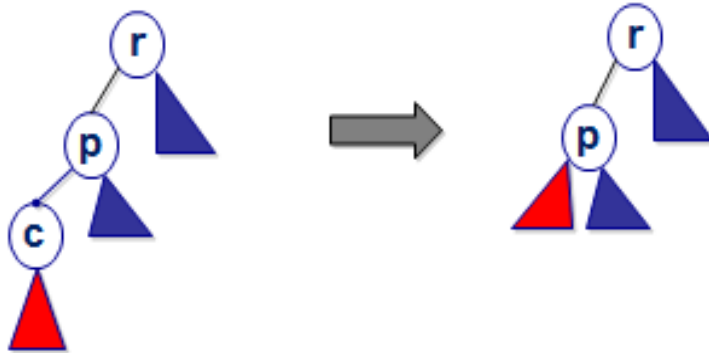
$-\infty$ là số interger nhỏ nhất biểu diễn được.

Gọi R là nút gốc, P là con của R và C là con của P.

II. Xét các trường hợp:

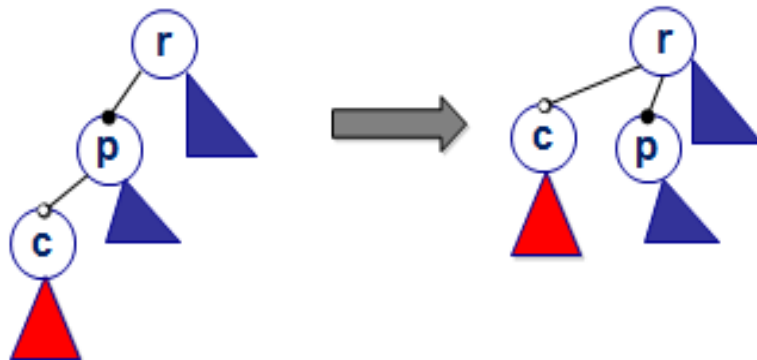
1. *C là Mandatory, P có kiểu bất kỳ (Mandatory, Optional, Alternative, Or)*

- Xoá C khỏi danh sách con của P
- Thêm danh sách con của C vào P
- Chuyển C thành P trong các biểu thức ràng buộc
- $\text{Value}(P \text{ mới}) = \text{value}(P \text{ cũ}) \cup \text{value}(C)$
- Xoá C khỏi danh sách nút của cây



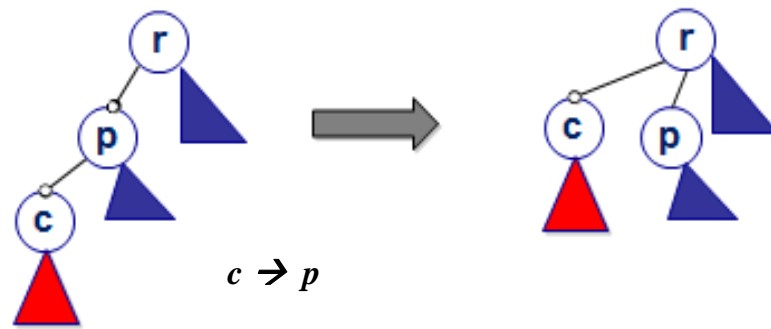
2. *C là optional, P là Mandatory*

- Chuyển C thành con của R



3. *C là optional, P là Optional*

- Chuyển C thành con của R
- Thêm ràng buộc: $C \rightarrow P$



4. C là optional, P là alternative

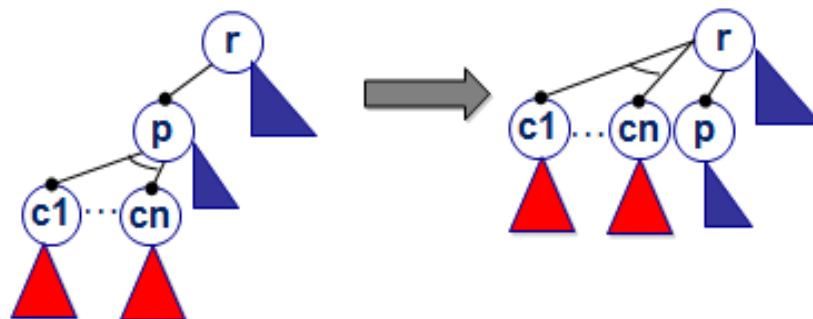
- Làm giống 3

5. C là optional, P là or

- Làm giống 3

6. $(C1, \dots, Cn)$ là Alternative group, P là Mandatory

- Chuyển cả nhóm $(C1, \dots, Cn)$ thành con của R



7. $(C1, \dots, Cn)$ là Alternative group, P là Optional

- Thêm nút con NotC vào Alternative group $(C1, \dots, Cn, \text{NotC})$

- Chuyển cả nhóm $(C1, \dots, Cn, \text{NotC})$ thành con của R

- Thêm ràng buộc:

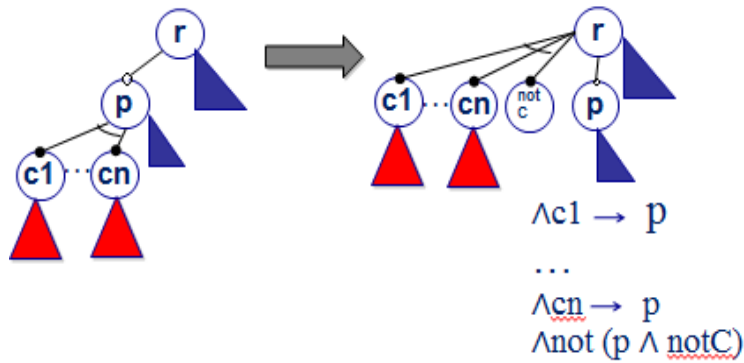
$C1 \rightarrow P;$

$C2 \rightarrow P;$

...

$Cn \rightarrow P;$

not (P and notC);



8. $(C1, \dots, Cn)$ là Alternative group, P là alternative

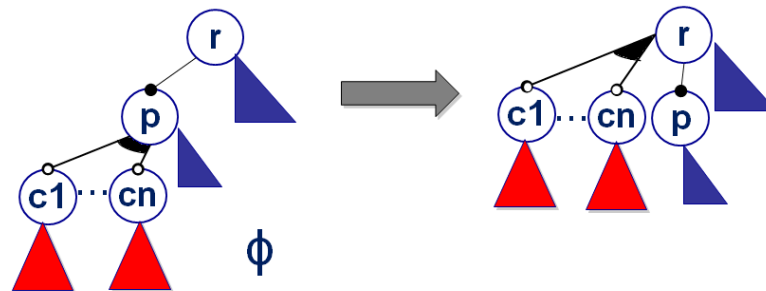
- Làm giống 7

9. $(C1, \dots, Cn)$ là Alternative group, P là Or

- Làm giống 7

10. $(C1, \dots, Cn)$ là Or group, P là Mandatory

- Chuyển cả nhóm $(C1, \dots, Cn)$ thành con của R



11. $(C1, \dots, Cn)$ là Or group, P là Optional

- Thêm nút con NotC vào Alternative group $(C1, \dots, Cn, \text{NotC})$

- Chuyển cả nhóm $(C1, \dots, Cn, \text{NotC})$ thành con của R

- Thêm ràng buộc:

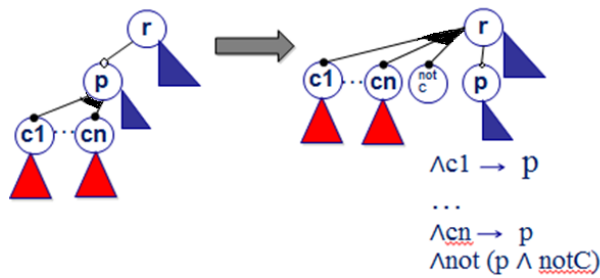
$$C1 \rightarrow P;$$

$$C2 \rightarrow P;$$

...

$$Cn \rightarrow P;$$

$$\text{not } (P \text{ and not } C);$$



12. (C1,...,Cn) là Or group, P là alternative

- Làm giống 7

13. (C1,...,Cn) là Or group, P là Or

- Làm giống 7

III. Sau khi kết thúc, thu được cây chỉ có gốc R và con pi. Xuất ra file output theo định dạng:

1. P là mandatory:

Gọi <giá trị i> là giá trị thuộc value (P), ghi ra file :

P: <giá trị 1>, <giá trị 2>,....<giá trị n>

2. P là optional:

Gọi <giá trị i> là giá trị thuộc value (P), ghi ra file :

P: <giá trị 1>, <giá trị 2>,....<giá trị n>, <âm vô cùng>

3. (P1,...,Pn) là alternative group:

Gọi <giá trị ij> là giá trị thuộc value (Pi), i thuộc [1,n], ghi ra file :

P: <giá trị 11>, <giá trị 12>,....<giá trị nn>

4. (P1,...,Pn) là or group:

Gọi <giá trị ij> là giá trị thuộc value (Pi), i thuộc [1,n], ghi ra file :

P1: <giá trị 11>, <giá trị 12>,....<giá trị 1m>, <âm vô cùng>

...

Pn: <giá trị n1>, <giá trị n2>,....<giá trị nm>, <âm vô cùng>

Bổ xung thêm ràng buộc:

NOT (([P1] = <âm vô cùng>) AND ... AND ([Pn] = <âm vô cùng>))

Ngoài ra, trong các ràng buộc, cần thay đổi như sau:

- Nếu P là kiểu boolean:

Thay P bằng $([P] = 0)$ trong các ràng buộc logic

Thay P bằng $[P]$ trong các ràng buộc so sánh, số

- Nếu P là kiểu integer mà không cần bỏ xung <âm vô cùng>:

Thay P bằng $([P] \text{ in } \{<\text{giá trị } 1>, \dots, <\text{giá trị } n>\})$ trong các ràng buộc logic

Thay P bằng $[P]$ trong các ràng buộc so sánh, số

- Nếu P là kiểu integer mà cần bỏ xung <âm vô cùng>:

Thay P bằng $([P] \text{ in } \{<\text{giá trị } 1>, \dots, <\text{giá trị } n>, <\text{âm vô cùng}>\})$ trong các ràng buộc logic

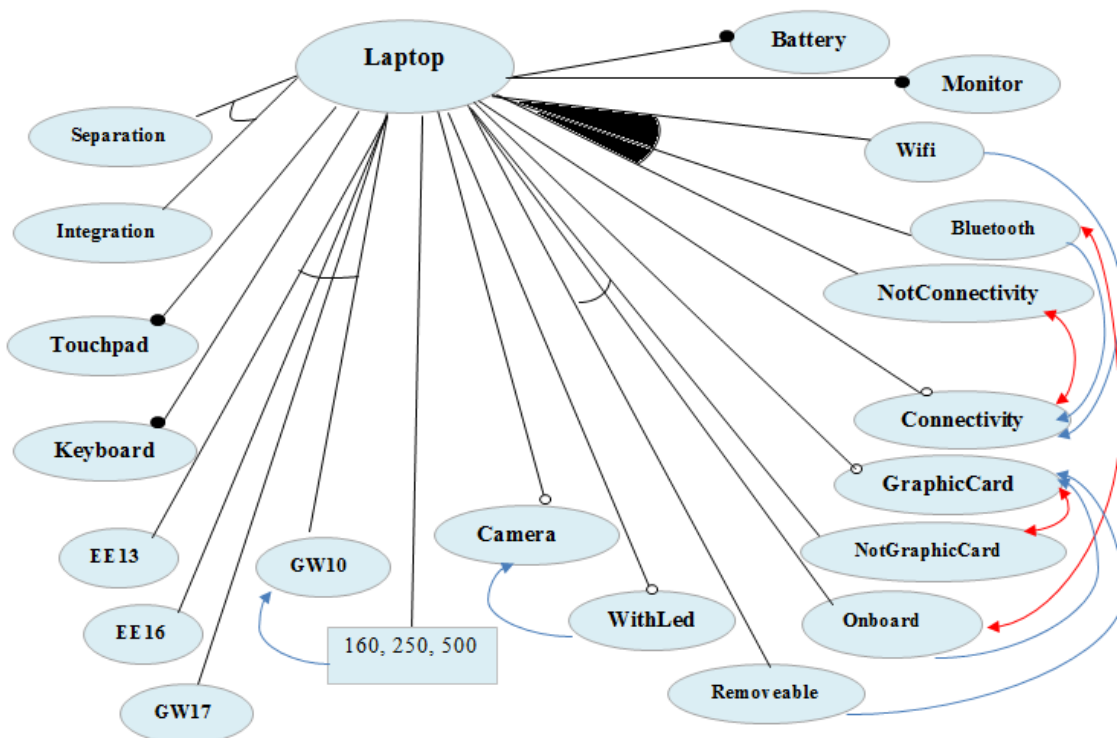
Thay P bằng $[P]$ trong các ràng buộc so sánh, số

- Nếu có ràng buộc dạng $a \rightarrow b$:

Thay bằng IF a THEN b;

Sau khi thực hiện phẳng hóa cây đặc trưng cho dòng sản phẩm phần mềm

Laptop của hãng Dell ta thu được kết quả là cây một mức như sau:



Hình II-2 Cây Laptop sau khi phẳng hóa

2.2.3 Sinh đầu vào cho tool PICT

PICT chạy như một công cụ dòng lệnh với đầu vào là file txt theo một định dạng nhất định. Vì vậy để sử dụng được công cụ này, chúng ta phải chuyển bài toán đầu vào (tức là cây đã được phẳng hóa và gán tham số) dưới định dạng đầu vào của tool PICT.

Ví dụ, file đầu vào Laptop.txt cho PICT có dạng như sau:

```
Laptop : Battery, IO, Monitor, HDD, GraphicCard, Connectivity
Battery : CellType
CellType : Integration, Separation
IO : Keyboard, Touchpad, Camera
Camera : WithLed
Monitor : EE16, EE13, GW10, GW17
HDD : 160, 250, 500
GraphicCard : Removeable, Onboard
Connectivity : Wifi, Bluetooth

IF [GraphicCard] = "Onboard" THEN [Connectivity] <>
"Bluetooth";
IF [HDD] in { 160, 250 } THEN [Monitor] = "GW10";
```

Hình II-3 File đầu vào cho PICT

2.2.4 Dùng PICT để sinh ra ca kiểm thử

Trước tiên, ta cần cài đặt PICT thông qua các bước sau:

Bước 1: Truy cập trang <http://www.pairwise.org/tools.asp>

Bước 2: Tải phiên bản mới nhất của PICT về máy

Bước 3: Chạy file cài đặt PICT

2.2.5 Chuyển ca kiểm thử được sinh bằng tool PICT thành ca kiểm thử thực tế

Việc chuyển ca kiểm thử được sinh bằng PICT thành ca kiểm thử thực tế cần lưu ý những điểm sau:

- Trong quá trình phẳng hóa cây, gán tham số và gán giá trị chúng ta đã bổ sung thêm một số nút phủ định do đó ca kiểm thử sinh ra chứa những nút này. Mà thực tế cây ban đầu không chứa những nút này vì vậy chúng ta phải loại bỏ những nút phủ định này trong những ca kiểm thử được sinh ra.
- Trong quá trình phẳng hóa cây các đặc trưng *mandatory* trong cùng một nhánh cũng được gộp lại theo luật 1 đã trình bày ở phần 2.2.2. Vì vậy chúng ta cũng cần phải tách ra như cây ban đầu.
- Thêm nữa trong quá trình gán giá trị cho cây đặc trưng đã được phẳng hóa chúng ta cũng thêm vào một số nút phụ. Vì vậy với mỗi ca kiểm thử có chứa nút phụ này ta tách thành số ca kiểm thử tương ứng với số giá trị của nó. Mỗi giá trị tương ứng với một ca kiểm thử.

2.3 Kết luận chương

Chương 2 đã trình bày một mở rộng của FTT với các đặc trưng số và ràng buộc số. Đồng thời, cũng trình bày được phương pháp sinh ca kiểm thử cho cây đặc trưng có ràng buộc số thông qua luật phẳng hóa.

CHƯƠNG III. CÀI ĐẶT VÀ THỬ NGHIỆM HỆ THỐNG PTG-FFT

3.1 Hệ thống PTG-FFT

3.1.1 Giới thiệu

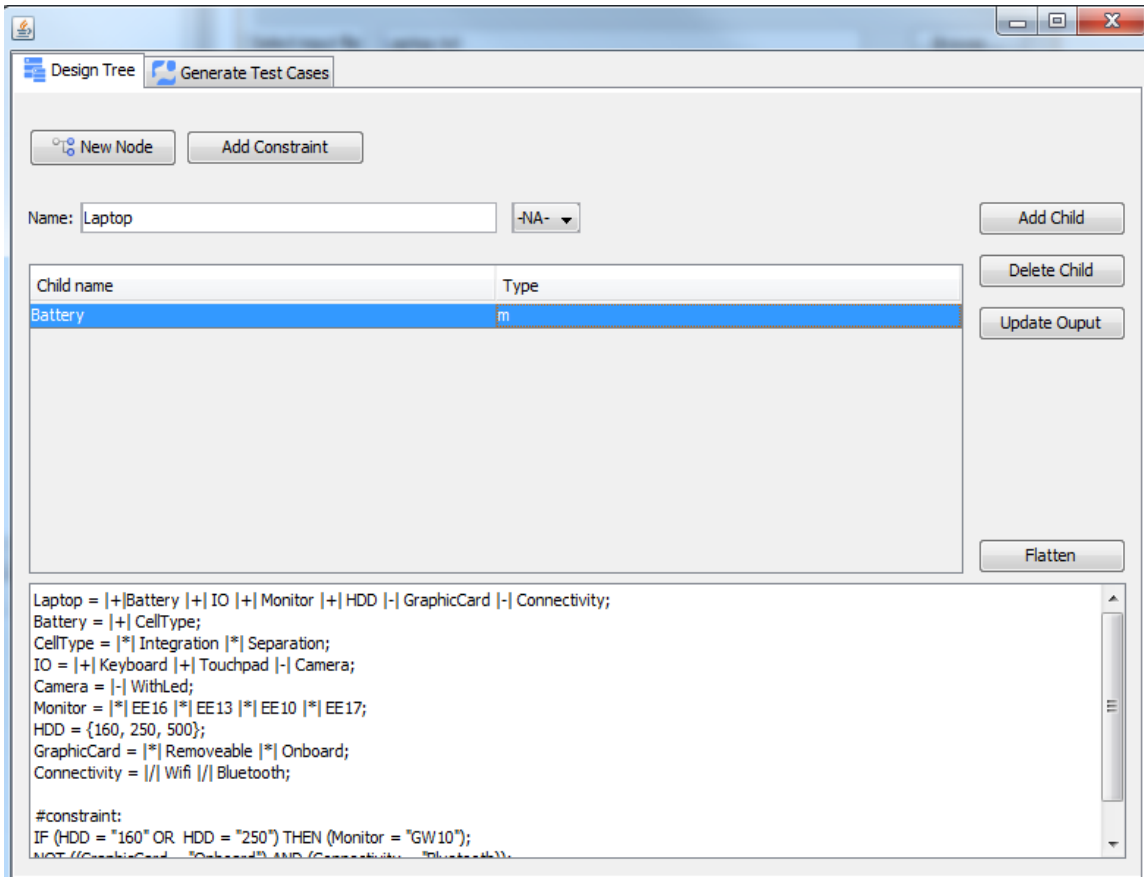
PTG-FFT là một ứng dụng được viết bằng ngôn ngữ Java trên nền tảng Netbean. Chương trình có các chức năng như sau:

1. *Thêm cây đặc trưng*: Giúp người dùng thêm vào cây đặc trưng của dòng sản phẩm phần mềm cụ thể. Chức năng này gồm hai thao tác chính như sau:
 - *Add Node*
 - *Add Constraint*
2. *Update output*: Sau khi thêm nút thành công thì lưu ra file feature.txt mới
3. *Flattening*: Phẳng hóa cây đặc trưng vừa được thêm vào, chức năng này sẽ phẳng hóa cây để tạo ra một file flattening.txt để chuyển hóa thành đầu vào cho PICT
4. *Call PICT*: chức năng chuyển hóa file flattening.txt thành file PICT.txt với định dạng mà PICT có thể đọc được, sau đó sẽ gọi tool PICT để sinh ra ca kiểm thử theo cặp cho cây đặc trưng đã được phẳng hóa
5. *Lưu kết quả ra file excel*: Chức năng này cho phép lưu kết quả chương trình ra dạng file excel.

3.1.2 Giao diện hệ thống

Giao diện hệ thống được chia thành hai phần (tương ứng với hai form trên màn hình khởi tạo):

- Phần 1: Design Tree (Thiết kế cây đặc trưng)
- Phần 2: Generate Test Cases (Sinh ca kiểm thử theo cặp cho cây)



Hình III-1 Giao diện hệ thống

3.1.3 Đặc tả dữ liệu đầu vào

Với trường hợp cây đặc trưng Hình 1.3 ta có file dữ liệu đầu vào như sau:

```
Laptop = |+|Battery |+| IO |+| Monitor |+| HDD |-| GraphicCard |-|
Connectivity;
Battery = |+| CellType;
CellType = |*| Integration |*| Separation;
IO = |+| Keyboard |+| Touchpad |-| Camera;
Camera = |-| WithLed;
Monitor = |*| EE16 |*| EE13 |*| GW10 |*| GW17;
HDD = {160, 250, 500};
GraphicCard = |*| Removeable |*| Onboard;
Connectivity = |/| Wifi |/| Bluetooth;
#constraint:
IF (HDD = "160" OR HDD = "250") THEN (Monitor = "GW10");
NOT ((GraphicCard = "Onboard") AND (Connectivity = "Bluetooth"));
```

Hình III-2 File đầu vào của chương trình cài đặt

3.2 Cài đặt chương trình

3.2.1 Thêm cây đặc trưng

- a, Thêm các đặc trưng và ràng buộc cho cây cho cây
- b, Lưu trữ cây đặc trưng

3.2.2 Phẳng hóa cây đặc trưng

3.2.3 Từ mô hình đặc trưng đã phẳng hóa sinh ra đầu vào cho tool PICT

3.2.4 Dùng tool PICT để sinh ra ca kiểm thử

3.2.5 Chuyển đầu ra của tool PICT thành ca kiểm thử thực tế

Sau khi thực hiện các quy tắc như đã trình bày ở phần 2.3.4, phần mềm sẽ thực hiện việc sinh các ca kiểm thử ra Microsoft Excel. Kết quả thu được như sau:

	A	B	C	D	E	F	G	H	I	J
1	Laptop	Battery	CellType	IO	Camera	Monitor	HDD	GraphicCard	Connectivity	
2	HDD	CellType	Integration	Keyboard	WithLed	GW10	160	Onboard	Wifi	
3	Connectivity	CellType	Separation	Touchpad	WithLed	GW17	500	Removeable	Bluetooth	
4	HDD	CellType	Integration	Camera	WithLed	GW17	500	Removeable	Wifi	
5	Monitor	CellType	Integration	Keyboard	WithLed	EE13	500	Removeable	Bluetooth	
6	IO	CellType	Separation	Touchpad	WithLed	EE16	500	Onboard	Wifi	
7	Connectivity	CellType	Separation	Camera	WithLed	GW10	250	Removeable	Bluetooth	
8	GraphicCard	CellType	Integration	Touchpad	WithLed	GW10	250	Onboard	Wifi	
9	Monitor	CellType	Separation	Camera	WithLed	GW17	500	Onboard	Wifi	
10	Battery	CellType	Separation	Keyboard	WithLed	GW17	500	Removeable	Bluetooth	
11	IO	CellType	Integration	Keyboard	WithLed	GW17	500	Removeable	Bluetooth	
12	GraphicCard	CellType	Separation	Camera	WithLed	GW17	500	Removeable	Bluetooth	
13	Battery	CellType	Integration	Camera	WithLed	EE16	500	Removeable	Bluetooth	
14	GraphicCard	CellType	Separation	Keyboard	WithLed	EE16	500	Removeable	Bluetooth	
15	Monitor	CellType	Separation	Touchpad	WithLed	GW10	160	Removeable	Bluetooth	
16	Connectivity	CellType	Integration	Keyboard	WithLed	EE16	500	Onboard	Wifi	
17	IO	CellType	Separation	Camera	WithLed	EE13	500	Onboard	Wifi	
18	Battery	CellType	Integration	Keyboard	WithLed	GW10	250	Onboard	Wifi	
19	Connectivity	CellType	Separation	Camera	WithLed	GW10	160	Onboard	Wifi	
20	HDD	CellType	Separation	Touchpad	WithLed	EE16	500	Removeable	Bluetooth	
21	Battery	CellType	Integration	Touchpad	WithLed	EE13	500	Onboard	Wifi	
22	Monitor	CellType	Separation	Camera	WithLed	GW10	250	Removeable	Bluetooth	
23	Connectivity	CellType	Integration	Camera	WithLed	EE13	500	Removeable	Bluetooth	
24	IO	CellType	Integration	Camera	WithLed	GW10	250	Removeable	Wifi	
25	HDD	CellType	Integration	Keyboard	WithLed	GW10	250	Onboard	Wifi	
26	Battery	CellType	Separation	Camera	WithLed	GW10	160	Removeable	Wifi	
27	HDD	CellType	Separation	Camera	WithLed	EE13	500	Removeable	Bluetooth	
28	GraphicCard	CellType	Separation	Keyboard	WithLed	EE13	500	Removeable	Wifi	
29	IO	CellType	Integration	Keyboard	WithLed	GW10	160	Removeable	Wifi	
30	GraphicCard	CellType	Integration	Touchpad	WithLed	GW10	160	Removeable	Wifi	
31	Monitor	CellType	Separation	Keyboard	WithLed	EE16	500	Removeable	Bluetooth	
32	IO	CellType	Integration	Touchpad	WithLed	GW10	500	Onboard	Wifi	
33										

Hình III-3 Ca kiểm thử cho cây đặc trưng ban đầu

3.3 Kết quả đạt được

3.3.1 Dữ liệu thử nghiệm

3.3.2 Kết quả thử nghiệm

3.4 Đánh giá kết quả

Với kết quả ở mục 3.3.2, ta tổng hợp được bảng kết quả sau:

Cây đặc trưng	Số đặc trưng	Số ràng buộc	Số ca kiểm thử khi chưa có ràng buộc số	Số ca kiểm thử
1	36	2	4704	64
2	34	2	480	29
3	24	1	216	15

4	15	1	97	10
5	25	2	1728	32

Bảng III-1 Kết quả thử nghiệm

Từ bảng kết quả trên ta thấy:

- Việc mở rộng ràng buộc dạng số đã làm giảm đáng kể số lượng test input cũng như thời gian thực hiện.
- Số ca kiểm thử phụ thuộc vào số ràng buộc và các quan hệ chứ không phụ thuộc vào số đặc trưng.
- Cây đặc trưng với ràng buộc số cho phép biểu diễn cây một cách linh hoạt hơn.

3.5 Kết luận chương

Chương 3 đã trình bày chương trình sinh ca kiểm thử cho cây đặc trưng có ràng buộc số. Chương trình đã đáp ứng được các chức năng như nhập cây đặc trưng, phẳng hóa cây đặc trưng, sinh ca kiểm thử theo cặp bằng PICT và chuyển về ca kiểm thử cho cây ban đầu

KẾT LUẬN

1. Kết quả đạt được

Luận văn của tác giả với đề tài “Sinh ca kiểm thử cho mô hình đặc trưng có ràng buộc số” về cơ bản đã hoàn thiện. Đề tài đã giải quyết được các vấn đề sau:

1. Trình bày các khái niệm cơ bản của kiểm thử phần mềm, các kỹ thuật kiểm thử, quy trình kiểm thử.
2. Nghiên cứu về mô hình cây đặc trưng dạng cơ bản, các lý thuyết về dòng sản phẩm phần mềm.
3. Đề xuất việc mở rộng mô hình cho cây đặc trưng cơ bản: thêm các đặc trưng số và ràng buộc số.
4. Xây dựng được chương trình thử nghiệm từ các lý thuyết nghiên cứu, và kết quả thử nghiệm ban đầu

2. Hạn chế

Bên cạnh những mặt đã đạt được, luận văn vẫn còn có một số mặt hạn chế:

1. Mới chỉ làm được với một số ràng buộc số.
2. Chương trình mới xử lý được file định dạng đầu vào là file text và lưu dưới dạng Excel chưa xử lý với các định dạng khác.
3. Do thời gian gấp rút và khả năng có hạn, luận văn còn nhiều thiếu sót, mong được hội đồng đánh giá và góp ý để tôi chỉnh sửa luận văn được tốt hơn.

3. Hướng phát triển

Với kết quả của quá trình nghiên cứu và hạn chế của chương trình, định hướng phát triển tiếp theo của luận văn :

- Xử lý với các đặc trưng với các giá trị liên tục, làm tăng khả năng biểu diễn của mô hình.
- Mở rộng FOT với các yêu cầu khác để phù hợp với các bài toán thiết kế test input như mở rộng về độ ưu tiên.
- Kết hợp với các phương pháp khác của BBT như phương pháp thử nghiệm kết hợp và kỹ thuật phân tích miền đầu vào “input-domain” ví dụ, phân vùng tương đương, giá trị biên, vv..) để giảm thiểu được các test input

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Việt

- [1] Le C.D., “Sinh input tự động cho tool PICT”, Luận văn tốt nghiệp thạc sỹ khoa học, 2013
- [2] Nguyen T.N., “Sinh ca kiểm thử theo cặp cho mô hình đặc trưng”, Luận văn tốt nghiệp thạc sỹ khoa học, 2013

Tài liệu Tiếng Anh

- [3] Ahmet, S. K., Halit, O., and Ali, D.-Mapping extended feature models to constraint logic programming over finite domains, In Proceedings of the 14th international conference on Software product lines, Springer-Verlag, Berlin, Heidelberg, 2010, pp 286-209
- [4] Alvarez D., Carlos E.-Automated Reasoning on Feature Models via Constraint Programming. Uppsala University, Disciplinary Domain of Science and Technology. Mathematics and Computer Science, Department of Information Technology, 2011
- [5] Do. T. B. N, - Numerical Feature-tree for testing, Tạp chí khoa học và công nghệ, tập 52- số 6C, 2014 pp 57-67
- [6] Kitamura, T., Do T.B.N., Ohsaki, H., Fang, L., and Yatabe, S. -Test-case design by feature trees, In Proceedings of the 4th International Symposium On Leveraging Applications of Formal methods, Verification and Validation, 2012, pp. 458-473
- [7] Kitamura, T., Yamada A., Hatayama G., Artho C., Choi E.H., Do T.B.N., Oiwa Y., Sakuragi S., “Combinatorial Testing for Tree-structured Test Models with Constraints”, in Proceedings of the 2015 IEEE International Conference on Software Quality, Reliability and Security (QRS2015), pp.141-150 IEEE CPS, 2015

Website tham khảo

- [8] [https:// www.pairwise.org](https://www.pairwise.org)