

Luận văn thạc sĩ

Chương 1

Giới thiệu

Chương 2

Kiến thức cơ sở

2.1 Dạng đặc tả sử dụng hệ chuyển trạng thái được gắn nhãn

Định nghĩa 2.1: Hệ chuyển trạng thái gắn nhãn (Labeled Transition System - LTS) được định nghĩa là một bộ có thứ tự gồm 4 thành phần: $M = \langle Q, \Sigma, \delta, q_0 \rangle$ [2]

Trong đó:

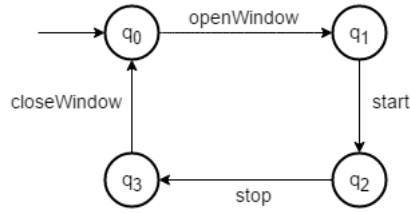
- $Q = \{q_0, q_1, \dots, q_{n-1}, q_n\}$ là tập các trạng thái,
- $\Sigma = \{w_0, w_1, \dots, w_{n-2}, w_{n-1}\}$ là tập các sự kiện,
- $\delta \subseteq Q \times \Sigma \times Q$ là hàm chuyển trạng thái, và
- $q_0 \subseteq Q$ là trạng thái khởi tạo.

Ta kí hiệu $q_i \xrightarrow{w_i} q_j$ nếu và chỉ nếu có một sự kiện w_i chuyển hệ thống từ trạng thái q_i sang trạng thái q_j , khi đó $(q_i, w_i, q_j) \in \delta$. Điều này có nghĩa khi một hệ thống đang ở trạng thái q_i , nếu có một sự kiện w_i xảy ra thì hệ thống sẽ chuyển sang trạng thái q_j . Tương tự, khi hệ thống đang ở trạng thái q_j nếu có một hành động w_k xảy ra thì hệ thống sẽ chuyển sang trạng thái q_k . Như vậy, chuỗi hai hành động $q_i \xrightarrow{w_i} q_j, q_j \xrightarrow{w_k} q_k$ có thể chuyển hệ thống từ trạng thái q_i sang trạng thái q_k . Khi đó, ta có thể kí hiệu $q_i \xrightarrow{w_i w_k} q_k$.

Ví dụ 2.1: Ví dụ về một hệ thống chuyển trạng thái được gắn nhãn.

Trên hình 2.1 là một ví dụ về một LTS $M = \langle Q, \Sigma, \delta, q_0 \rangle$, trong đó:

- $Q = \{q_0, q_1, q_2, q_3\}$,



Hình 2.1: Một hệ thống chuyển trạng thái được gán nhãn.

- $\Sigma = \{openWindow, start, stop, closeWindow\}$,
- $\delta = \{(q_0, openWindow, q_1), (q_1, start, q_2), (q_2, stop, q_3), (q_3, closeWindow, q_0)\}$,
và
- q_0 là trạng thái khởi tạo.

Định nghĩa 2.2: Kích thước của một tập hợp [1].

Kích thước của một tập hợp $Q = \{q_0, q_1, \dots, q_{n-1}, q_n\}$ là số phần tử của tập hợp Q , kí hiệu là $|Q|$.

Ví dụ 2.2: Với LTS được cho bởi hình 2.1, tập các trạng thái $Q = \{q_0, q_1, q_2, q_3\}$ nên $|Q| = 4$.

Định nghĩa 2.3: Kích thước của một LTS [3].

Kích thước của một LTS $M = \langle Q, \Sigma, \delta, q_0 \rangle$ là số trạng thái của M , kí hiệu là $|M|$, trong đó $|M| = |Q|$. **Ví dụ 2.3:** Với LTS được cho bởi hình 2.1, kích thước của LTS đó là $|M| = |Q| = 4$.

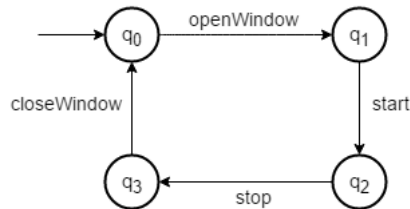
Định nghĩa 2.4: Vết của LTS.

Vết của một LTS $M = \langle Q, \Sigma, \delta, q_0 \rangle$ là một chuỗi hữu hạn các sự kiện có dạng $\sigma = w_0 w_1 \dots w_k$ với $w_k \in \Sigma$ và $0 \leq k \leq n$ sao cho tồn tại trạng thái $q_i \in Q$ mà $q_0 \xrightarrow{w} q_i$

Như vậy, vết σ của LTS M là một chuỗi các sự kiện có thể quan sát được mà M có thể thực hiện được từ trạng thái khởi tạo q_0 .

Ví dụ 2.4: Vết của LTS.

Hình 2.2 minh là một LTS $M = \langle Q, \Sigma, \delta, q_0 \rangle$, trong đó:



Hình 2.2: Minh họa vết của LTS.

- $Q = \{q_0, q_1, q_2, q_3\}$,
- $\Sigma = \{openWindow, start, stop, closeWindow\}$,
- $\delta = \{(q_0, openWindow, q_1), (q_1, start, q_2), (q_2, stop, q_3), (q_3, closeWindow, q_0)\}$,
và
- q_0 là trạng thái khởi tạo.

Ta thấy, chuỗi các hành động *openWindow start stop* là một vết của M, bởi vì tại trạng thái khởi tạo q_0 , khi sự kiện *openWindow* xảy ra, hệ thống chuyển sang trạng thái q_1 , tiếp tục xảy ra sự kiện *start* hệ thống chuyển sang trạng thái q_2 , khi xảy ra sự kiện *stop* hệ thống chuyển sang trạng thái q_3 . Chuỗi các hành động *openWindow start stop* chuyển hệ thống từ trạng thái khởi tạo q_0 sang trạng thái $q_3 \in Q$ nên chuỗi các hành động *openWindow start stop* là một vết của LTS. Tương tự, chuỗi các hành động *openWindow, openWindow start, openWindow start stop closeWindow, openWindow start stop closeWindow openWindow, ...* đều là vết của M.

Định nghĩa 2.5: Ngôn ngữ của LTS.

Ngôn ngữ của LTS M kí hiệu là $L(M)$ được định nghĩa như sau:

$$L(M) = \{\alpha \mid \alpha \text{ là một vết của } M\}$$

Ví dụ 2.5: Ví dụ về ngôn ngữ của LTS.

Với LTS M như ở hình 2.2, ngôn ngữ của M là:

$$L(M) = \{openWindow, openWindow start, openWindow start stop, \dots\}$$

2.2 Dạng đặc tả sử dụng hàm logic (Bool)

Định nghĩa 2.6: Hàm logic [1].

$B = \{T, F\}$ là miền giá trị logic. Với X là tập hợp các biến logic, một hàm logic $\theta(X)$ được định nghĩa: $\theta(X) : B^{|X|} \rightarrow B$.

Ví dụ 2.6: Với X là tập hợp gồm 3 phần tử, $X = \{x, y, z\}$ với $x, y, z \in B$, cho hàm logic $\theta(x, y, z) = x \wedge y \vee z$ ta được một ánh xạ $\theta(X) : B^3 \rightarrow B$.

Định nghĩa 2.7: Phép gán [1].

Với X là tập hợp các biến logic, phép gán v được định nghĩa: $v : X \rightarrow B$.

Ví dụ 2.7: Với X là tập hợp gồm 3 phần tử, $X = \{x, y, z\}$ với $x, y, z \in B$, $v(x) = T$, $v(x) = F$, $v(y) = T$, $v(y) = F$, $v(z) = T$ và $v(z) = F$ là các phép gán trên tập X.

Định nghĩa 2.8: Phép gán hàm [1].

Với $\phi(X)$ là hàm một logic trên tập X, v là một phép gán trên tập X, phép gán hàm kí hiệu $\phi[v]$ là kết quả thu được khi thay các phần tử $x \in X$ bởi $v(x)$.

Với X và X' là các tập biến logic, $\psi(X, X')$ là hàm logic trên hai tập X và X' , với $v(X)$ và $v(X')$ lần lượt là các phép gán trên tập X và X' , kí hiệu $\psi[v, v']$ là kết quả thu được khi thay một cách tương ứng $x \in X$ bởi $v(X)$ và $x' \in X'$ bởi $v'(X')$.

Ví dụ 2.8: Với $X = \{x\}$, $X' = \{x'\}$ là tập hợp các biến logic, $\phi(x) = \bar{x}$ là một hàm logic trên tập X . Nếu $v(x) = T$ thì $\phi[v] = F$ và nếu $v(x) = F$ thì $\phi[v] = T$. Với $\psi(x, x') = x \vee x'$ là một hàm logic trên tập X và X' , nếu $v(x) = T$, $v(x') = F$ thì $\psi[v, v'] = T \vee F = T$.

Định nghĩa 2.9: Dạng đặc tả sử dụng hàm logic.

Dạng đặc tả sử dụng hàm logic là một bộ có thứ tự gồm 4 phần tử:

$$N = \langle X, E, \tau(X, E, X'), \iota(X) \rangle$$

Trong đó:

- X là tập các biến logic dùng để biểu diễn các trạng thái của hệ thống. $X = \{x_0, x_1, \dots, x_n\}$,
- E là tập các biến logic dùng để biểu diễn các hành vi của hệ thống. $E = \{e_0, e_1, \dots, e_{n-1}, e_n\}$,
- $\tau \subseteq X \wedge E \wedge X$ là hàm logic biểu diễn việc chuyển trạng thái của hệ thống, và
- $\iota(X)$ là hàm logic dùng để biểu diễn các trạng thái khởi tạo của hệ thống.

Ví dụ 2.9: Với dạng đặc tả sử dụng hàm logic $N = \langle X, E, \tau(X, E, X'), \iota(X) \rangle$.

Trong đó:

- $X = \{x_1, x_2, x_5, x_6\}$,
- $E = \{x_3, x_4\}$
- $\tau(X, E, X') = \{(\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3 \wedge \bar{x}_4 \wedge \bar{x}_5 \wedge x_6), (\bar{x}_1 \wedge \bar{x}_2 \wedge x_3 \wedge \bar{x}_4 \wedge x_5 \wedge \bar{x}_6), \}$

Với $\tau(x, e, x')$ là hàm logic biểu diễn việc chuyển trạng thái của hệ thống, xét một chuyển trạng thái thứ i bất kì, khi $\tau[x_i, e_i, x_{i+1}] = \text{True}$ là một bộ 3 các phép gán cho các hàm mã hóa x_i, e_i, x_{i+1} thì $v[x_i] = \text{True}$, $v[e_i] = \text{True}$ và $v[x_{i+1}] = \text{True}$. Chuỗi hữu hạn các phép gán cho hàm mã hóa các sự kiện $\xi = v(e_0)v(e_1)\dots v(e_{n-1})$ được gọi là vết của N . Khi đó, $\iota[v^0] = \text{True}$ và $\tau[v^i, v^{i+1}] = \text{True}$ với $\forall i : 0 \leq i < n$.

Tập các vết của N được gọi là ngôn ngữ của N , kí hiệu là $L(N)$, $L(N) = \{\xi \mid \xi$

là một vết của N

2.2.1 Khái niệm về bảng ánh xạ

Bảng ánh xạ (mapping) là một bảng dùng để lưu lại các ánh xạ khi chuyển đổi từ dạng đặc tả sử dụng LTS sang dạng đặc tả sử dụng hàm logic và ngược lại, Gọi Map là kí hiệu của bảng ánh xạ. Với một LTS $M = \langle Q, \Sigma, \delta, q_0 \rangle$ và một dạng đặc tả sử dụng hàm logic $N = \langle X, E, \tau(x, e, x'), \iota(x) \rangle$. Ta định nghĩa:

$$\text{Map} = \{Q = Q_1 \cup Q_2, \Sigma, \delta(q, e, q'), X = X_1 \cup X_2, E, \tau(x, e, x')\}$$

Trong đó:

- $Q = Q_1 \cup Q_2$ là tập các trạng thái của LTS. Với Q_1 là tập các trạng thái đầu vào (from), Q_2 là tập các trạng thái đầu ra (to).
- Σ là tập các sự kiện của LTS,
- $\delta(q, e, q')$ là hàm chuyển trạng thái, $X = X_1 \cup X_2$ là tập các biến logic dùng để biểu diễn tập các trạng thái. Với X_1, X_2 lần lượt là dạng đặc tả sử dụng hàm logic của tập các trạng thái đầu vào và đầu ra,
- E là tập các biến logic dùng để biểu diễn các hành vi của hệ thống, $E = \{e_0, e_1, \dots, e_{n-1}\}$, và
- $\tau(x, e, x')$ là hàm logic biểu diễn việc chuyển trạng thái của hệ thống.

Ví dụ: Với một LTS như hình vẽ. Ta có tập các trạng thái $Q = \{a, b, c\}$. Với tập các trạng thái đầu vào $Q_1 = \{a, b, c\}$, tập trạng thái đầu ra $Q_2 = \{b, c, a\}$ và tập các sự kiện $E = \{send, out, ack\}$.

Ta có bảng ánh xạ:

Q_1	a	b	c	Q_2	a	b	c
X_1	x_0x_1	$x_0\bar{x}_1$	\bar{x}_0x_1	X_2	x_4x_5	$x_4\bar{x}_5$	\bar{x}_4x_5

Σ	send	out	ack
E	x_2x_3	$x_2\bar{x}_3$	\bar{x}_2x_3

$\delta(q, e, q')$	$\delta(a, send, b)$	$\delta(b, out, c)$	$\delta(c, ack, a)$
$\tau(x, e, x')$	$x_0x_1 \wedge x_2x_3 \wedge x_4\bar{x}_5$	$x_0\bar{x}_1 \wedge x_2\bar{x}_3 \wedge x_4\bar{x}_5$	$\bar{x}_0x_1 \wedge \bar{x}_2x_3 \wedge x_4x_5$

Bảng 2.1: Ví dụ về bảng ánh xạ

Chương 3

Phương pháp chuyển đổi

3.1 Thuật toán

Đầu vào (Input): Một hệ chuyển trạng thái gắn nhãn.

Đầu ra (Output): Dạng đặc tả sử dụng hàm logic và bảng ánh xạ.

Thuật toán chia làm 4 bước:

- Bước 1: Mã hóa tập các trạng thái đầu vào Q_1 - Thuật toán 1.
- Bước 2: Mã hóa tập các trạng thái đầu ra Q_2 - Thuật toán 1.
- Bước 3: Mã hóa tập các sự kiện - Thuật toán 2.
- Bước 4: Mã hóa tập các chuyển trạng thái - Thuật toán 3.

3.1.1 Thuật toán mã hóa tập các trạng thái

Thuật toán 3.1: Mã hóa tập các trạng thái

Đầu vào: Tập các trạng thái Q .

Đầu ra : Dạng mã hóa của tập các trạng thái.

```
1 for mỗi trạng thái  $q_i$  trong tập các trạng thái  $Q$  do
2    $x = \text{True}$ 
3    $k = \text{Thứ tự của } q_i \text{ trong } Q$ 
4   Chuyển  $k$  sang số nhị phân với độ dài  $\lceil \log_2(|Q|) \rceil + 1$  bit
5   for mỗi bit  $x_i$  trong chuỗi nhị phân biểu diễn  $k$  do
6     if  $x_i = 1$  then
7       | Biểu diễn  $x_i$  dưới dạng  $x_i$ 
8     else
9       | Biểu diễn  $x_i$  dưới dạng  $\bar{x}_i$ 
10    end
11     $x = \bigvee_{i=1}^n \{x_i \wedge x\}$ 
12  end
13  Lưu  $x$  vào bảng ánh xạ ứng với vị trí của  $q_i$ .
14 end
```

Với Q là tập các trạng thái $Q = \{q_0, q_1, \dots, q_{n-1}, q_n\}$. Số biến logic cần dùng để mã hóa tập các trạng thái Q là $\lceil \log_2 n \rceil + 1 = m$. Gọi x_1, x_2, \dots, x_m là chuỗi biến logic dùng để mã hóa các trạng thái của Q . Mỗi trạng thái q_i trong Q sẽ được biểu diễn dưới dạng $x_1 \wedge x_2 \wedge \dots \wedge x_m$.

Xét trạng thái bất kì q_k trong tập Q , theo bước (2) ta xác định được thứ tự của q_k trong tập Q là k , theo bước (3) thì k sẽ biểu diễn dưới dạng số nhị phân $\lceil \log_2 n \rceil + 1$ bit. Theo bước (4)-(10) chúng ta sẽ mã hóa được trạng thái q_k . Thêm vào đó, theo bước (11) dạng biểu diễn của trạng thái q_k sẽ được lưu vào trong bảng ánh xạ. Mục đích của việc lưu vào bảng ánh xạ là để dùng cho việc chuyển ngược lại từ dạng đặc tả sử dụng hàm logic sang dạng đặc tả sử dụng LTS. Bởi vì, thứ tự của mỗi trạng thái trong q_k là duy nhất nên số nhị phân biểu diễn thứ tự của trạng thái q_k cũng sẽ là duy nhất, vì thế dạng mã của trạng thái q_k là duy nhất.

Một cách tương tự cho các trạng thái khác trong Q , chúng ta sẽ mã hóa cho tất cả các trạng thái. Thêm vào đó, sau khi mã hóa, các thông tin về trạng thái và dạng mã hóa của các trạng thái này đều được lưu vào bảng ánh xạ, nhằm mục đích sử dụng cho việc chuyển đổi ngược lại từ dạng đặc tả sử dụng hàm logic sang dạng đặc tả sử dụng LTS.

Độ phức tạp: Độ phức tạp của thuật toán là $O(n)$, trong đó n là kích thước

của tập các trạng thái cần mã hóa.

3.1.2 Thuật toán mã hóa tập các sự kiện

Thuật toán 3.2: Mã hóa tập các sự kiện

Đầu vào: Tập các sự kiện Σ .

Đầu ra : Dạng mã hóa của tập các sự kiện.

```
1 for mỗi sự kiện  $w_i$  trong tập các sự kiện  $\Sigma$  do
2    $e = \text{True}$ 
3    $k = \text{Thứ tự của } w_i \text{ trong } \Sigma$ 
4   Chuyển  $k$  sang số nhị phân với độ dài  $\lceil \log_2(|\Sigma|) \rceil + 1$  bit
5   for mỗi bit  $x_i$  trong chuỗi nhị phân biểu diễn  $k$  do
6     if  $x_i = 1$  then
7       | Biểu diễn  $x_i$  dưới dạng  $x_i$ 
8     else
9       | Biểu diễn  $x_i$  dưới dạng  $\bar{x}_i$ 
10    end
11     $e = \bigvee_{i=1}^n \{x_i \wedge e\}$ 
12  end
13  Lưu  $e$  vào bảng ánh xạ ứng với vị trí của  $w_i$ 
14 end
```

3.1.3 Thuật toán mã hóa tập các chuyển trạng thái

Thuật toán 3.3: Mã hóa tập các chuyển trạng thái

Đầu vào: Tập các chuyển trạng thái của LTS.

Đầu ra : Tập các chuyển trạng thái được biểu diễn bằng dạng đặc tả sử dụng hàm logic.

```
1 for mỗi chuyển trạng thái được biểu diễn  $\delta(q, w, q')$  do
2   Lấy dạng biểu diễn của  $q$  từ bảng ánh xạ
3   Lấy dạng biểu diễn của  $w$  từ bảng ánh xạ
4   Lấy dạng biểu diễn của  $q'$  từ bảng ánh xạ
5   Biểu diễn chuyển trạng thái dưới dạng:  $x_i \wedge e_i \wedge x_{i+1}$ 
6 end
7 return  $\tau = \bigvee_{i=1}^n \{x_i \wedge e_i \wedge x_{i+1}\}$ 
```

3.2 Mã hóa các chuyển trạng thái

Một chuyển trạng thái (Transition) được biểu diễn là một bộ 3 $\delta = Q_1 \times \Sigma \times Q_2$ hay $\delta = q_i \wedge e_i \wedge q_{i+1}$ với $\forall q_i \in Q_1, \forall e_i \in \Sigma$ và $\forall q_{i+1} \in Q_2$. Theo bước (2) trạng thái đầu vào $\text{from} = q_i$ sẽ được chuyển sang dạng đặc tả sử dụng hàm logic. Theo bước (3) sự kiện e_i sẽ được chuyển sang dạng đặc tả sử dụng hàm logic. Theo bước (4) trạng thái đầu ra $\text{to} = q_{i+1}$ sẽ được chuyển sang dạng đặc tả sử dụng hàm logic. Sau bước (5), mỗi chuyển đổi trạng thái sẽ được biểu diễn dưới dạng $x_i \wedge e_i \wedge x_{i+1}$.

Với thuật toán mã hóa 3, đầu ra của thuật toán này là $\tau = \bigvee_{i=1}^n \{x_i \wedge e_i \wedge x_{i+1}\}$. Xét $\tau_i = \{x_i \wedge e_i \wedge x_{i+1}\}$, $\tau_j = \{x_j \wedge e_j \wedge x_{j+1}\}$ với $j = i + 1$, ta có $\psi[\tau_i] = \text{True}$ thì $v[x_i] = \text{True}$, $v[x_{i+1}] = \text{True}$, $v[e_i] = \text{True}$, $\psi[\tau_j] = \text{True}$ thì $v[x_j] = \text{True}$, $v[x_{j+1}] = \text{True}$, $v[e_j] = \text{True}$, khi đó $\tau[e_i, e_j] = \text{True}$.

3.3 Chứng minh

Để chứng minh ngôn ngữ của dạng đặc tả được biểu diễn bằng LTS tương đương với ngôn ngữ của dạng đặc tả sử dụng hàm logic, chúng ta cần chứng minh 2 mệnh đề:

- Mệnh đề 1: Ngôn ngữ của dạng đặc tả hệ thống biểu diễn bằng LTS sau khi chuyển đổi sang dạng đặc tả biểu diễn bằng hàm logic được đoán nhận bởi ngôn ngữ của dạng đặc tả được biểu diễn bởi hàm logic. (1)
- Mệnh đề 2: Ngôn ngữ của dạng đặc tả được chuyển đổi từ hàm logic sau khi chuyển đổi sang dạng đặc tả biểu diễn bằng LTS được đoán nhận bởi ngôn ngữ của LTS.

Chứng minh mệnh đề 1:

Gọi $\alpha = w_0 w_1 \dots w_{n-1}$ là một dẫn xuất bất kì trong ngôn ngữ L, áp dụng thuật toán mã hóa như đã trình bày ở 4.1, ta được một dẫn xuất $\alpha' = e_0 e_1 \dots e_{n-2} e_{n-1}$ và một bảng mapping A, ta sẽ chứng minh α' cũng được đoán nhận với ngôn ngữ của dạng đặc tả được biểu diễn bởi hàm logic.

Thật vậy, vì α' là đầu ra của α trong thuật toán mã hóa, mà w_0 là sự kiện đầu tiên của LTS nên $\iota[e_0] = \text{True}$. Mặt khác, $\delta(q_i, w_i) = q_{i+1}$, $\delta(q_{i+1}, w_{i+1}) = q_{i+2}$ với $\forall i : 0 \leq i < n$ nên $\tau[x_i, e_i, x_{i+1}] = \text{True}$ và $\tau[x_{i+1}, e_{i+1}, x_{i+2}] = \text{True}$, $\tau[e_i, e_{i+1}] = \text{True}$. Do đó, α' là một vết của N.

Chứng minh mệnh đề 2:

Gọi $\xi = e_0e_1\dots e_{n-1}$ là một vết của N . Gọi A là bảng ánh xạ lưu các ánh xạ khi chuyển đổi từ dạng đặc tả sử dụng LTS sang dạng đặc tả sử dụng hàm logic. Áp dụng thuật toán mã hóa đã trình bày ở 4.1, ta được một chuỗi $\xi' = w_0w_1\dots w_{n-2}w_{n-1}$. Ta sẽ chứng minh ξ' được đoán nhận bởi LTS. Thật vậy, với e_0 là dạng biểu diễn cho của sự kiện thứ 0 của dạng đặc tả sử dụng hàm logic, khi đó $\iota[e_0] = \text{True}$, dựa vào bảng ánh xạ e_0 tương ứng với w_0 của dạng biểu diễn diễn sử dụng LTS do đó w_0 là sự kiện thứ 0 hay sự kiện đầu tiên của LTS. Vì ξ là một vết của N nên $\tau[e_i, e_{i+1}] = T$ với $\forall i : 0 \leq i < n$, vì thế $\tau[x_i, e_i, x_{i+1}] = \text{True}$, $\tau[x_{i+1}, e_{i+1}, x_{i+2}] = \text{True}$. Dựa vào bảng ánh xạ, x_i ứng với q_i , x_{i+1} ứng với q_{i+1} , x_{i+2} ứng với q_{i+2} , e_i ứng với w_i , e_{i+1} ứng với w_{i+1} trong dạng đặc tả sử dụng LTS, do đó $\delta(q_i, w_i) = q_{i+1}$, $\delta(q_{i+1}, w_{i+1}) = q_{i+2}$ với $\forall i : 0 \leq i < n$, δ là hàm chuyển trạng thái của LTS. Ta thấy, ξ' hoàn toàn được đoán nhận bởi ngôn ngữ của LTS.

Chương 4

Ví dụ về việc chuyển đổi
giữa các dạng đặc tả

Tài liệu tham khảo

- [1] Yu-Fang Chen, Edmund M. Clarke, Azadeh Farzan, Ming-Hsien Tsai, Yih-Kuen Tsay, and Bow-Yaw Wang, *Automated Assume-Guarantee Reasoning through Implicit Learning*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] Pham Ngoc Hung, *Assume-Guarantee Verification of Evolving Component-Based Software*. Japan Advanced Institute of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy, September, 2009.
- [3] P. N. Hung, N. V. Ha, T. Aoki and T. Katayama, *On Optimization of Minimized Assumption Generation Method for Component-based Software Verification* IEICE Trans. on Fundamentals, Special Issue on Software Reliability Engineering, Vol. E95-A, No.9, pp. 1451-1460, Sep. 2012.