ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

PHẠM HỒNG THÁI

MỘT SỐ PHƯƠNG PHÁP KIỂM CHỨNG TÍNH ĐÚNG ĐẮN CỦA HỆ THỜI GIAN THỰC BẰNG THUẬT TOÁN

TÓM TẮT LUẬN ÁN TIẾN SĨ TOÁN HỌC

Hà Nội - 2005

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Người hướng dẫn khoa học : TS. Đặng Văn Hưng
Đồng hướng dẫn : PGS. TS. Đinh Mạnh Tường

Phản biện 1 : PGS. TS. Nguyễn Quang Hoan
Học viện Bưu chính Viễn thông
Phản biện 2 : PGS. TS. Nguyễn Đình Hóa
Viện Công nghệ Thông tin, ĐHQGHN
Phản biện 3 : PGS. TS. Nguyễn Đức Nghĩa
Trường Đại học Bách Khoa Hà Nội

Luận án được bảo vệ trước Hội đồng cấp Nhà nước chấm luận án Tiến sĩ họp tại Trường Đại học Công nghệ Vào hồi 15 giờ, ngày 20 tháng 05 năm 2005

Có thể tìm hiểu Luận án tại :
– Thư viện Quốc gia
– Trung tâm Thông tin – Thư viện Đại học Quốc gia Hà Nội

MỞ ĐẦU

Kiểm chứng mô hình là một trong các phương pháp kiểm tra tính đúng đắn của hệ thống đang ngày càng trở nên phổ biến. Với những thành tựu của mình, kiểm chứng mô hình đã đóng góp nhiều phương pháp, thuật toán, cấu trúc dữ liệu và trên cơ sở đó cung cấp nhiều bộ công cụ cho phép việc kiểm tra có thể được tiến hành một cách tự động. Tuy nhiên cho đến hiện nay, **tình hình chung** của kiểm chứng mô hình chủ yếu tập trung vào các hệ phi thời gian hoặc các tính chất thời điểm của hệ thời gian thực, tính chất thời khoảng vẫn còn ít thuật toán và hầu như các thuật toán này chỉ giải quyết bài toán theo ngữ nghĩa đã thu hẹp của tính chất cần kiểm chứng.

Từ đó, **mục đích của luận án** là quan sát và thiết kế thuật toán kiểm chứng các tính chất thời khoảng theo ngữ nghĩa tổng quát của nó. Đó là lớp các hệ thống được biểu diễn bởi *ôtômat thời gian* và các tính chất khoảng được biểu diễn bởi các công thức trong lôgic khoảng như công thức *khoảng tuyến tính* (LDP), công thức *bất biến khoảng tuyến tính* (LDI).

Để đề xuất thuật toán kiểm chứng các công thức trên, luận án sử dụng **phương pháp** *qui hoạch tuyến tính*. Trong đó, chúng tôi đề nghị phép toán *hợp song song* để biểu diễn lưới ôtômat thời gian thực bởi biểu thức chính quy thời gian, từ đó thiết lập bài toán qui hoạch tuyến tính và chứng minh kết quả giải bài toán qui hoạch tuyến tính cũng là kết quả của bài toán kiểm chứng. Một phương pháp khác cũng được chúng tôi sử dụng, đó là *duyệt đồ thị vùng đạt được*. Để sử dụng phương pháp này chúng tôi chứng minh tính *rời rạc hoá được* của các công thức LDP, LDI, từ đó xây dựng các đồ thị trọng số, đồ thị "rời rạc", và các thuật toán duyệt trên những đồ thị này để cung cấp câu trả lời cho bài toán kiểm chứng. **Kết quả** này đã được báo cáo tại các xemine, hội nghị trong nước và nước ngoài, đã đăng trong tạp chí khoa học của Đại học Quốc gia Hà Nội, và được xuất bản bởi các nhà xuất bản IEEE và Springer-Verlag.

Cấu trúc của luận án gồm 4 chương. Trong chương 1 chúng tôi trình bày tóm tắt một số đặc trưng của kiểm chứng mô hình, các kết quả và các mặt hạn chế còn tồn tại trong lĩnh vực kiểm chứng tính chất thời khoảng cho đến thời điểm hiện nay. Các công cụ cơ bản

dùng để đặc tả hệ thống và tính chất như ôtômat thời gian và lôgic khoảng được trình bày trong chương 2. Chương 3 và 4 dùng để trình bày kết quả. Trong đó, ở chương 3 chúng tôi trình bày phương pháp kiểm chứng bằng qui hoạch tuyến tính và trong chương 4 chúng tôi trình bày phương pháp duyệt đồ thị phân vùng trên cơ sở rời rạc hóa được của các công thức. Để minh hoạ các thuật toán đã trình bày, một bộ kiểm chứng mô hình đơn giản do chúng tôi cài đặt cũng được mô tả trong phụ lục A.

Chương 1

KIỂM CHỨNG MÔ HÌNH VÀ HỆ THỜI GIAN THỰC

1.1 ĐẶC TẢ VÀ KIỂM TRA HỆ THỐNG

Mục đích chính của các phương pháp hình thức là giúp tạo ra khả năng cho phép xây dựng và phát triển các hệ thống hoạt động được một cách đúng đắn, đáng tin cậy. Cách đây vài thập kỷ, nhiệm vụ này là hết sức khó khăn, chủ yếu từ việc chưa có một phương pháp nhất quán để mô tả hệ thống, từ đó không tạo được nền tảng để xây dựng những kỹ thuật kiểm tra thống nhất và dẫn tới cách thức kiểm tra không hữu hiệu, thiếu tính thuyết phục.

Do vậy, để thực hiện nhiệm vụ kiểm tra, trước hết hệ thống phải được **đặc tả** một cách rõ ràng và nhất quán. Đặc tả là quá trình mô tả hệ thống và các tính chất mong muốn của nó bằng một ngôn ngữ cụ thể. Tính chất của hệ thống có thể bao gồm nhiều loại: dáng điệu chức năng, dáng điệu thời gian, đặc trưng hoạt động, hoặc cấu trúc nội tại. Một ngôn ngữ đặc tả cần đủ khả năng để mô tả được hệ thống và các loại tính chất của nó. Thông thường, đặc tả hình thức dùng một ngôn ngữ với cú pháp và ngữ nghĩa xác định nên mang tính ổn định và nhất quán cao. Ví dụ một vài ngôn ngữ đặc tả như Z, VDM, RAISE sử dụng các cấu trúc toán học như tập hợp, quan hệ, hàm ... và hiện nay vẫn đang được sử dụng khá thành công trong nhiều lĩnh vực. Các hệ thống cũng có thể được trừu tượng hoá bằng một số công cụ biểu diễn khác như ôtômat, biểu thức chính quy, lưới Petri, đại số tiến trình hay bằng các hệ lôgic.

Theo truyền thống, **kiểm tra** là chứng minh về mặt toán học chương trình hay hệ thống thoả mãn yêu cầu đặt ra, nói cách khác hệ

thống phải làm việc đúng như phân tích và thiết kế trong phần đặc tả. Trong thập kỷ 70 đã có nhiều công trình trình bày các *kỹ thuật chứng minh* được đưa ra bởi Dijkstra, Floyd, Gries, Hoare, Owicki, Manna, Plueni và nhiều tác giả khác. Vào đầu thập kỷ 80, Clarke, Emerson, Queille, Sifakis đề nghị một phương pháp khác có nhiều ưu điểm hơn được gọi là *kiểm chứng mô hình*. Phương pháp này cho phép người dùng có thể sử dụng máy tính và các phần mềm được lập trình sẵn để kiểm tra tự động sản phẩm thiết kế của mình trước khi sản xuất. Dựa trên hai phương pháp này nhiều công cụ kiểm tra đã ra đời như *hệ chứng minh định lý* (*theorem prover*) hay các *bộ kiểm chứng mô hình* (*model checker*). Trong luận án này chúng tôi quan tâm đến phương pháp thứ hai tức kiểm chứng mô hình và trong phần tiếp theo chúng tôi sơ lược lại một vài kết quả của phương pháp đó.

1.2 KIỂM CHỨNG MÔ HÌNH

Thuật ngữ *kiểm chứng mô hình* (*model checking*) được E.M. Clarke và E.A. Emerson đưa ra lần đầu tiên vào năm 1981: "Kiểm chứng mô hình là một kỹ thuật *tự động* mà cho trước một *mô hình trạng thái hữu hạn* và một *tính chất logic*, kỹ thuật sẽ cho phép *kiểm tra* một cách hệ thống một trạng thái (hoặc mô hình) liệu có thoả mãn tính chất đó". Cụ thể bài toán kiểm chứng mô hình có thể được viết dưới dạng: $\mathbf{S} \models \mathbf{P}$?, trong đó **S** là mô hình hệ thống, **P** là một tính chất và $\models$ kí hiệu cho vị từ "thoả được". Câu hỏi đặt ra là "Liệu hệ thống **S** có thoả mãn tính chất **P** ?"

Thông thường **S** được thể hiện bởi ôtômat và **P** được biểu diễn bởi các công thức trong lôgic nào đó. Có vài hướng tiếp cận để giải bài toán. Hướng tiếp cận thứ nhất được nghiên cứu bởi Clarke và Emerson (1981), và cũng bởi Queille và Sifakis (1981) một cách độc lập. Theo hướng tiếp cận này, một thủ tục duyệt trên không gian trạng thái của hệ thống sẽ được sử dụng để kiểm tra liệu bất kỳ dãy chuyển trạng thái nào của hệ cũng thoả mãn tính chất. Hướng tiếp cận thứ hai dựa vào bài toán kiểm tra tính rỗng của ngôn ngữ (Alur, Kurskan 1994). Theo hướng tiếp cận này, tính chất **P** cũng được thể hiện bởi một ôtômat. Nếu ngôn ngữ $\mathcal{L}(\mathbf{S})$ được sinh bởi **S** là chứa trong ngôn ngữ $\mathcal{L}(\mathbf{P})$ được sinh bởi **P** thì **S** thoả **P**. Mặt khác, do $\mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{P}) \Leftrightarrow \mathcal{L}(\mathbf{S} \times \neg\mathbf{P}) = \varnothing$ nên có thể chuyển bài toán trên về bài toán kiểm tra tính rỗng của ngôn ngữ sinh bởi ôtômat tích.

Nhược điểm của phương pháp kiểm chứng mô hình là bài toán bùng nổ không gian trạng thái (số trạng thái tăng nhanh theo số thành phần của hệ thống). Để khắc phục, một số tác giả đưa ra kỹ thuật biểu diễn bằng kí hiệu (*symbolic model checking*), với đặc trưng chính là biểu diễn mọi thành phần của bài toán (trạng thái, phép chuyển, tính chất) thông qua các kí hiệu đại diện. Kỹ thuật này cho phép sử dụng các sơ đồ biểu diễn nhị phân BDD (*Binary Decision Diagram*) với dữ liệu được cài đặt tốn rất ít bộ nhớ và các phép toán trên nó được thực hiện một cách rất hữu hiệu. Trên cơ sở này năm 1992, Clarke và McMillan đã xây dựng bộ kiểm chứng mô hình đầu tiên với tên gọi SMV (*Symbolic Model Verifier*) cho phép kiểm chứng các hệ thống với số lượng trạng thái lên đến 10^{20} trạng thái. Các kỹ thuật tương tự và trên cơ sở đó một số bộ kiểm chứng mô hình khác cũng dần ra đời phục vụ cho kiểm chứng các hệ thời gian thực.

Hệ thời gian thực là các hệ thống mà tính đúng đắn của nó liên quan chặt chẽ đến các ràng buộc về mặt thời gian như thời gian thực hiện, thời gian hưởng ứng, chu kỳ tác vụ, thời gian trễ trong tương tác, truyền thông v..v... Đầu thập kỷ 90 một số công cụ đặc tả các hệ thời gian thực lần lượt ra đời mà nền móng cơ bản nhất là *ôtômat thời gian* (Alur và Dill – 1994), và các logic để đặc tả các tính chất cần kiểm chứng cũng được mở rộng thành các lôgic thời gian thực như *lôgic cây tính toán theo thời gian, lôgic khoảng*.

Tư tưởng phổ biến của các bộ kiểm chứng thời gian thực là duyệt đồ thị vùng (region graph) của ôtômat thời gian. Vì số lượng các vùng là rất lớn nên một số biện pháp thu gọn cũng được đề nghị. Ví dụ, kết hợp nhiều vùng thành miền (zones) dựa trên các kỹ thuật như mô phỏng (simulation), trừu tượng hoá (abstract), hoặc lược bỏ bớt các phép chuyển đan xen thừa trong hợp các thành phần bằng kỹ thuật thứ tự bộ phận (partial orders), kết hợp với tối ưu hoá cấu trúc dữ liệu. Các kỹ thuật symbolic và on-the-fly (kỹ thuật kiểm chứng song song với quá trình sinh trạng thái để tiết kiệm bộ nhớ) cũng được áp dụng, từ đó các bộ kiểm chứng thời gian thực thường xuyên được cải tiến và ngày càng hữu hiệu hơn. Tuy nhiên, các bộ kiểm chứng này cũng mới tập trung chủ yếu vào các tính chất *thời điểm*.

Để kiểm chứng các tính chất *thời khoảng* (là tính chất liên quan đến độ dài khoảng thời gian xuất hiện của các trạng thái) Zhou

Chaochen và đồng sự lần đầu tiên (1991) đã xây dựng một hệ lôgic có khả năng đặc tả các tính chất này được gọi là hệ *lôgic khoảng* (Duration Calculus – DC). Trên cơ sở đó một số thuật toán của Zhou Chaochen, Henzinger, Kesten, Dang Van Hung, Li XuanDong, Zhao Jianhua, Li Yong ... đã dần dần được xây dựng. Tuy nhiên, bài toán kiểm chứng tính chất thời khoảng là khó hơn rất nhiều so với kiểm chứng tính chất thời điểm, do vậy các kết quả trên chỉ mới dừng lại trên các lớp con của ôtômat thời gian và/hoặc với ngữ nghĩa thu hẹp của các tính chất khoảng. Trong luận án này chúng tôi đề nghị các thuật toán kiểm chứng với lớp hệ thống mở rộng hơn và các tính chất thời khoảng được xét trong ngữ nghĩa tổng quát hơn.

Chương 2

ĐẶC TẢ HỆ THỐNG VÀ TÍNH CHẤT

Để đặc tả hệ thống thời gian thực, luận án sử dụng *ôtômat thời gian*, còn tính chất thời khoảng sẽ được đặc tả bởi các *công thức khoảng* trong DC. Chương này giới thiệu tóm tắt về hai công cụ đó.

2.1 MÔ HÌNH THỜI GIAN

2.1.1 Thể hiện đồng hồ và ràng buộc thời gian

Để đánh dấu và đo lượng thời gian chúng ta sử dụng một tập hữu hạn X các *biến thời gian x* (hay còn gọi là các "*đồng hồ*") lấy giá trị mặc định trên tập số thực $\mathbb{R}^+$. Một bộ ν bất kỳ các giá trị đồng hồ được gọi là *thể hiện đồng hồ* (clock interpretations) hay còn gọi ngắn gọn là *thể hiện*. Một giá trị của *x* trong bộ ν được kí hiệu bởi ν(*x*).

Cho $\delta \in \mathbb{R}^+$ là một số thực không âm và $\lambda \subseteq X$ là tập con các đồng hồ của X. Khi đó ν + δ kí hiệu cho thể hiện gán mỗi đồng hồ *x* tới giá trị ν(*x*) + δ, và ν[λ := 0] là một thể hiện nhận được từ ν bằng cách gán 0 tới mỗi $x \in \lambda$. Một ràng buộc thời gian là một công thức φ được cho theo cú pháp $\varphi := x \le c \mid c \le x \mid x - y \le c \mid c \le x - y \mid \varphi \wedge \varphi$, ở đây $x, y \in X$ và $c \in \mathbb{N}$. Một thể hiện ν được gọi là thoả ràng buộc φ nếu thay mọi biến *x* trong φ bởi ν(*x*) ta nhận được một công thức đúng, và kí hiệu $\nu \vDash \varphi$, ngược lại ta kí hiệu $\nu \nvDash \varphi$.

2.1.2 Kỹ thuật phân vùng đồng hồ

Kí hiệu Φ(X) là tập các ràng buộc đồng hồ trên X. Cho $\phi \subseteq \Phi(X)$, gọi c_x là số nguyên c lớn nhất xuất hiện trong các ràng buộc của đồng hồ *x* trong ϕ. Alur và Dill đề xuất một quan hệ tương đương trên tập các minh họa hướng tới ϕ như sau. Kí hiệu $\lfloor t \rfloor$ và fr(*t*) lần lượt là phần nguyên và phần thập phân của số thực *t*. Hai thể hiện ν và ν' được gọi là tương đương và được kí hiệu $\nu \cong \nu'$ nếu:

- $\forall x \in X$, hoặc $\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$ hoặc cả hai cùng lớn hơn c_x,
- $\forall x, y \in X$ và $\nu(x) \le c_x$, $\nu(y) \le c_y$ thì $fr(\nu(x)) \le fr(\nu(y))$ khi và chỉ khi $fr(\nu'(x)) \le fr(\nu'(y))$,
- $\forall x \in X$ và $\nu(x) \le c_x$ thì $fr(\nu(x)) = 0$ khi và chỉ khi $fr(\nu'(x)) = 0$.

Quan hệ ≅ phân hoạch tập các minh họa đồng hồ thành các lớp tương đương gọi là *vùng đồng hồ* (clock regions) với tính chất: *hai minh họa thuộc cùng một vùng đồng hồ sẽ cùng thoả hoặc cùng không thoả một ràng buộc bất kỳ trong ϕ.* Vùng đồng hồ chứa ν được kí hiệu bởi [ν] và dễ dàng chứng minh được số vùng là hữu hạn.

2.2 ÔTÔMAT THỜI GIAN

2.2.1 Cú pháp và ngữ nghĩa

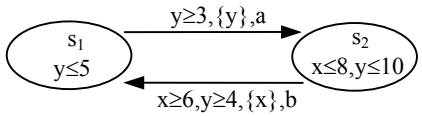
Ôtômat thời gian là một bộ $\mathcal{A} = <S, s_0, \Sigma, X, I, E>$, trong đó S là tập hữu hạn các *vị trí*, $s_0 \in S$ là *vị trí ban đầu*, Σ là tập hữu hạn các *nhãn*, X là tập hữu hạn các *đồng hồ*, I là một ánh xạ gán mỗi vị trí $s \in S$ với một ràng buộc đồng hồ I(*s*) được gọi là *bất biến* của *s*, $E \subseteq S \times \Phi(X) \times \Sigma \times 2^X \times S$ là tập hữu hạn các *phép chuyển vị trí*.

Một phép chuyển vị trí $e = <s, \varphi, a, \lambda, s'> \in E$ ($s, s' \in S$, $\varphi \in \Phi(X)$, $a \in \Sigma$, $\lambda \subseteq X$) cho phép ôtômat chuyển vị trí từ *s* tới *s'* với nhãn *a* nếu điều kiện φ được thoả mãn. Khi phép chuyển *e* được thực hiện các đồng hồ trong λ sẽ được khởi tạo lại giá trị về 0. Cụ thể, ta gọi (*s*, ν) là một trạng thái (của ôtômat thời gian $\mathcal{A}$) nếu *s* là một vị trí và ν là một minh họa đồng hồ bất kỳ sao cho $\nu \vDash I(s)$, khi đó phép chuyển trạng thái của ôtômat sẽ thuộc 1 trong 2 dạng sau: ta nói hệ chuyển trạng thái từ q = (*s*, ν) đến q' = (*s*, ν + δ) bằng một *phép chuyển liên tục* và kí hiệu $q \xrightarrow{\delta} q'$, nếu $\nu + \delta' \vDash I(s)$, $\forall \delta' \in [0, \delta]$, hoặc ta nói hệ chuyển trạng thái từ q = (*s*, ν) đến q' = (*s'*, ν[λ := 0]) bằng một *phép chuyển rời rạc* và kí hiệu $q \xrightarrow{a} q'$, nếu tồn tại phép chuyển $e = <s, \varphi, a, \lambda, s'>$ sao cho $\nu \vDash \varphi$ và $\nu[\lambda := 0] \vDash I(s')$.

Trường hợp hai phép chuyển $q \xrightarrow{\delta} q'$ và $q' \xrightarrow{a} q''$ xảy ra liên

tiếp, ta có thể kí hiệu: $q \xrightarrow{\delta,a} q''$, tức $(s, \nu) \xrightarrow{\delta,a} (s', \nu')$. Kí hiệu này minh hoạ hệ thống nằm tại s với thể hiện ν, thời gian trôi một khoảng từ 0 đến δ, đến thời điểm này nếu thể hiện đồng hồ $\nu + \delta$ thoả ràng buộc của phép chuyển vị trí thì hệ chuyển đến s' và một số giá trị trong $\nu + \delta$ được đặt về 0, $\nu + \delta$ chuyển thành ν' thỏa $I(s')$.

Trạng thái (s_0, ν_0) với s_0 là vị trí ban đầu và $\nu_0(x) = 0$ với mọi x được gọi là trạng thái ban đầu của ôtômat.



Hình 2.2. Ôtômat thời gian

Ví dụ: Ôtômat thời gian $\mathcal{A}$ trong hình 2.2 có hai đồng hồ x và y. Tập hợp các vị trí là $\{s_1, s_2\}$ cùng với các bất biến $I(s_1) = (y \le 5)$ và $I(s_2) = (x \le 8 \wedge y \le 10)$. $\mathcal{A}$ có hai phép chuyển vị trí $\{< s_1, y \ge 3, a, \{y\}, s_2 >, < s_2, x \ge 6 \wedge y \ge 4, b, \{x\}, s_1 >\}$. Đồng hồ y được đặt về 0 mỗi khi hệ chuyển từ s_1 đến s_2 và ngược lại khi chuyển từ vị trí s_2 về s_1 đồng hồ x sẽ được đặt về 0.

2.2.2 Đường chạy và dáng điệu của ôtômat thời gian

- Một *đường chạy* r của ôtômat thời gian $\mathcal{A}$ là một dãy hữu hạn hoặc vô hạn các phép chuyển trạng thái r: $(s_0,\nu_0) \xrightarrow{\delta_1,a_1} (s_1,\nu_1) \xrightarrow{\delta_2,a_2} \dots \xrightarrow{\delta_m,a_m} (s_m,\nu_m) \dots$ trong đó (s_0,ν_0) là trạng thái ban đầu của $\mathcal{A}$ và $\forall i \ge 1$, $(s_{i-1},\nu_{i-1}) \xrightarrow{\delta_i,a_i} (s_i,\nu_i)$ là một phép chuyển trạng thái của $\mathcal{A}$.
- Một *dáng điệu* ρ của ôtômat thời gian $\mathcal{A}$ tương ứng với đường chạy r là một dãy vô hạn hoặc hữu hạn các cặp vị trí - thời gian $\rho: (s_0, t_0), (s_1, t_1), \dots, (s_m, t_m) \dots$ với $t_0 = 0$ và $\delta_i = t_i - t_{i-1}$, $\forall i \ge 1$.

Với đường chạy r bất kỳ và dáng điệu ρ tương ứng, ta có : $t_i \le t_{i+1}$ và t_i là thời điểm hệ chuyển vị trí từ s_{i-1} tới s_i, $\forall i \ge 0$. Tức hệ thống nằm tại vị trí s_{i-1} trong $\delta_i = t_i - t_{i-1}$ đơn vị thời gian và chuyển đến vị trí s_i bởi một phép chuyển vị trí $<s_{i-1}, \varphi, a, \lambda, s_i>$ nào đó.

Dựa trên kỹ thuật phân vùng đồng hồ (tiểu mục 2.1.2), Alur và Dill xây dựng đồ thị vùng đạt được $\mathcal{RG}$ của ôtômat thời gian $\mathcal{A}$ với tính chất: nếu trong $\mathcal{A}$ tồn tại một đường chạy từ vị trí s_1 đến s_2 thì trong $\mathcal{RG}$ cũng có một đường đi nào đó từ đỉnh v_1 đến v_2 và ngược lại. Ở đây v_1 và v_2 là các đỉnh của đồ thị đặc trưng cho s_1 và s_2. Từ đó

bài toán kiểm chứng tính đạt được của một vị trí trong ôtômat có thể đưa về bài toán duyệt trên đồ thị vùng $\mathcal{RG}$.

2.2.3 Hợp song song của các ôtômat thời gian

Đối với nhiều hệ thống phức hợp được tạo thành từ các hệ thống đơn, chúng ta có thể sử dụng tích đồng bộ của các ôtômat thời gian để biểu diễn chúng. Lấy $\mathcal{A}_i = <S_i, s_{0i}, \Sigma_i, X_i, I_i, E_i>$ (i = 1, 2) là hai ôtômat thời gian. Hợp song song của $\mathcal{A}_1$, $\mathcal{A}_2$ (ký hiệu bởi $\mathcal{A}_1 \parallel \mathcal{A}_2$) là ôtômat tích của $\mathcal{A}_1$, $\mathcal{A}_2$ hoạt động đồng bộ theo các phép chuyển. Điều này có nghĩa bất kỳ phép chuyển nhãn a của $\mathcal{A}_1$ hoặc $\mathcal{A}_2$ cũng là phép chuyển của $\mathcal{A}$. Nếu $a \in \Sigma_1 \setminus \Sigma_2$ hoặc $a \in \Sigma_2 \setminus \Sigma_1$ phép chuyển này trong $\mathcal{A}$ được gọi là phép chuyển đan xen (interleaving) và nếu $a \in \Sigma_1 \cap \Sigma_2$ ta gọi phép chuyển này là phép chuyển đồng bộ (synchronising), tức một phép chuyển đồng bộ xảy ra trong $\mathcal{A}$ khi và chỉ khi nó xảy ra đồng thời cả trong $\mathcal{A}_1$ lẫn $\mathcal{A}_2$. Đối với các hệ thống hợp song song bởi nhiều hơn hai thành phần, định nghĩa cũng được mở rộng một cách tương tự.

2.3 LÔGIC KHOẢNG

Lôgic khoảng (duration calculus - DC) được xây dựng và phát triển bởi Zhou Chaochen và các đồng sự như một lôgic để lý giải và tính toán về thời gian xuất hiện của các trạng thái trong các hệ thời gian thực. Trong DC, mỗi trạng thái s của hệ thống được xem như một hàm bool theo thời gian s: **TIME** $\mapsto \{0, 1\}$, trong đó $s(t)$ nhận giá trị 1 nếu tại thời điểm t hệ thống nằm tại trạng thái s và nhận giá trị 0 trong trường hợp ngược lại. Với S là tập các trạng thái, ta gọi tập bất kỳ các hàm bool $\mathcal{I} = \{s_{\mathcal{I}}(t) \mid s \in S\}$ là một thể hiện của hệ thống.

2.3.1 Mô hình trong lôgic khoảng

Một mô hình σ của DC là bộ $(\mathcal{I}, [b, e])$ biểu diễn một quan sát dáng điệu của hệ thống. Nó bao gồm một thể hiện $\mathcal{I}$ và đoạn thời gian quan sát $[b, e]$ $(b, e \in \mathbb{R}^+, 0 \le b \le e < \infty)$. Với σ, tổng khoảng thời gian (time duration) xuất hiện của trạng thái s có thể được tính bởi:

$$d_s = \int_b^e s_I(t)dt .$$

Kí hiệu $\int s = d_s$, đây là toán tử quan trọng phân biệt sự khác nhau về bản chất của lôgic khoảng với các lôgic thời gian khác. Kí hiệu ℓ

được dùng để chỉ độ dài của đoạn quan sát, tức $\ell = e - b$.

Một tập mô hình DC bất kỳ xác định một hệ thời gian thực $\mathcal{S}$ nào đó và ngược lại. Tập mô hình xác định hệ thời gian thực $\mathcal{S}$ được kí hiệu bởi $\mathcal{M}(\mathcal{S})$, chẳng hạn một ôtômat thời gian $\mathcal{A}$ bất kỳ cũng tương ứng với tập mô hình $\mathcal{M}(\mathcal{A})$ nào đó.

2.3.2 Công thức khoảng và bài toán kiểm chứng mô hình

Một công thức viết trong DC được gọi là công thức khoảng. Cho đến hiện nay số lượng các lớp công thức quyết định được đã biết trong DC là còn rất hạn chế. Dưới đây là hai lớp công thức khoảng cơ bản được quan tâm bởi nhiều tác giả và luận án này giải quyết vấn đề kiểm chứng chúng đối với ôtômat thời gian.

Cho $\mathcal{A}$ = <S, s_0, Σ, X, E, I> là một ôtômat thời gian. Một **bất biến khoảng tuyến tính** (linear duration invariant - LDI) trên S là một tính chất được biểu diễn bởi một công thức $\mathcal{D}$ trong DC có dạng:

$$\mathcal{D}: A \le \ell \le B \Rightarrow \sum_{i=1}^{m} c_i \int s_i \le M ,$$

trong đó $s_i \in$ S; A, B, $c_i \neq 0$, M là các hằng số thực (B có thể ∞), $\int s_i$ kí hiệu cho độ dài khoảng thời gian xuất hiện của vị trí s_i, ℓ là độ dài của khoảng thời gian quan sát.

Trường hợp $A = 0$, $B = \infty$, phần điều kiện của công thức trên có thể được lược bỏ và ta thu được công thức mới

$$\mathcal{D}: \sum_{i=1}^{m} c_i \int s_i \le M$$

biểu diễn cho một tính chất được gọi là **tính chất khoảng tuyến tính** (linear duration properties - LDP).

Các công thức trên mô tả lớp các tính chất thời khoảng hay gặp trong thực tế. Ví dụ tính an toàn của hệ thống dưới dạng phát biểu "một trạng thái không an toàn s nào đó sẽ không bao giờ xảy ra trong hệ thống" có thể được biểu diễn bởi một công thức LDI dạng $\ell \ge 0 \Rightarrow \int s \le 0$. Hoặc tính cân bằng tương đối (relative fairness) của 2 tiến trình p_1 và p_2 có thể được mô tả bởi hai công thức LDI:

$$\ell \ge 0 \Rightarrow \int p_1 - \int p_2 \le 1 \text{ và } \ell \ge 0 \Rightarrow \int p_2 - \int p_1 \le 1.$$

Trong phần này của luận án chúng tôi cũng đã trình bày một ví dụ chi tiết về đặc tả hệ bếp ga (gaz burner) bằng DC.

Lấy σ = ($\mathcal{I}$, [b, e]) là một mô hình DC.

Kí hiệu $l(\sigma) = e - b$ và $\theta(\sigma) = \sum_{i=1}^{m} c_i \int s_i$, với $\int s_i$ được tính từ σ.

Định nghĩa 2.4. Cho $\mathcal{S}$ là hệ thời gian thực được xác định bởi tập mô hình $\mathcal{M}(\mathcal{S})$ và cho $\mathcal{D}$ là một công thức LDP (LDI). Khi đó:

- Mô hình σ = ($\mathcal{I}$, [b, e]) ∈ $\mathcal{M}(\mathcal{S})$ được gọi là thoả $\mathcal{D}$ (kí hiệu $\sigma \models \mathcal{D}$) nếu và chỉ nếu $\theta(\sigma) \le M$ với $\mathcal{D}$ là công thức LDP hoặc $A \le l(\sigma) \le B \Rightarrow \theta(\sigma) \le M$ với $\mathcal{D}$ là công thức LDI.
- Hệ thời gian thực $\mathcal{S}$ được gọi là thoả công thức $\mathcal{D}$ và được kí hiệu bởi $\mathcal{S} \models \mathcal{D}$ nếu và chỉ nếu $\sigma \models \mathcal{D}$, với mọi $\sigma \in \mathcal{M}(\mathcal{S})$.

Bài toán kiểm chứng mô hình trong luận án được phát biểu: Cho một ôtômat thời gian $\mathcal{A}$ = <S, s_0, Σ, X,E,I> được xác định bởi tập mô hình $\mathcal{M}(\mathcal{A})$, cho công thức LDP (LDI) $\mathcal{D}$ trên tập vị trí S. Hãy tìm một thủ tục để quyết định liệu $\mathcal{A} \models \mathcal{D}$, tức là liệu $\sigma \models \mathcal{D}$ với mọi $\sigma \in \mathcal{M}(\mathcal{A})$.

Chương 3. KIỂM CHỨNG MÔ HÌNH VỚI KỸ THUẬT QUI HOẠCH TUYẾN TÍNH

Trong chương này chúng tôi trình bày phương pháp kiểm chứng tính thỏa của lưới ôtômat thời gian thực hoạt động đồng bộ đối với công thức khoảng LDI, dựa trên việc giải các bài toán qui hoạch tuyến tính. Có thể tóm tắt phương pháp bởi các nét chính sau:

1. Biểu diễn hệ thống bằng *biểu thức chính quy thời gian*,
2. Đưa biểu thức chính quy thời gian bất kỳ về hợp của một số các *biểu thức chính quy thời gian hữu hạn*,
3. Đưa biểu thức chính quy thời gian hữu hạn về hợp của một số các *biểu thức chính quy thời gian đơn giản*,
4. Thiết lập tương ứng mỗi biểu thức chính quy thời gian đơn giản với một *bài toán qui hoạch tuyến tính*.
5. Dựa trên nghiệm của họ các bài toán qui hoạch tuyến tính đưa ra câu trả lời cho kết quả của bài toán kiểm chứng.

3.1 BIỂU THỨC CHÍNH QUY THỜI GIAN VÀ BÀI TOÁN QUI HOẠCH TUYẾN TÍNH

3.1.1. Biểu thức chính quy thời gian TRE

Biểu thức chính quy thời gian R (Timed Regular Expression - TRE) và tập trạng thái state(R) của nó được định nghĩa đệ qui như sau:

Định nghĩa 3.1 [TRE]

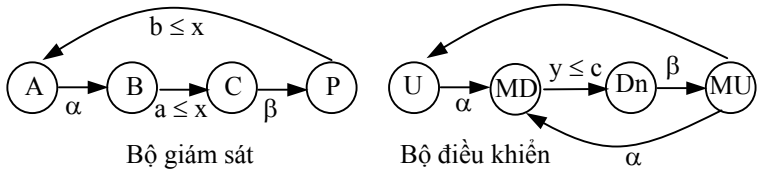
- ε là một TRE (được gọi là TRE rỗng) và state(ε) = ∅,
- đối với $s \in S$, với các số thực a, b bất kỳ, $0 \le a \le b$ (b có thể ∞), $(s, [a, b])$ là một TRE và state($(s, [a, b])$) = $\{s\}$,
- nếu R_1, R_2 là các TRE, thì R_1*, $R_1 \frown R_2$, $R_1 \oplus R_2$ là các TRE, và state(R_1*) = state(R_1); state($R_1 \frown R_2$) = state($R_1 \oplus R_2$) = state(R_1) ∪ state(R_2),
- nếu R_1, R_2 là các TRE, và state(R_1) ∩ state(R_2) = ∅, thì $R_1 \otimes R_2$ là một TRE và state($R_1 \otimes R_2$) = state(R_1) ∪ state(R_2).

Mỗi $(s, [a, b])$ được gọi là một nguyên thủy và để ngắn gọn ta chỉ viết s thay cho $(s, [0, \infty])$. Như thông thường, toán tử $\frown$ thể hiện phép nối (hay hợp), khi đó biểu thức $(s_1, [a_1, b_1]) \frown (s_2, [a_2, b_2])$ biểu thị hệ thống nằm tại s_1 một khoảng thời gian d_1 bị chặn bởi $[a_1, b_1]$ (tức $a_1 \le d_1 \le b_1$), sau đó hệ chuyển vị trí đến s_2, tại đây hệ thống có thể nằm lại một khoảng thời gian bị chặn bởi $[a_2, b_2]$. Toán tử ⊕ thể hiện phép hoặc, tức biểu thức $(s_1, [a_1,b_1]) \frown ((s_2, [a_2,b_2] \oplus (s_3, [a_3,b_3]))$ biểu thị hệ thống có thể chuyển từ s_1 đến s_2 hoặc từ s_1 đến s_3. Toán tử * thể hiện phép lặp, có nghĩa biểu thức $(s_1, [a_1, b_1])$* cho phép hệ thống chuyển vị trí về lại s_1 một số lần bất kỳ. Đặc biệt đối với hệ hoạt động song song toán tử ⊗ thể hiện phép hợp song song, tức biểu thức $(s_1, [a_1, b_1]) \otimes (s_2, [a_2, b_2])$ biểu thị thành phần thứ nhất của hệ thống chuyển vị trí đến s_1 đồng thời (cùng một thời điểm) với thành phần thứ hai chuyển vị trí đến s_2.

Định nghĩa 3.2 [Tập mô hình xác định bởi TRE]

- Mô hình $\sigma = (\mathcal{I}, [0, T]) \in \mathcal{M}(\varepsilon)$ nếu và chỉ nếu T = 0.
- Mô hình $\sigma = (\mathcal{I}, [0, T]) \in \mathcal{M}((s, [a, b]))$ nếu và chỉ nếu $a \le T \le b$ và $\forall t \in [0, T]$, $s_{\mathcal{I}}(t) = 1$, $\forall s' \ne s$, $s'_{\mathcal{I}}(t) = 0$.
- Mô hình $\sigma = (\mathcal{I}, [0, T]) \in \mathcal{M}(R_1 \frown R_2)$ nếu và chỉ nếu tồn tại $0 \le T' \le T$, $\sigma_1 = (\mathcal{I}_1, [0, T']) \in \mathcal{M}(R_1)$, $\sigma_2 = (\mathcal{I}_2, [0, T-T']) \in \mathcal{M}(R_2)$ sao cho $\forall s \in$ state(R_1) ∪ state(R_2), $s_{\mathcal{I}1}(t) = s_{\mathcal{I}}(t)$, $\forall t \in [0, T')$ và $s_{\mathcal{I}2}(t-T') = s_{\mathcal{I}}(t)$, $\forall t \in [T', T]$, và $\forall s' \notin$ state(R_1) ∪ state(R_2), $s'_{\mathcal{I}}(t) = 0, \forall t \in [0, T]$. Khi đó, có thể xem $\sigma = \sigma_1 \frown \sigma_2$.
- Mô hình $\sigma = (\mathcal{I}, [0, T]) \in \mathcal{M}(R_1 \otimes R_2)$ nếu và chỉ nếu có $\sigma_1 = (\mathcal{I}, [0, T]) \in \mathcal{M}(R_1)$, $\sigma_2 = (\mathcal{I}_2, [0, T]) \in \mathcal{M}(R_2)$ sao cho $\forall t \in [0, T]$, $s_{\mathcal{I}1}(t) = s_{\mathcal{I}}(t)$, $\forall s \in$ state(R_1) và $s_{\mathcal{I}2}(t) = s_{\mathcal{I}}(t)$, $\forall s \in$ state(R_2) và $s'_{\mathcal{I}}(t) = 0$, $\forall s' \notin$ state(R_1) ∪ state(R_2), khi đó có thể xem $\sigma_1 \otimes \sigma_2$ như mô hình σ.
- Mô hình $\sigma = (\mathcal{I}, [0, T]) \in \mathcal{M}(R_1 \oplus R_2)$ nếu và chỉ nếu $\sigma \in \mathcal{M}(R_1)$ hoặc $\sigma \in \mathcal{M}(R_2)$.
- Mô hình $\sigma = (\mathcal{I}, [0, T]) \in \mathcal{M}(R^*)$ nếu và chỉ nếu tồn tại số nguyên $k \ge 0$ sao cho $\sigma \in \mathcal{M}(R^k)$, ở đây $R^0 \triangleq \varepsilon$ và với $k > 0$, $R^k \triangleq R \frown R^k$. Hoặc, một cách tương đương, có các mô hình $\sigma_1, ..., \sigma_k \in \mathcal{M}(R)$ sao cho $\sigma = \sigma_1 \frown \sigma_2 \frown ... \frown \sigma_k$.

Như vậy mỗi TRE R xác định một hệ thống được cho bởi tập mô hình $\mathcal{M}(R)$, mỗi mô hình tương ứng với một thể hiện cụ thể của hệ thống trong khoảng quan sát từ 0 đến thời điểm T nào đó.



Hình 3.1. Hệ chắn tàu.

Ví dụ [Hệ chắn tàu]. Hình 3.1 biểu diễn một hệ thống điều khiển các thanh chắn chỗ giao nhau của đường sắt và đường bộ. Hệ gồm 2 ôtômat biểu diễn 2 thành phần là bộ giám sát dùng để phát hiện các đoàn tàu đi vào chỗ giao nhau và bộ điều khiển ra lệnh việc nâng hạ các thanh chắn. Hai thành phần này hoạt động đồng bộ, trong đó phép chuyển từ A đến B đồng bộ với phép chuyển từ U đến D và từ MU đến MD (nhãn α), phép chuyển từ C đến P đồng bộ với phép chuyển từ Dn đến MU (nhãn β). Các ràng buộc trên hai đồng hồ {x, y} được cho trong hình. Chi tiết cụ thể xin xem trong luận án.

Một TRE RCM dùng biểu diễn hệ có thể được thiết lập như sau:

RCM = (A⊗U) ∩
((B, [a,∞))∩ C⊗(MD, [0,c])∩Dn) ∩
((P, [b,∞))∩ A⊗(MU∩U ⊕ MU)))* ∩
(ε ⊕
(B⊕B∩C) ⊗ ((MD, [0,c])⊕(MD, [0,c])∩Dn) ⊕
(B ∩ C⊗(MD, [0,c]) ∩ Dn) ∩
((P ⊕ P∩A) ⊗ (MU ⊕ MU∩U)))

Một TRE R được gọi là thỏa công thức $\mathcal{D}$ ($R \vDash \mathcal{D}$) nếu $\sigma \vDash \mathcal{D}$, $\forall \sigma \in \mathcal{M}(R)$. Các TRE R_1 và R_2 được gọi là $\mathcal{D}$-tương đương (kí hiệu bởi $R_1 \equiv_{\mathcal{D}} R_2$) nếu $R_1 \vDash \mathcal{D}$ khi và chỉ khi $R_2 \vDash \mathcal{D}$.

3.1.2 Biểu thức TRE hữu hạn

Một TRE trong đó không chứa phép toán * được gọi là TRE hữu hạn và nếu không chứa cả phép toán ⊕ sẽ được gọi là TRE đơn giản.

Ví dụ, ((*s*, [1,5])∩(*u*, [1,7]))⊗((*v*$_1$, [3,10])⊕(*v*$_2$,[2,9])) là một TRE hữu hạn, và ((*s*, [1,5])∩(*u*, [1,7])) ⊗ (*v*, [3,10]) là một TRE đơn giản.

Có thể khẳng định: bất kỳ TRE hữu hạn R đều là $\mathcal{D}$-tương đương với hợp của một số TRE đơn giản R_i, tức $R \equiv_{\mathcal{D}} R_1 \oplus ... \oplus R_k$. Từ đó việc kiểm chứng một TRE hữu hạn R có thể đưa về kiểm chứng họ các TRE đơn giản R_i (định lý 3.1 và 3.2).

3.1.3 Xây dựng bài toán qui hoạch tuyến tính

Để kiểm chứng một TRE đơn giản R, chúng ta sẽ xây dựng bài toán qui hoạch tuyến tính sao cho thông qua nghiệm của bài toán này có thể xác định được kết quả của bài toán kiểm chứng. Có thể tóm tắt cách xây dựng bài toán qui hoạch tuyến tính này như sau: kết hợp mỗi nguyên thủy (*s*, [*a, b*]) của R với một biến thực *t*, ta có ràng buộc $a \le t \le b$. Bằng đệ qui theo định nghĩa của R ta xây dựng tiếp tập các ràng buộc cho mọi biến *t* (định nghĩa 3.6 của luận án), tập các ràng buộc tuyến tính này được kí hiệu bởi $\mathcal{C}$(R). Tương tự, xây dựng d_s(R) và l(R) biểu thị cho tổng khoảng thời gian xuất hiện của vị trí *s* và độ dài tối đa của các mô hình trong R.

Đặt $\theta(R) = \sum_{s \in S} c_s d_s(R)$ và $D(R) = \max\{\theta(R) \mid \mathcal{C}(R) \wedge A \le l(R) \le B\}$.

Định lý 3.3 dưới đây chỉ ra rằng kết quả của bài toán kiểm chứng có thể quyết định được thông qua giá trị của D(R).

Định lý 3.3. Đối với TRE đơn giản R, $R \vDash \mathcal{D}$ khi và chỉ khi $D(R) \le M$.

Ví dụ. Lấy R = ((*s*, [1,5])∩(*u*, [1,7]))⊗(*v*, [3, 10]) và $\mathcal{D} = 4 \le \ell \le 8 \Rightarrow 2\int s - \int v \le 5$. Lấy *x, y, z* là các biến kết hợp với các nguyên thuỷ (*s*, [1,5]), (*u*, [1,7]), (*v*, [3,10]). $R \vDash \mathcal{D}$ có thể được kiểm chứng bởi việc giải bài toán qui hoạch tuyến tính: Tìm max{2*x* − *z*} với các ràng buộc $1 \le x \le 5$, $1 \le y \le 7$, $3 \le z \le 10$, $z = x + y$, $4 \le z \le 8$ và kiểm tra liệu giá trị của nó có bé hơn hoặc bằng 5 ? Dễ thấy nghiệm của bài toán qui hoạch tuyến tính là $x = 5$, $y = 1$, $z = 6$ và giá trị lớn nhất của hàm mục tiêu là $4 < 5$. do vậy $R \vDash \mathcal{D}$.

3.2 CHUYỂN TRE VỀ HỢP CỦA CÁC TRE HỮU HẠN

3.2.1 Cận thời gian của biểu thức TRE

Với TRE R khác rỗng ($\mathcal{M}(R) \ne \varnothing$), gọi m(R), M(R) là các cận dưới và trên của độ dài các mô hình được xác định bởi R. R' được gọi là biểu thức con của R nếu nó xuất hiện tại một vị trí nào đó trong R. Khi đó độ dài các mô hình của R' ngoài việc bị chặn bởi m(R'), M(R'), nó còn bị chặn bởi m(R) và M(R) tạo thành các cận mới được kí hiệu bởi m(R', R) và M(R', R). Các đại lượng này có thể tính được một cách đệ qui thông qua các định nghĩa 3.7 và 3.8 trong luận án.

3.2.2 Khử phép toán * trong biểu thức TRE

Kí hiệu $\lfloor x \rfloor$ là phần nguyên của số thực *x*. Cho A là một TRE đơn giản không chứa TRE con rỗng (việc loại TRE rỗng ra khỏi A không ảnh hưởng đến $\mathcal{M}(\mathcal{A})$). Các định lý sau chỉ ra các trường hợp có thể thay A* trong R bởi một TRE đơn giản A' sao cho biểu thức nhận được R' là $\mathcal{D}$-tương đương với R. Bằng cách thay dần các TRE dạng A* như vậy trong R, cuối cùng ta nhận được TRE hữu hạn $R' \equiv_{\mathcal{D}} R$.

Định lý 3.4. Giả sử A là một TRE đơn giản với m(A) = 0. Lấy A' là TRE đạt được từ A bằng cách thay mỗi nguyên thuỷ (*s*, [0, *b*]) ($b > 0$ do qui ước A không chứa TRE con rỗng) của A bởi (*s*, [0, ∞)). Khi đó, bằng việc thay mỗi xuất hiện của A* trong TRE R bởi A', chúng ta đạt được TRE R' là $\mathcal{D}$-tương đương với R.

Định lý 3.5. Giả sử A là một TRE đơn giản với m(A) > 0. Lấy một xuất hiện của A* trong R mà đối với nó M(A*, R) < ∞ hoặc $B < \infty$ (*B* trong điều kiện $A \le \ell \le B$). Lấy $A' = \oplus_{\{i=0..k\}} A^i$, ở đây $k = \lfloor \min\{M(A^*, R), B\}/m(A) \rfloor + 1$. Khi đó bằng việc thay thế xuất hiện

của A* trong R bởi A', chúng ta đạt được một biểu thức mới R' là $\mathcal{D}$-tương đương với R.

Lấy R' là một biểu thức con của R. R' được gọi là nằm dưới ⊗ nếu có một biểu thức con dạng $R_1 \otimes R_2$ của R sao cho R' là một xuất hiện trong R_1 hoặc trong R_2. Dễ thấy nếu A* không nằm dưới ⊗, thì M(A*, R) = ∞. Cho A là một TRE đơn giản. Gọi maxθ(A) là giá trị lớn nhất của $\{\theta(\sigma) \mid \sigma \in \mathcal{M}(A)\}$ mà nó có thể tính được bằng việc giải bài toán qui hoạch tuyến tính: tìm giá trị lớn nhất của hàm mục tiêu θ(A) với tập các ràng buộc $\mathcal{C}$(A).

Định lý 3.6. Nếu $B = \infty$, và A* là một biểu thức con của R không xuất hiện dưới ⊗, ở đây A là một TRE đơn giản với m(A) > 0 thì:

- Trường hợp maxθ(A) ≤ 0, thay A* trong R bởi $A' = \oplus_{\{i = 0..k\}} A^i$, trong đó $k = \lfloor A/m(A) \rfloor + 1$ (A trong điều kiện $A \le \ell \le B$), chúng ta nhận được biểu thức R' là $\mathcal{D}$-tương đương với R.
- Trường hợp maxθ(A) > 0, kết luận $R \not\models \mathcal{D}$.

Tóm lại, từ các định lý 3.4, 3.5 và 3.6, chúng ta có thể phát hiện sớm trường hợp $R \not\models \mathcal{D}$ hoặc khử dần các * trong các biểu thức A* với A là biểu thức đơn giản trong các trường hợp sau:

- m(A) = 0,
- m(A) > 0, B hữu hạn hoặc M(A*, R) hữu hạn,
- m(A) > 0, B vô hạn và A* không xuất hiện dưới ⊗.

Trường hợp còn lại (m(A) > 0, B vô hạn và A* xuất hiện dưới ⊗) để khử * và kiểm tra tính rỗng của TRE R ($\mathcal{M}(R) = \varnothing$) là phức tạp hơn và được chúng tôi trình bày riêng trong tiểu mục 3.2.4.

3.2.3 Kiểm chứng tính an toàn của hệ chắn tàu

Trong tiểu mục này của luận án chúng tôi đưa ra một ví dụ về áp dụng phương pháp đã mô tả để kiểm chứng TRE RCM biểu diễn hệ chắn tàu được cho trong hình 3.1.

3.2.4 Kiểm tra tính rỗng và khử * xuất hiện dưới ⊗

Trong tiểu mục này chúng tôi trình bày phương pháp kiểm tra tính rỗng của biểu thức R và khử * xuất hiện dưới ⊗ cho vài trường hợp riêng đặc biệt và chỉ ra rằng bài toán có thể giải được bởi kỹ thuật qui hoạch hỗn hợp tuyến tính và nguyên.

3.3 CÁC CÔNG TRÌNH LIÊN QUAN VÀ NHẬN XÉT VỀ PHƯƠNG PHÁP

3.3.1 Các công trình liên quan

Để kiểm chứng LDI, một số tác giả (Z. Chaochen, D.V. Hung, L. XuanDong, …) đã sử dụng các phép toán nối (⌢), hợp (⊕) và lặp (*) để tạo các biểu thức chính quy thời gian mô tả các lớp con của ôtômat. Trong luận án này chúng tôi đưa thêm phép toán "hợp song song" (⊗) để biểu diễn lưới ôtômat thời gian thực hoạt động đồng bộ, trên cơ sở đó thiết lập bài toán qui hoạch tuyến tính. Đặc điểm chung của phương pháp này là phải giải một số rất lớn các bài toán qui hoạch. Trong một số công trình khác, để giảm bớt độ phức tạp, các tác giả đã xét tính chất đơn giản hơn với dáng điệu của hệ thống chỉ đi qua (hoặc không) một dãy trạng thái cố định cho trước. Điển hình như tính chất "khoảng tuần tự" (Temporal Duration Property – Li Yong and D.V. Hung) và chúng tôi cũng đã xây dựng thuật toán để kiểm chứng tính chất này cho lưới ôtômat thời gian.

3.3.2 Vài nhận xét về phương pháp

Dễ dàng thiết lập được bài toán qui hoạch tuyến tính từ một TRE đơn giản và có thể sử dụng để giải lớp các công thức tuyến tính tổng quát. Tuy nhiên, phương pháp này chỉ thích hợp với bài toán kiểm chứng trên các đoạn quan sát có điểm bắt đầu và kết thúc trùng với thời điểm hệ chuyển vị trí. Ngoài ra số lượng bài toán qui hoạch tuyến tính phải giải là rất lớn và với lớp công thức phức tạp, việc rút gọn biểu thức về dạng đơn giản là khó.

Chương 4. KỸ THUẬT RỜI RẠC HÓA VÀ DUYỆT ĐỒ THỊ ĐẠT ĐƯỢC

Một nhược điểm trong phương pháp qui hoạch tuyến tính là số bài toán qui hoạch phải giải là quá lớn. Do đó, một số tác giả đã quay lại áp dụng phương pháp truyền thống trong kiểm chứng tính chất thời điểm là duyệt đồ thị phân vùng cho các tính chất thời khoảng nhưng chỉ xét với các đồ thị vùng nguyên. Như đã biết một công thức đúng trong mô hình thời gian nguyên chưa hẳn đã đúng trong mô hình thời gian thực. Tuy nhiên, điều này vẫn xảy ra với một số công thức, các công thức như vậy được gọi là rời rạc hóa được. Với nhận

xét này các tác giả Jianhua và Dang Van Hung (JH) đã chứng minh LDP là rời rạc hóa được đối với các quan sát xuất phát từ vị trí ban đầu của ôtômat (tương đương với lớp mô hình $\mathcal{M}_0(\mathcal{A})$ – xem tiểu mục 4.1.1), từ đó thiết kế thuật toán để kiểm chứng LDP đối với lớp mô hình này trên cơ sở duyệt đồ thị phân vùng nguyên. Một nhược điểm khác của phương pháp qui hoạch tuyến tính và cũng là nhược điểm của phương pháp duyệt đồ thị phân vùng là các thời điểm bắt đầu và kết thúc của đoạn quan sát phải trùng với các thời điểm chuyển vị trí của ôtômat (tương đương với lớp mô hình $\mathcal{M}_{uv}(\mathcal{A})$). Điều này là đủ đối với kiểm chứng tính chất thời điểm nhưng với tính chất thời khoảng cần phải xét rộng hơn với các đoạn thời gian quan sát bất kỳ, tức phải xét trên toàn bộ tập mô hình $\mathcal{M}(\mathcal{A})$.

Để khắc phục nhược điểm này, dựa trên tư tưởng rời rạc hóa được của JH, kết hợp với khái niệm ε-nguyên hóa của Henzinger và Kesten chúng tôi chứng minh LDP và LDI là rời rạc hóa được theo ngữ nghĩa đầy đủ (trên toàn bộ tập mô hình $\mathcal{M}(\mathcal{A})$). Từ đó, tính rời rạc hóa được sử dụng trong luận án này mang hai ý nghĩa: rút gọn đồ thị vùng thành đồ thị vùng nguyên để giảm độ phức tạp (như các tác giả JH đã sử dụng) và "rời rạc hóa" đồ thị vùng nguyên để thiết kế thuật toán kiểm chứng LDI. Nói cách khác, tính rời rạc hoá được không chỉ để giảm độ phức tạp mà còn đóng vai trò quan trọng trong việc chứng minh tính quyết định được của LDI.

4.1 TÍNH RỜI RẠC HÓA ĐƯỢC VÀ ĐỒ THỊ VÙNG NGUYÊN

4.1.1 Tập mô hình DC của ôtômat thời gian

Tổng quát, một hệ thời gian thực bất kỳ là tương ứng với một tập các mô hình trong DC. Cũng tương tự, một hệ thống được biểu diễn bởi ôtômat thời gian $\mathcal{A}$ xác định một tập mô hình $\mathcal{M}(\mathcal{A})$ nào đó. Cụ thể, xét dáng điệu $\rho = (s_0, t_0)(s_1, t_1)(s_2, t_2)$ ... của ôtômat thời gian $\mathcal{A} = <S, s_0, \Sigma, X, I, E>$, tức dãy các vị trí $\bar{s} = s_0, s_1, s_2, ...$ và thời gian $\bar{t} = t_0, t_1, t_2, ...$ thoả: $t_0 = 0$, $t_i \le t_{i+1}$, $\forall i \ge 0$, và $\forall i \ge 1$, $\exists <s_{i-1}, \varphi_i, a_i, \lambda_i, s_i>$ sao cho nếu gọi ν_{i-1}, ν_i là các thể hiện đồng hồ tương ứng tại các thời điểm t_{i-1}, t_i thì $(s_{i-1}, \nu_{i-1}) \xrightarrow{\delta_i, a_i} (s_i, \nu_i)$, trong đó $\delta_i = t_i - t_{i-1}$. Khi đó dễ thấy $\bar{s}$, $\bar{t}$ tương ứng với một thể hiện $\mathcal{I}$ trong DC mà nó được định nghĩa bởi: $s_{i\mathcal{I}}(t) = 1$ nếu và chỉ nếu $t \in [t_i, t_{i+1})$, $\forall i \ge 0$. Ngược lại

một thể hiện $\mathcal{I}$ bất kỳ xác định dãy $\bar{s}$, $\bar{t}$ thoả các tính chất trên là một biểu diễn dáng điệu nào đó của ôtômat. Từ đó một quan sát hoạt động của ôtômat trong đoạn thời gian $[b, e]$ ($b \le e < \infty$) sẽ là một mô hình $\sigma = (\mathcal{I}, [b, e])$ trong đó $\mathcal{I}$ là một thể hiện hệ thống biểu diễn dáng điệu của ôtômat. Ta cũng kí hiệu σ bởi $(\bar{s}, \bar{t}, [b, e])$ và tập các σ được kí hiệu bởi $\mathcal{M}(\mathcal{A})$. Các tập mô hình con của $\mathcal{A}$ được kí hiệu:

- $\mathcal{M}_0(\mathcal{A}) = \{\sigma = (\bar{s}, \bar{t}, [0, T]) \in \mathcal{M}(\mathcal{A}), \forall T \ge 0\}$,
- $\mathcal{M}_{uv}(\mathcal{A}) = \{\sigma = (\bar{s}, \bar{t}, [t_u, t_v]) \in \mathcal{M}(\mathcal{A}), \forall t_u \le t_v \in \bar{t}\}$,
- $\mathcal{M}_I(\mathcal{A}) = \{\sigma=(\bar{s}, \bar{t}, [b,e]) \in \mathcal{M}(\mathcal{A}) \mid t_i, b, e \in \mathbb{N}, (t_i \in \bar{t}, \forall i \ge 0)\}$.

Nói cách khác, $\mathcal{M}_0(\mathcal{A})$ là tập mô hình biểu thị các quan sát xuất phát từ thời điểm ban đầu (0) của ôtômat. $\mathcal{M}_{uv}(\mathcal{A})$ chứa các mô hình mà thời điểm bắt đầu và kết thúc của quan sát trùng với các thời điểm chuyển vị trí của ôtômat. Điều này có nghĩa nếu $\sigma = (\bar{s}, \bar{t}, [b, e]) \in \mathcal{M}_{uv}(\mathcal{A})$, ở đây $\bar{s} = s_0, s_1, ..., s_m, ...$ và $\bar{t} = t_0, t_1, ..., t_m, ...$ thì có tồn tại các chỉ số u, v sao cho $b = t_u$ và $e = t_v$. Cuối cùng một mô hình của $\mathcal{M}_I(\mathcal{A})$ biểu diễn cho các quan sát trên các dáng điệu nguyên của ôtômat, tức các t_i và các cận quan sát b, e là những số nguyên. Các mô hình (đường chạy, dáng điệu) như vậy được gọi là các mô hình (đường chạy, dáng điệu) nguyên.

4.1.2 Khái niệm ε-nguyên hoá và một vài tính chất

Định nghĩa 4.1. Cho một số thực dương x và ε, $(0 \le \varepsilon < 1)$. x_ε là một số nguyên nhận được từ x như sau: $x_\varepsilon = \lfloor x \rfloor$ nếu phần thập phân của x là bé hơn hoặc bằng ε và $x_\varepsilon = \lceil x \rceil$ nếu ngược lại.

Khi đó, x_ε nhận được từ x bằng cách làm tròn xuống hoặc lên phụ thuộc vào ε và phần dư của x, và được gọi là ε-nguyên hoá của x. Ví dụ nếu $x = 3.72$ thì $x_{0.5} = 4$ và $x_{0.8} = 3$.

Một số tính chất quan trọng của khái niệm ε-nguyên hoá được chúng tôi chứng minh trong các bổ đề 4.1 – 4.4, trong đó bổ đề 4.4 khẳng định: với mọi $0 \le \varepsilon < 1$ và với mọi mô hình (thực) $\sigma \in \mathcal{M}(\mathcal{A})$, ε-nguyên hóa của σ (tức ε-nguyên hóa mọi thời điểm chuyển vị trí t_i và các cận quan sát b, e) cũng cho ta một mô hình (nguyên) σ_ε của $\mathcal{A}$. Trên cơ sở này trong tiểu mục 4.1.3 tiếp theo sau đây sẽ chứng minh tính rời rạc hóa được của các công thức LDP, LDI.

4.1.3 Tính rời rạc hóa được của các công thức LDP, LDI

Định nghĩa 4.2. Cho ôtômat thời gian $\mathcal{A}$ và công thức $\mathcal{D}$. $\mathcal{D}$ được gọi là *rời rạc hoá được* đối với $\mathcal{A}$ nếu $\mathcal{M}(\mathcal{A}) \models \mathcal{D} \Leftrightarrow \mathcal{M}_I(\mathcal{A}) \models \mathcal{D}$, tức $\mathcal{A}$ thoả $\mathcal{D}$ khi và chỉ khi mọi mô hình nguyên của $\mathcal{A}$ thoả $\mathcal{D}$.

Định lý 4.1. Mọi tính chất khoảng tuyến tính và bất biến khoảng tuyến tính $\mathcal{D}$ là rời rạc hóa được đối với ôtômat thời gian $\mathcal{A}$.

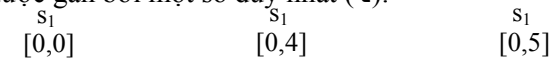
Định lý 4.1 được chứng minh bằng phản chứng: nếu $\exists\sigma \in \mathcal{M}(\mathcal{A})$ sao cho $\sigma \not\models \mathcal{D}$ thì $\exists\varepsilon \in [0,1)$ sao cho $\sigma_\varepsilon \in \mathcal{M}_I(\mathcal{A})$ và $\sigma_\varepsilon \not\models \mathcal{D}$ (chiều ngược lại là hiển nhiên vì $\mathcal{M}_I(\mathcal{A}) \subseteq \mathcal{M}(\mathcal{A})$).

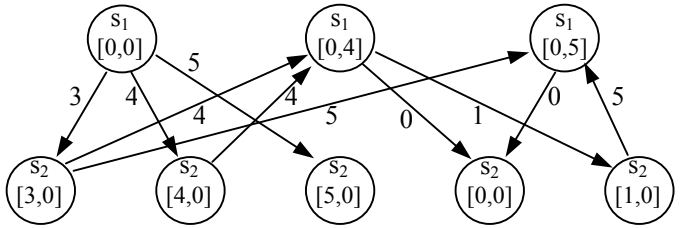
4.1.4 Đồ thị vùng đạt được nguyên

Một đồ thị vùng nói chung chỉ phản ánh tính đạt được mà không phản ánh được thời gian của các phép chuyển vị trí. Tuy nhiên, với đồ thị vùng nguyên các cận bé và lớn nhất của thời gian chuyển vẫn có thể tính được. Trong phần này chúng tôi đưa ra thuật toán xây dựng đồ thị vùng nguyên $\mathcal{RG}$ và các cận thời gian như vậy. Từ $\mathcal{RG}$ chúng tôi tiếp tục xây dựng các đồ thị trọng số tương ứng để kiểm chứng LDP và LDI. Thuật toán xây dựng đồ thị vùng nguyên $\mathcal{RG} = (\mathsf{V}, \mathsf{E})$ của ôtômat thời gian $\mathcal{A}$ có thể tóm tắt như sau.

Một đỉnh $\mathsf{v} \in \mathsf{V}$ là một vùng trạng thái $<s, \pi>$, trong đó π là vùng đồng hồ, s được gọi là vị trí đặc trưng của v (và kí hiệu $s \in \mathsf{v}$). Vùng $<s', \pi'>$ được gọi là hậu duệ của $<s, \pi>$ nếu $\exists\nu \in \pi, \exists\nu' \in \pi', \exists d \in \mathbb{N}$ và $\exists e = <s, \varphi, a, \lambda, s'>$ sao cho $(s, \nu) \xrightarrow{d,a} (s', \nu')$. Nếu $<s, \pi> \in \mathsf{V}$ và $<s', \pi'>$ là hậu duệ của $<s, \pi>$ thì $(<s, \pi>, <s', \pi'>)$ là một cung trong E. Ngoài ra, mỗi cung e được gán nhãn là đoạn $[\mathsf{l}(\mathsf{e}), \mathsf{u}(\mathsf{e})]$, ở đây $\mathsf{l}(\mathsf{e})$ và $\mathsf{u}(\mathsf{e})$ biểu thị thời gian tối thiểu và tối đa mà ôtômat có thể nằm lại tại vị trí s trước khi chuyển đến s'. Xuất phát từ đỉnh ban đầu $<s_0, \pi_0> \in \mathsf{V}$, ta có thể xây dựng hoàn toàn $\mathcal{RG}$ bằng một thuật toán chi tiết được cho trong bảng 4.1 của luận án.

Một tính chất quan trọng của đồ thị vùng nguyên là mỗi đường chạy của ôtômat sẽ tương ứng với một đường đi trong đồ thị sao cho tổng thời gian trên đường chạy luôn bị chặn bởi tổng các cận l và u trên đường đi này. Hình 4.4 dưới đây minh họa đồ thị vùng đạt được nguyên của ôtômat được cho trong hình 2.2. Trong đồ thị này, mọi cung đều có cận dưới và cận trên bằng nhau ($\mathsf{l} = \mathsf{u}$) nên nhãn của chúng chỉ được gán bởi một số duy nhất (l).





Hình 4.4. Đồ thị vùng đạt được nguyên $\mathcal{RG}$

4.2 KIỂM CHỨNG CÔNG THỨC LDP

4.2.1 Tính tương đương của $\mathcal{M}(\mathcal{A})$ và $\mathcal{M}_{uv}(\mathcal{A})$ đối với LDP

Thuật toán của JH có thể dễ dàng được mở rộng để áp dụng cho lớp mô hình $\mathcal{M}_{uv}(\mathcal{A})$ (ví dụ có thể áp dụng thuật toán trên n lần với n là số đỉnh của đồ thị, mỗi lần thực hiện thuật toán với một đỉnh khác nhau đóng vai trò như đỉnh xuất phát của đồ thị). Việc mở rộng hơn nữa kết quả trên (đến lớp mô hình $\mathcal{M}(\mathcal{A})$) là không hiển nhiên. Tuy nhiên, trong định lý 4.2 sau đây chúng tôi chỉ ra một kết quả khá thú vị đó là kết quả kiểm chứng trên hai tập mô hình $\mathcal{M}(\mathcal{A})$ và $\mathcal{M}_{uv}(\mathcal{A})$ là như nhau đối với công thức LDP.

Định lý 4.2. Cho ôtômat thời gian $\mathcal{A}$ và công thức LDP $\mathcal{D}$. Khi đó: $\mathcal{M}(\mathcal{A}) \models \mathcal{D} \Leftrightarrow \mathcal{M}_{uv}(\mathcal{A}) \models \mathcal{D}$.

Như vậy để kiểm chứng tính thỏa của $\mathcal{D}$ đối với ôtômat $\mathcal{A}$ ta chỉ cần kiểm chứng trên tập mô hình $\mathcal{M}_{uv}(\mathcal{A})$ và có thể áp dụng thuật toán của JH (như đã nêu trên). Tuy nhiên để giảm độ phức tạp chúng tôi đề nghị thuật toán kiểm chứng trên cơ sở thuật toán Warshall-Floyd để tìm đường đi có trọng số lớn nhất giữa mọi cặp đỉnh của đồ thị như được trình bày trong tiểu mục 4.2.2 dưới đây.

4.2.2 Đồ thị trọng số G phục vụ kiểm chứng LDP

Cho một ôtômat thời gian $\mathcal{A}$, một đồ thị vùng đạt được $\mathcal{RG}$ của $\mathcal{A}$ và công thức LDP: $\mathcal{D} = \sum_{i=1}^{m} c_i \int s_i \le M$.

Định nghĩa 4.5. Đồ thị trọng số $\mathsf{G} = (\mathsf{V}, \mathsf{E}, \omega)$ là đồ thị vùng đạt được nguyên $\mathcal{RG}$ với hàm trọng số cung ω: $\mathsf{E} \mapsto \mathbb{R}$ xác định như sau:

với mỗi cung $e = (v, v') \in E$ và $s \in v$, nếu $c_s < 0$, lấy $\omega(e) = c_s.l(e)$ ngược lại, lấy $\omega(e) = c_s.u(e)$. Dễ thấy $\omega(e)$ biểu thị giá trị lớn nhất của biểu thức $c_s \int s$ với $s \in v$.

Với $\omega(e)$, tính thỏa của một đường đi trong G và từ đó của tập đường đi $\mathcal{P}(G)$ đối với công thức $\mathcal{D}$ cũng được định nghĩa trong định nghĩa 4.6 của luận án. Từ đó $\mathcal{D}$ được kiểm chứng thông qua G bởi:

Định lý 4.3. $\mathcal{M}_{uv}(\mathcal{A}) \models \mathcal{D} \Leftrightarrow \mathcal{P}(G) \models \mathcal{D}$.

4.2.3 Thuật toán kiểm chứng LDP

Kết hợp các định lý 4.2, 4.3, thuật toán kiểm chứng LDP đối với $\mathcal{M}(\mathcal{A})$ được tiến hành trên đồ thị trọng số G như sau:

- Nếu G có cung với trọng số vô hạn hoặc khuyên với trọng số dương, thuật toán dừng với kết luận $G \not\models \mathcal{D}$ (từ đó $\mathcal{A} \not\models \mathcal{D}$).
- Loại bỏ các khuyên với trọng số âm ra khỏi đồ thị.
- Tìm đường đi có trọng số lớn nhất giữa mọi cặp đỉnh (bằng thuật toán Floyd-Warshall).
- So sánh trọng số này với M để kết luận tính thỏa của đồ thị G cũng là tính thỏa của ôtômat $\mathcal{A}$ đối với $\mathcal{D}$.

4.3 KIỂM CHỨNG CÔNG THỨC LDI

4.3.1 Quan hệ giữa lớp mô hình $\mathcal{M}_{uv}(\mathcal{A})$ và đồ thị đạt được $\mathcal{RG}$ hướng tới LDI

Chúng tôi chỉ ra rằng một kết quả như định lý 4.2 đối với LDP là không xảy ra đối với LDI. Tuy nhiên, bằng định nghĩa *thể hiện trọng số* của một đường đi trong đồ thị $\mathcal{RG}$ chúng tôi chứng minh rằng tập các thể hiện trọng số nguyên là "LDI-tương đương" với tập các mô hình $\mathcal{M}_{uv}(\mathcal{A})$. Trên cơ sở đó, trong tiểu mục 4.3.2 dưới đây chúng tôi tiếp tục "rời rạc hóa" $\mathcal{RG}$ thành đồ thị trọng số G (với trọng số 0/1) mà tập đường đi $\mathcal{P}(G)$ của nó là cũng "LDI-tương đương" với tập mô hình nguyên $\mathcal{M}_I(\mathcal{A})$. Từ đó, kết hợp với tính rời rạc hoá được của LDI, việc kiểm chứng tính thỏa của công thức LDI $\mathcal{D}$ đối với $\mathcal{M}(\mathcal{A})$ có thể đưa về kiểm chứng tính thỏa của $\mathcal{D}$ đối với tập đường đi $\mathcal{P}(G)$.

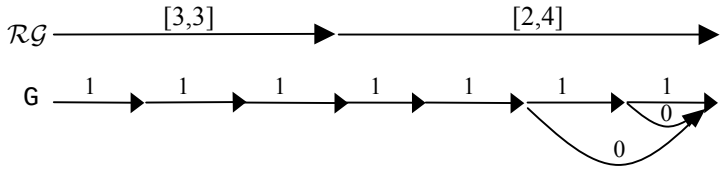
4.3.2 Đồ thị trọng số G phục vụ kiểm chứng LDI

Đồ thị trọng số $G = (V, E, \omega)$ được xây dựng từ $\mathcal{RG}$ bởi tư tưởng cơ bản sau: Đối với mỗi cung $e = (v_i, v_j, [l_{ij}, u_{ij}])$ (của $\mathcal{RG}$) chia e

thành u_{ij} cung con với trọng số 1 bởi $u_{ij} - 1$ đỉnh con. Từ đỉnh con thứ l_{ij} đến đỉnh con cuối cùng bổ sung các cung nối các đỉnh con này với v_j và lấy trọng số 0. Như vậy, mỗi cung $(v_i, v_j, [l_{ij}, u_{ij}])$ sẽ được thay thế bởi một họ gồm $u_{ij} - l_{ij} + 1$ đường đi trong G nối v_i và v_j với độ dài lần lượt bằng l_{ij}, $l_{ij} + 1$, ..., u_{ij}. Ngoài trọng số cung, mỗi đỉnh v của G cũng được gán một trọng số $\omega(v)$ là hệ số c_s trong công thức LDI với s là vị trí đặc trưng của v. Trên cơ sở này chúng ta có thể định nghĩa độ dài $l(p)$ và giá $\theta(p)$ của đường đi p như tổng độ dài của các cung và tổng các tích trọng số đỉnh với độ dài cung trong đường đi p. Từ đó một đường đi p là thỏa công thức LDI $\mathcal{D}$ nếu và chỉ nếu $A \le l(p) \le B \Rightarrow \theta(p) \le M$, và định lý 4.4 dưới đây cho phép đưa bài toán kiểm chứng LDI đối với ôtômat $\mathcal{A}$ về bài toán kiểm chứng LDI đối với đồ thị G.

Định lý 4.4. $\mathcal{M}_I(\mathcal{A}) \models \mathcal{D} \Leftrightarrow \mathcal{P}(G) \models \mathcal{D}$.

Hình 4.6 là một minh hoạ cho G được xây dựng từ đồ thị $\mathcal{RG}$ đơn giản với 2 cung



Hình 4.6. Đồ thị trọng số G từ đồ thị vùng đạt được $\mathcal{RG}$

4.3.3. Thuật toán kiểm chứng LDI

Trong tiểu mục này chúng tôi đưa ra thuật toán duyệt đồ thị G để kiểm chứng công thức LDI $\mathcal{D}$. Tư tưởng cơ bản của thuật toán là sử dụng kỹ thuật quay lui tìm các đường đi với độ dài nằm trong đoạn $[A, B]$. Với mỗi đường tìm được, so sánh giá của nó với M để quyết định tính thỏa của đường đi này đối với $\mathcal{D}$. Ngoài ra nếu phát hiện chu trình dương (tức chu trình với giá lớn hơn 0) có thể kết luận đường đi và do đó $\mathcal{P}(G) \not\models \mathcal{D}$. Kết quả kiểm chứng này cũng cung cấp câu trả lời kiểm chứng đối với lớp mô hình $\mathcal{M}_I(\mathcal{A})$ và do định lý 4.1 cũng là câu trả lời cho kiểm chứng $\mathcal{D}$ đối với lớp mô hình $\mathcal{M}(\mathcal{A})$.

KẾT LUẬN

Trong luận án này chúng tôi đã trình bày bài toán kiểm chứng mô hình đối với một số lớp tính chất biểu diễn được bằng các công thức khoảng tuyến tính trong lôgic khoảng. Một cách trực giác có thể thấy việc kiểm chứng các tính chất khoảng là khó hơn rất nhiều so với tính chất thời điểm bởi hai lý do: thứ nhất đối với tính chất thời điểm chỉ cần tiến hành kiểm chứng trên các quan sát bắt đầu và kết thúc tại các thời điểm hệ chuyển trạng thái, trong khi ngữ nghĩa của các tính chất thời khoảng đòi hỏi các quan sát phải được bắt đầu và kết thúc tại những thời điểm bất kỳ. Lý do thứ hai đến từ việc phải tính khoảng thời gian xuất hiện của các vị trí, do vậy cần quan tâm đến lịch sử của các vị trí này trong dáng điệu của ôtômat. Mặt khác thời gian được lấy trên tập trù mật $\mathbb{R}$ nên việc tính toán chúng cũng trở nên phức tạp hơn. Điểm khác biệt này lý giải vì sao phần lớn công thức khoảng là không quyết định được và các kết quả kiểm chứng tính chất thời khoảng cho đến hiện nay vẫn còn khiêm tốn. Hầu như các thuật toán được xây dựng thường chỉ hạn chế trên các lớp con của ôtômat thời gian và các tính chất tuyến tính trong DC, cuối cùng thường là chúng có độ phức tạp rất cao. Có thể kể đến một số kết quả đã có như:

- Kiểm chứng công thức hội của các LDP hệ số nguyên đối với lớp ôtômat thời gian hữu hạn (Finite Timed Automata) bằng phương pháp giải bài toán qui hoạch tuyến tính nguyên.
- Kiểm chứng công thức LDI đối với các lớp con của ôtômat thời gian bằng phương pháp giải bài toán qui hoạch tuyến tính.
- Kiểm chứng công thức LDP, TDP đối với lớp ôtômat thời gian bằng phương pháp duyệt đồ thị vùng nguyên.

Các kết quả trên hoặc chỉ xét với các lớp con của ôtômat thời gian hoặc với ôtômat thời gian tổng quát nhưng về ngữ nghĩa của công thức khoảng vẫn chỉ được xét trên lớp mô hình $\mathcal{M}_0(\mathcal{A})$.

Từ đó, luận án này quan tâm đến lớp hệ thống tổng quát hơn đó là ôtômat thời gian và ngữ nghĩa của công thức được xét trên tập mô hình $\mathcal{M}(\mathcal{A})$. Trên cơ sở đó luận án đã đạt được một số kết quả sau:

- Đề xuất toán tử $\otimes$ dùng để biểu diễn tính đồng bộ hoá của hệ thống (định nghĩa 3.1 trang 29, định nghĩa 3.2 trang 32) thông qua biểu thức chính quy thời gian TRE.

- Trên cơ sở biểu thức chính quy thời gian, đề nghị thuật toán kiểm chứng lớp hệ thống biểu diễn được bởi hợp song song các ôtômat thời gian thực bằng phương pháp đưa về giải bài toán qui hoạch tuyến tính (định lý 3.3 trang 36). Phân tích ưu khuyết điểm của phương pháp.
- Chứng minh công thức LDP, LDI là rời rạc hoá được (định lý 4.1 trang 60).
- Mở rộng đồ thị đạt được nguyên của ôtômat thời gian bằng cách trang bị các cận thời gian cho đồ thị. Trên cơ sở đó xây dựng đồ thị trọng số phục vụ kiểm chứng công thức LDP (thuật toán **Checking-LDP** trang 71).
- Xây dựng đồ thị trọng số là "rời rạc hóa" của đồ thị vùng nguyên phục vụ kiểm chứng công thức LDI (thuật toán **Traverse(`vstart`)** và **Checking-LDI** trang 83).

Ngoài những kết quả đã đạt được, trong thời gian tới đề tài của luận án có thể được tiếp tục với các hướng nghiên cứu sau:

1. Giảm độ phức tạp của thuật toán kiểm chứng LDI bằng cách tìm thêm các mối quan hệ tương đương giữa các lớp mô hình.
2. Cải tiến thuật toán kiểm chứng LDI trên cơ sở nhận xét đồ thị phục vụ kiểm chứng rất thưa cung (và chỉ có trọng số 0/1).
3. Mở rộng phương pháp kiểm chứng trên cho một số công thức rời rạc hoá được có liên quan đến công thức khoảng tuyến tính.
4. Xây dựng bộ kiểm chứng mô hình hoàn chỉnh cho LDP, LDI.

PHỤ LỤC A. Bộ kiểm chứng mô hình LDP, LDI

Một chương trình kiểm chứng đơn giản được viết bằng ngôn ngữ C++ thể hiện các thuật toán đưa ra trong luận án. Chương trình đã chạy và cho kết quả trên các ví dụ về ôtômat cho trong hình 2.2 và hệ chắn tàu trong hình 3.1. Với mỗi trường hợp chúng tôi cho chương trình chạy với một bộ dữ liệu cụ thể và đưa ra các kết quả chi tiết, ngoài ra phụ lục cũng trình bày kết quả cuối cùng (thỏa hay không thỏa) khi cho chương trình chạy với 20 bộ dữ liệu sinh ngẫu nhiên.

ꕥ o ꕥ

NHỮNG CÔNG TRÌNH CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. D.V. Hung and P.H. Thai (1998), "On Checking Parallel Real-Time Systems for Linear Duration Invariants", In Proceedings of *International Symposium on Software Engineering for Parallel and Distributed Systems,* Kyoto, Japan, April 1998. Bernd Kramer, Naoshi Uchihira, Peter Croll, Stefano Russo (eds.), IEEE Computer Society Press, pages 61-71.

2. Phạm Hồng Thái (2002), "Kiểm tra mô hình lưới ôtômat thời gian", Kỷ yếu hội thảo quốc gia: *Một số vấn đề chọn lọc của Công nghệ thông tin*. Nha Trang, 6-2002, NXB Khoa học kỹ thuật. Hà Nội, 9-2003, trang 278-287.

3. Pham Hong Thai (2003), "Checking Parallel Real-Time Systems for Temporal Duration Properties by Linear Programming", *Journal of Sciences, VNU*, Vol. 19, No. 4, 2003. pp. 49-62.

4. Phạm Hồng Thái, Mai Tuấn Linh (2003), "Khảo sát tính rời rạc hoá được của bất biến khoảng tuyến tính", Kỷ yếu hội thảo quốc gia: *Một số vấn đề chọn lọc của Công nghệ thông tin*, Thái Nguyên, 8-2003, NXB Khoa học kỹ thuật. Hà Nội, 1-2005, trang 364-374.

5. Pham Hong Thai (2004), "On Discretisable Formulas in Duration Calculus", *Journal of Sciences, VNU*, Vol. 20, No. 1, 2004, pp. 53-66.

6. Phạm Hồng Thái, Mai Tuấn Linh (2004), "Về bài toán kiểm chứng tính chất khoảng tuyến tính", Báo cáo tại hội thảo quốc gia: *Một số vấn đề chọn lọc của Công nghệ thông tin*, Đà nẵng, 8-2004.

7. Pham Hong Thai and Dang Van Hung (2004), "Verifying Linear Duration Constrains of Timed Automata". In the proceedings of *First International Colloqium on Theoretical Aspects of Computing ICTAC04*, Guiyang, China, 22-24, September 2004, LNCS 3407, Springer-Verlag, Feb. 2005, pp. 295-309.