ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN ĐỨC THỌ

KIỂM CHỨNG TỰ ĐỘNG CÁC HỆ THỜI GIAN THỰC XÁC SUẤT

Ngành: Công nghệ thông tin

Chuyên ngành: Kỹ thuật phần mềm

Mã số:

LUẬN VĂN THẠC SỸ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TIẾN SỸ ĐẶNG VĂN HƯNG

Hà Nội - 2016

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu do tôi tìm hiểu, nghiên cứu, tham khảo và tổng hợp từ các tài liệu nghiên cứu trước đây và làm theo hướng dẫn của người hướng dẫn khoa học. Phần nội dung đóng góp của luận văn do tôi thực hiện.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác, các nội dung được trích dẫn đã có tham chiếu đầy đủ.

Tôi xin chịu hoàn toàn trách nhiệm về lời cam đoan của mình. Nếu có điều gì sai trái, tôi xin chịu mọi hình thức kỷ luật theo quy định của nhà trường.

Tác giả

Nguyễn Đức Thọ

LỜI CẢM ƠN

Đầu tiên tôi xin gửi lời cảm ơn sâu sắc tới thầy TS.Đặng Văn Hưng, Bộ môn Kỹ thuật Phần mềm, Khoa Công nghệ Thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, người đã định hướng đề tài và tận tình hướng dẫn, chỉ bảo cho tôi trong suốt quá trình thực hiện luận văn tốt nghiệp này.

Tôi cũng xin trân trọng cảm ơn các thầy cô trong Khoa Công nghệ Thông tin, Trường Đại học Công nghệ, Đại học Quốc Gia Hà Nội đã tận tình giảng dạy, hướng dẫn nghiên cứu khoa học cho tôi trong suốt thời gian theo học tại trường cũng như trong quá trình làm luận văn này.

Xin cảm ơn các anh, chị, em và các bạn học viên bộ môn Kỹ thuật Phần mềm, những người đã giúp đỡ, động viên tinh thần và chia sẻ kinh nghiệm quý báu giúp tôi vượt qua các khó khăn, vướng mắc để có thể hoàn thành luận văn này.

Mặc dù đã cố gắng, nhưng tôi tin chắc luận văn của tôi còn nhiều thiếu sót và có rất nhiều nội dung có thể hoàn thiện tốt hơn. Tôi rất mong nhận được những ý kiến đánh giá, phê bình và góp ý của các thầy cô, anh chị và các bạn.

Trân trọng,

Tác giả

Nguyễn Đức Thọ

MỤC LỤC

Danh mục các ký hiệu và chữ viết tắt

STT	Thuật ngữ, chữ viết tắt	Diễn giải
1	PTA	Probability Timed Automata Ô tô mát thời gian xác suất.
2	DTMC	Discrete Time Markov Chain Xích Markov thời gian rời rạc
3	CTMC	Continuous Time Markov Chain Xích Markov thời gian liên tục
4	MDP	Markov Decision Process Quá trình Quyết định Markov
5	TA	Timed Automata Ô tô mát thời gian
6	CTL	Computation Tree Logic Cây logic tính toán
7	PCTL	Probability Computation Tree Logic Cây logic tính toán xác suất
8	PTCTL	Probability Timed Computation Tree Logic Cây logic tính toán thời gian xác suất

Các ký hiệu

STT	Ký hiệu	Giá trị biểu diễn
1	$\mathscr{L}$	Tập các nhãn gắn trên các cạnh
2	$\mathbb{N}$	Tập các số nguyên
3	$\mathbb{R}$	Tập các số thực
4	$\mathbb{Q}$	Tập các số hữu tỉ, có thể biểu diễn được dưới dạng a/b với a, b là các số nguyên
5	$\subseteq$	Quan hệ tập con
6	$\vDash$	Thỏa mãn điều kiện
7	$\mathcal{X}$	Tập các đồng hồ trong PTA
8	χ	Ràng buộc thời gian trong PTA

Danh mục các bảng

Danh mục các hình vẽ, đồ thị

MỞ ĐẦU

Trong những năm gần đây, đã có nhiều nghiên cứu về các phương pháp kiểm chứng mô hình nhằm kiểm tra thuộc tính của các hệ thống, các giao thức tự động hoặc áp dụng để sinh các bộ kịch bản kiểm thử nhằm kiểm tra thuộc tính của các hệ thống. Việc kiểm chứng mô hình đòi hỏi các hệ thống cần được mô hình hóa và biểu diễn trên các không gian trạng thái, với nhiều kỹ thuật mô hình hóa khác nhau đã được nghiên cứu và triển khai, áp dụng trong thực tế. Việc biểu diễn các hệ thống được thực hiện bởi các ô tô mát, và các hành động trên các hệ thống được biểu diễn bởi các chuyển dịch trạng thái tương ứng trên ô tô mát, trong khi các thuộc tính cần kiểm chứng được biểu diễn bởi các mệnh đề logic. Kiểm chứng mô hình xác suất là dạng mở rộng của kiểm chứng mô hình, nhằm biểu diễn và kiểm chứng các tính chất của hệ thống trong đó việc chuyển trạng thái của hệ thống xảy ra có yếu tố xác suất, theo đó việc chuyển từ một trạng thái sang một hoặc nhiều trạng thái khác theo phân bổ xác suất.

Ô tô mát thời gian xác suất (PTA) là khái niệm mở rộng của Ô tô mát thời gian, bổ sung thêm phân bổ xác suất rời rạc, và có thể áp dụng để mô hình hóa các giao thức có yếu tố ngẫu nhiên, các hệ thống có khả năng chịu lỗi cũng như có thể áp dụng để mô hình hóa hệ thống trong nhiều lĩnh vực khác như kinh tế, kỹ thuật, sinh học, .v.v.

Đề tài này tập trung vào việc nghiên cứu các đặc tính, mô hình hóa các hệ thời gian thực xác suất và khả năng áp dụng trong việc kiểm chứng mô hình nhằm kiểm chứng tự động các thuộc tính của hệ thời gian thực xác suất bằng công cụ. Phạm vi nghiên cứu của đề tài bao gồm: (1) nghiên cứu các tính chất Markov của các hệ thống, các loại chuỗi Markov và các tính chất của nó; (2) các hệ tự động thời gian thực xác suất và các phương pháp kiểm chứng tự động tính chất của hệ thời gian thực xác suất; (3) nghiên cứ công cụ kiểm chứng mô hình PRISM và khả năng áp dụng trong việc kiểm chứng các tính chất của hệ thời gian thực xác suất, (4) Áp dụng nghiên cứu trong việc mô hình hóa giao thức Alternative Bit Protocol bằng hệ thời gian thực xác suất và thực hiện cài đặt trên công cụ PRISM, thực hiện kiểm chứng tự động các tính chất của hệ thống bằng khả năng kiểm chứng của PRISM.

Chương 1. TỔNG QUAN

Hoạt động của một hệ thống máy tính, cũng như của bất kỳ một hệ thống nào trong các lĩnh vực khác như sinh học, hóa học, vật lý, … đều là chuyển đổi giữa các trạng thái khác nhau của hệ thống. Nhà toán học người Nga Andrei Andreevich Markov (1856-1922) đã nghiên cứu một loại phân bổ xác suất quan trọng trong không gian trạng thái, đặt nền tảng cho một lớp các bài toán, mô hình chuyển trạng thái được gọi chung là không gian trạng thái Markov, hoặc các quá trình có tính chất Markov (Markov properties). Các quá trình có tính chất Markov được áp dụng rộng rãi trong các nghiên cứu cơ bản về hệ thống máy tính, cơ chế làm việc của các ô tô mát, hoặc việc mô hình hóa các hệ thống trong các lĩnh vực khác như sinh học, kinh tế, giao thông.

Để bổ sung thêm giá trị thời gian, cũng như các ràng buộc thời gian đối với việc chuyển giữa các trạng thái trong thời gian thực, các nghiên cứu bổ sung sau này đã hoàn thiện các mô hình biểu diễn các ô tô mát thời gian, thực hiện bởi [1] với các giả thiết đồng hồ thời gian chính xác tuyệt đối và các hoạt động xảy ra ngay tức thì (độ trễ bằng 0 tuyệt đối). Các yếu tố xác suất được nghiên cứu và bổ sung trong biểu diễn ô tô mát thời gian, mở rộng thành mô hình biểu diễn ô tô mát thời gian xác suất, được định nghĩa bằng các phân bố thời gian rời rạc và các lựa chọn ngẫu nhiên cho các nhánh trong ô tô mát thời gian. Mô hình biểu diễn ô tô mát thời gian xác suất giúp có thể mô hình hóa các hệ thống ngoài đời thực, sử dụng các công cụ kiểm chứng để kiểm chứng các đặc tính của các hệ được biểu diễn.

Phạm vi đề tài nhằm nghiên cứu các tính chất của ô tô mát thời gian thực xác suất và thực hiện kiểm chứng tự động các tính chất đó bằng công cụ. Có thể phát biểu bài toán kiểm chứng mà đề tài cần giải quyết như sau: Cho hệ thống thời gian thực xác suất M. Thực hiện kiểm chứng tự động bằng công cụ xem M có thỏa mãn tính chất P hay không.

Để có thể giải quyết bài toán kiểm chứng tự động bằng công cụ, phạm vi nghiên cứu của đề tài sẽ tập trung vào các nội dung chính bao gồm:

1. Mô hình hóa hệ xác suất thời gian thực bằng ô tô mát thời gian thực xác suất PTA.
2. Hình thức hóa các tính chất xác suất cần kiểm chứng bằng cây lô gic tính toán xác suất PCTL.
3. Nghiên cứu công cụ hỗ trợ cài đặt PTA và biểu diễn tính chất để thực hiện kiểm chứng tự động.

4. Áp dụng với nghiên cứu với giao thức Alternating Bit Protocol: Mô hình hóa hệ giao thức bằng PTA, hình thức hóa các tính chất bằng PCTL và thực hiện cài đặt hệ giao thức trên công cụ PRISM.

Cấu trúc trình bày của đề tài gồm sáu phần chính. Phần đầu là giới thiệu tổng quan về đề tài, phần hai là cơ sở khoa học của đề tài, nêu các biểu diễn và tính chất các quá trình Markov. Phần ba trình bày việc kiểm chứng tự động các ô tô mát thời gian thực xác suất, gồm các cú pháp và ngữ nghĩa các PTA, đặc tả tính chất cho PTA và các phương pháp kiểm chứng tự động đối với PTA. Phần bốn giới thiệu công cụ kiểm chứng mô hình PRISM, là công cụ có khả năng kiểm chứng tự động các tính chất của các hệ mô hình hóa khác nhau trong đó có PTA. Phần năm trình bày một trường hợp áp dụng PTA để mô hình hóa hệ giao thức Alternating Bit Protocol và hình thức hóa các tính chất của ABP, cài đặt bằng công cụ PRISM và kiểm chứng tự động các tính chất của ABP. Phần cuối cùng là phần kết luận, đề xuất các hướng nghiên cứu mở rộng của đề tài.

Chương 2. CƠ SỞ KHOA HỌC CỦA ĐỀ TÀI

Giới thiệu chung về chuỗi Markov

Chuỗi Markov (Markov chain hay Markov process) là một quá trình ngẫu nhiên với các đặc tính Markov. Thuật ngữ "chuỗi Markov" (hay "xích Markov", "quá trình Markov") dành để chỉ trình tự các biến ngẫu nhiên mà một tiến trình trải qua, với đặc tính Markov được định nghĩa là khả năng xuất hiện của các biến ngẫu nhiên tiếp theo chỉ phụ thuộc vào biến hiện tại (tạo thành 1 chuỗi). Khi áp dụng trong không gian trạng thái, nó có thể được dùng để mô tả các hệ thống có các chuỗi trạng thái liên kết với nhau, và những biến đổi sang trạng thái tiếp theo chỉ phụ thuộc trạng thái hiện tại của hệ thống.

Quá trình có tính chất Markov

Một quá trình là một ô tô mát (hay quá trình, hoặc một chuỗi trạng thái) bắt đầu với một trong các trạng thái này và dịch chuyển từ trạng thái này sang trạng thái khác. Nếu ô tô mát đang ở trạng thái s_i, sau đó chuyển sang trạng thái s_j ở bước tiếp theo với xác suất được biểu diễn bằng p_{ij}, và giá trị này không phụ thuộc vào các trạng thái ô tô mát trước khi chuyển sang trạng thái hiện tại. Các giá trị xác suất p_{ij} được gọi là các xác suất chuyển (transition probability). Ô tô mát có thể ở trạng thái hiện tại, với xác suất được ghi nhận là p_{ii}. Việc phân bố xác suất các trạng thái ban đầu được định nghĩa bởi S. Thông thường các trạng thái ban đầu được xác định bởi một hoặc một số trạng thái, trong đó nếu các phân bố xác suất chuyển từ trạng thái hiện tại sang các trạng thái tiếp theo chỉ phụ thuộc vào trạng thái hiện tại thì quá trình như vậy được gọi là có tính chất Markov, gọi ngắn gọn là Quá trình Markov.

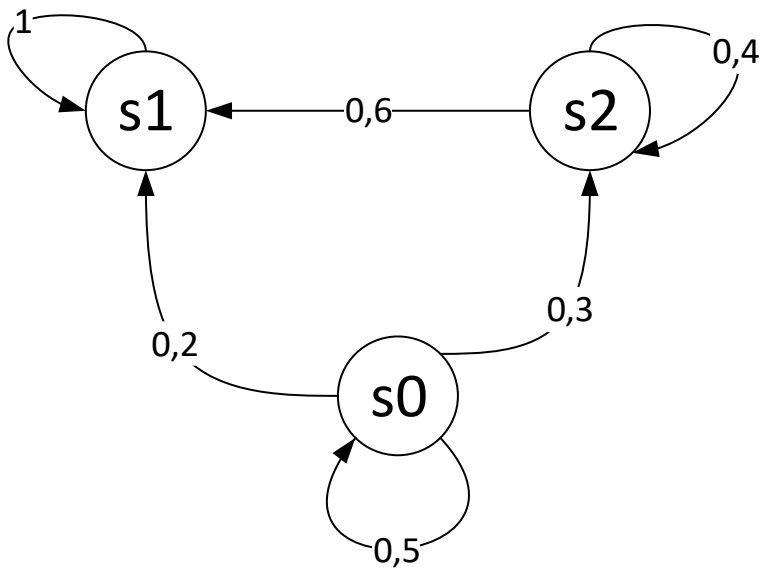
$$p_{n+1} = Pr(s_{n+1} = x \mid s_1 = x_1, s_2 = x_2, ..., s_n = x_n) = Pr(s_{n+1} = x \mid s_n = x_n)$$

2.1 Xích Markov thời gian rời rạc (DTMC)

Định nghĩa

Hình 2.1 là minh họa cho một xích Markov (hay một chuỗi Markov) đơn giản, với 3 trạng thái s0, s1, s2 và xác suất chuyển giữa các trạng thái thể hiện trên các đường nối giữa các trạng thái.

DTMC là quá trình biến đổi trạng thái rời rạc, thuần nhất theo thời gian. Khái niệm thuần nhất theo thời gian được hiểu là xác suất p_{ij} chuyển trạng thái từ s_i sang s_j của chuỗi Markov là giá trị không phụ thuộc thời gian. Nói cách khác tại mọi thời điểm được xét, giá trị xác suất p_{ij} không thay đổi.



Hình 2.1: Markov chain

Khái niệm tập không gian trạng thái trong chuỗi Markov không được thống nhất về phạm vi giới hạn trong các tài liệu khác nhau, không gian trạng thái có thể dùng để biểu diễn quá trình xảy ra với bất kỳ dạng biến đổi nào, do vậy có thể là không gian hữu hạn hoặc vô hạn, đếm được hoặc không đếm được. Tuy nhiên, phần lớn các ứng dụng chuỗi Markov chỉ sử dụng tập không gian trạng thái hữu hạn, hoặc tập không gian trạng thái vô hạn đếm được, có các đặc điểm phân tích thống kê đơn giản hơn.

Vector xác suất

Vector xác suất với *r* thành phần là một vector dòng với các giá trị không âm, và tổng các giá trị là 1. Gọi u là vector xác suất biểu diễn trạng thái ban đầu của chuỗi Markov, khi đó thành phần thứ i của u biểu thị xác suất chuỗi Markov sẽ bắt đầu ở trạng thái s_i.

Ma trận xác suất (còn gọi là ma trận chuyển – transition matrix)

Giá trị p_{ij} cho biết xác suất chuyển từ trạng thái i sang trạng thái j. Do xác suất là giá trị không âm, và hệ sẽ phải chuyển đến một trạng thái nào đó, ta có:

$p_{ij} \geq 0$, $\quad i, j \geq 0$; $\quad \sum_{j=0}^{r} p_{ij} = 1$, $\quad i = 0, 1, \ldots. r$

Ma trận P kích thước *r x r* gồm các phần tử p_{ij} là ma trận chuyển của chuỗi Markov có *r+1* trạng thái.

$$\begin{bmatrix} p_{00} & p_{01} & \dots & p_{0r} \\ p_{10} & p_{11} & \dots & p_{12} \\ \dots & \dots & \dots & \dots \\ p_{r0} & p_{r1} & \dots & p_{rr} \end{bmatrix}$$

Chuỗi Markov hấp thụ (absorbing Markov chain)

Một trạng thái s_i của chuỗi Markov được gọi là trạng thái hấp thụ (absorbing) nếu chuỗi không thể thoát ra khỏi trạng thái đó ($p_{ii} = 1$). Một chuỗi Markov là hấp thụ nếu nó có ít nhất một trạng thái hấp thụ, và từ một trạng thái bất kỳ nó có thể chuyển về trạng thái hấp thụ (có thể qua nhiều bước). Trong một chuỗi Markov hấp thụ, một trạng thái không phải là trạng thái hấp thụ được gọi là trạng thái trung gian hay trạng thái chuyển (transient state).

Dạng chuẩn của ma trận chuyển (canonical form)

Xét một chuỗi Markov hấp thụ bất kỳ. Đánh số lại các trạng thái để các trạng thái trung gian đứng trước. Giả sử có r trạng thái hấp thụ và t trạng thái trung gian, ma trận chuyển sẽ có dạng chuẩn như sau:

$$\mathbf{P} = \begin{array}{c} \\ \text{TR.} \\ \text{ABS.} \end{array} \begin{array}{c} \begin{array}{cc} \text{TR.} & \text{ABS.} \end{array} \\ \left(\begin{array}{c|c} \mathbf{Q} & \mathbf{R} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right) \end{array}$$

Trong đó **I** là ma trận đơn vị r-r, **0** là ma trận 0 cho các xác suất chuyển r-t, **R** là ma trận chuyển t-r (khác 0) và **Q** là ma trận chuyển t-t. Trong dạng này, t trạng thái đầu tiên là trạng thái trung gian, và r trạng thái cuối cùng là trạng thái hấp thụ. TR thể hiện các dòng/cột ứng với trạng thái trung gian (gồm t dòng/cột), ABS thể hiện các dòng/cột ứng với trạng thái hấp thụ (gồm r dòng/cột).

Ma trận cơ sở N (fundamental matrix)

Với một chuỗi Markov hấp thụ với ma trận chuyển **P**, ma trận nghịch đảo của I-Q: $\mathbf{N} = \mathbf{(I-Q)}^{-1}$ được gọi là ma trận cơ sở của **P**. Giá trị n_{ij} của **N** cho biết con số kỳ vọng mà chuỗi Markov sẽ ở trạng thái trung gian s_j nếu xuất phát từ trạng thái trung gian s_i.

Chuỗi Markov liên thông (ergodic chain)

Một chuỗi Markov được gọi là liên thông nếu nó có thể chuyển sang mọi trạng thái từ bất kỳ trạng thái nào (không nhất thiết phải trong 1 lần chuyển).

Trong nhiều sách, chuỗi Markov như vậy còn được gọi là chuỗi Markov tối giản (irreducible chains)

Thời gian chu kỳ của trạng thái

Nếu một chuỗi Markov liên thông xuất phát từ trạng thái s_i, số lượt chuyển kỳ vọng để đạt đến trạng thái s_j lần đầu tiên được gọi là thời gian lần đầu tiên đi từ s_i đến s_j. Giá trị này được biêu diễn bởi m_{ij}. Theo quy ước $m_{ii} = 0$. Nếu một chuỗi Markov liên thông xuất phát từ trạng thái s_i, số lượt chuyển kỳ vọng của chuỗi đến khi quay về s_i lần đầu tiên được gọi là thời gian chu kỳ cho trạng thái s_i. Biểu diễn giá trị này là r_i.

Chuỗi Markov đều (regular Markov chain)

Một chuỗi Markov được gọi là chuỗi đều (regular) nếu tồn tại giá trị mũ của ma trận chuyển là ma trận chỉ bao gồm các phần tử dương.

Nói cách khác, với một giá trị n nào đó, có thể di chuyển từ bất kỳ trạng thái nào sang bất kỳ trạng thái bất kỳ khác sau đúng n bước. Có thể thấy từ định nghĩa, mọi chuỗi Markov đều là chuỗi Markov liên thông. Tuy nhiên một chuỗi Markov liên thông không nhất thiết là chuỗi Markov đều.

Định nghĩa xích Markov thời gian rời rạc DTMC

Theo [10], Chuỗi Markov thời gian rời rạc (gắn nhãn) D là bộ (S, s_0, P, L), trong đó:

- S là tập hữu hạn các trạng thái
- s_0 là trạng thái ban đầu
- P: S x S $\rightarrow$ [0, 1] là ma trận xác suất, $\sum_{s' \in S} P(s, s') = 1$, mọi $s \in S$
- L: S $\rightarrow 2^{AP}$ là một nhãn mệnh đề logic với giá trị true tại trạng thái s.

Một hành trình xuyên qua một DTMC là một chuỗi (hữu hạn hoặc vô hạn) các trạng thái $\omega = s0\ s1\ s2 \ldots$ với $P(s_i, s_{i+1}) > 0$ với mọi $i \geq 0$.

Một xích Markov thời gian rời rạc chỉ chấp nhận các lựa chọn theo xác suất, khác biệt cơ bản với quyết định lựa chọn bất định (nondeterministic choice): tần suất được chọn của các cạnh xác suất được xác định bởi xác suất của cạnh đó, trong khi với các lựa chọn bất định được thực hiện bởi môi trường và có thể tự do lựa chọn cạnh bất kỳ. Trong DTMC không có khái niệm về thời gian thực, mặc dù lý luận về thời gian rời rạc có thể thực hiện

thông qua các biến trạng thái theo dõi thời gian và việc đếm số bước chuyển trạng thái.

Xác suất chuyển từ trạng thái s_i sang s_j sau n bước:

Với **P** là ma trận chuyển của chuỗi Markov. Giá trị thứ ij $p_{ij}^{(2)}$ của ma trận $\mathbf{P}^n$ cho biết xác xuất của chuỗi Markov, bắt đầu tại trạng thái s_i, sẽ ở trạng thái s_j sau n bước chuyển.

Tính chất của xích Markov đều

Cho **P** là ma trận chuyển của một xích Markov đều. Khi đó, $n \to \infty$, ma trận hàm mũ $\mathbf{P}^n$ sẽ tiến dần tới ma trận giới hạn **W** với mọi dòng đều có cùng giá trị vector **w**. Vector **w** là ma trận dương tuyệt đối (mọi giá trị của w đều là số dương, và tổng các giá trị là 1)

Tính chất hội tụ của ma trận đều

Cho **P** là ma trận chuyển đều, với

$$\mathbf{W} = \lim_{n\to\infty} \boldsymbol{P}^n$$

Với **w** là vector dòng của **W**, và c là vector cột với các giá trị là 1. Khi đó:

a) **wP** = **w**, và mọi vector dòng v với **vP** = **v** chứa các giá trị bội số của **w**.
b) **Pc** = **c**, và mọi vector cột x sao cho **Px** = **x** là bộ số của **c**

Vector w được gọi là vector dòng cố định của ma trận P. Vector c được gọi là vector cột cố định của ma trận P.

Định lý về tính chất hội tụ

Cho P là ma trận chuyển của một chuỗi đều và v là một vector xác suất bất kỳ, khi đó:

$$\lim_{n\to\infty} v(P)^n = w$$

Với w là vector dòng cố định của P, là vector xác định duy nhất.

Như vậy, ta bắt đầu chuỗi Markov với xác suất ban đầu **v**, và giá trị vector xác suất $\mathbf{vP}^n$ cho biết khả năng chuỗi Markov tại các trạng thái khác nhau sau n bước chuyển. Định lý về tính chất hội tụ cho thấy, với một chuỗi Markov nói chung, xác suất của chuỗi ở trạng thái s_j đạt gần giá trị w_j.

Từ tính chất nêu trên cho ta thấy ý nghĩa mới của vector w. Giả thiết vector ban đầu cho thấy s_i có khả năng w là vector ban đầu với xác suất w_i

với mọi i. Như vậy xác suất chuỗi ở trạng thái bất kỳ sau n dịch chuyển được xác định bởi $\mathbf{wP}^n = \mathbf{w}$, và giống nhau với mọi bước chuyển. Phương pháp bắt đầu này cung cấp cho ta một chuỗi được gọi là "ổn định". Thực tế là **w** là vector xác xuất duy nhất thỏa mãn **wP = w** cho thấy ta phải có một vector xác suất ban đầu như trên để có một chuỗi ổn định.

Tính chất duy nhất của vector dòng cố định trong chuỗi liên thông

Với một xích Markov liên thông, chỉ có một vector xác suất duy nhất w thỏa mãn **wP = w** và **w** là vector với tất cả các phần tử dương. Bất kỳ vector dòng nào thỏa mãn **vP = v** đều là bội số của **w**. Bất kỳ vector cột **x** thỏa mãn **Px = x** phải là vector hằng số.

Thời gian chu kỳ (mean recurrence time) của xích liên thông

Với xích Markov liên thông, thời gian chu kỳ của trạng thái s_i là $r_i = 1/w_i$, với w_i là giá trị thứ i của vector xác suất cố định của ma trận chuyển.

Từ công thức có thể thấy, với một xích Markov đều, mọi phần tử của vector xác suất cố định **w** đều là số dương.

Số bước chuyển của chuỗi hấp thụ

Gọi t_i là số bước kỳ vọng trước khi ô tô mát bị hấp thụ khi nó xuất phát từ trạng thái s_i, và t là vector cột với giá trị thứ i là t_i. Khi đó

$$\mathbf{t = Nc,}$$

trong đó c là một vector cột với tất cả giá trị bằng 1.

Xác suất hấp thụ tại trạng thái hấp thụ

Gọ b_{ij} là xác suất của ô tô mát bị hấp thụ tại trạng thái hấp thụ s_j khi nó xuất phát từ trạng thái trung gian s_i. Gọi **B** là ma trận với các giá trị b_{ij}. Khi đó **B** là một ma trận chuyển t-r, và

$$\mathbf{B = NR,}$$

Trong đó **N** là ma trận cơ sở và **R** là ma trận trong biểu diễn dạng chuẩn của ma trận chuyển.

Biểu diễn và kiểm chứng tính chất DTMC

Logic PCTL (Probabilistic CTL) thay thế các biểu diễn định lượng của CTL với các toán tử xác suất $\mathcal{P}_{\bowtie\rho}(.)$ với $\rho \in [0,1]$ là ràng buộc xác suất hoặc ngưỡng, và $\bowtie \in \{\leq, <, \geq, >\}$. Cú pháp của mệnh đề logic ϕ của PCTL như sau:

$$\phi ::= \text{true} \mid \alpha \mid \phi \wedge \phi \mid \neg \phi \mid \mathcal{P}_{\bowtie\rho}(\propto)$$

Trong đó $\propto$ là công thức đường (trạng thái tiếp theo Xϕ hoặc ϕ_1 U ϕ_2). Ý nghĩa của toán tử xác suất như sau:

$$s \vDash \mathcal{P}_{\bowtie \rho}(\propto) \text{ nếu và chỉ nếu } Pr_s\{\omega \in Path_s \mid \omega \vDash \propto\} \bowtie \rho$$

được hiểu là xác suất trên các cung đường $\propto$ được tính và so sánh với ràng buộc xác suất, cho giá trị true hoặc false. Lưu ý trong khi $\mathcal{P}_{\geq 0}(\phi_1 \text{ U } \phi_2)$ tương đương toán tử tồn tại.

Giải thuật kiểm chứng PCTL thực hiện tương tự như CTL trên ϕ, bằng cách tìm các tập Sat(ϕ) thỏa mãn mệnh đề logic ϕ.

2.2 Quá trình quyết định Markov (MDP)

Để mô hình hóa các hệ thống trong thực tế, bên cạnh các yếu tố ngẫu nhiên còn có phần ra quyết định của người điều khiển. Quá trình quyết định Markov (Markov Decision Process - MDP) là nền tảng toán học cho việc mô hình hóa các hệ chuyển trạng thái với các tình huống như vậy.

Định nghĩa Quá trình quyết định Markov MDP

Theo [10], Quá trình quyết định Markov (gắn nhãn) $\mathcal{M}$ là bộ (S, s_0, Steps, L), trong đó:

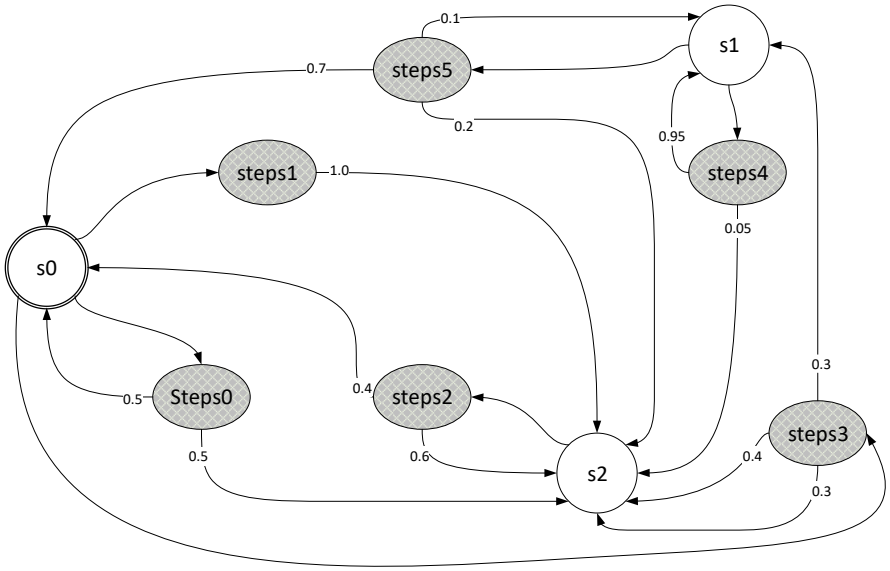
- S là tập hữu hạn các trạng thái
- s_0 là trạng thái ban đầu
- Steps là một hàm gán mỗi trạng thái $s \in S$ một tập hợp hữu hạn, không rỗng Steps(s) các phân bổ xác suất tại S.
- L: S $\rightarrow 2^{AP}$ là một nhãn mệnh đề logic với giá trị true tại trạng thái s.

Việc thực thi của quá trình quyết định Markov thông qua các thay đổi không xác định và các lựa chọn theo xác suất: khi thuộc một trạng thái cụ thể, hệ thống chọn một cách không xác định một trong các phân bổ xác suất steps $\in$ Steps đối với trạng thái đích. Một hành trình ω của $\mathcal{M}$ là một chuỗi (hữu hạn hoặc vô hạn) các trạng thái

$$s_0 \xrightarrow{\mu 0} s_1 \xrightarrow{\mu 1} s_2 \xrightarrow{\mu 2} \dots$$

Trong đó $s_i \in S$, $\mu_i \in Steps(s_i)$ và $\mu_i(s_{i+1}) > 0$.

Việc lựa chọn trạng thái tiếp theo quyết định bởi phân bố xác suất của steps, và lựa chọn steps tại mỗi trạng thái chỉ phụ thuộc trạng thái đó mà không phụ thuộc vào các quá trình và các trạng thái đã qua, do vậy quá trình quyết định Markov MDP cũng có tính chất Markov.



Hình 2.2 Minh họa MDP với 3 trạng thái (s0, s1, s2) và tập các phân bố xác suất Steps (0-5)

Hình 2.2 minh họa một MDP đơn giản với tập S gồm 3 trạng thái s0, s1, s2, trong đó s0 là trạng thái ban đầu, tập các phân bố xác suất trong S gồm 6 steps. Tại mỗi bước dịch chuyển của MDP, một steps bất kỳ được chọn, và MDP sẽ chuyển sang trạng thái tiếp theo theo phân bố xác suất trong steps đã chọn.

Định nghĩa một lập lịch (adversary) của một MDP $\mathcal{M}$ là một hàm A ánh xạ mọi hành trình hữu hạn ω của $\mathcal{M}$ tới một phân bố xác suất A(ω) trên S sao cho A(ω) có giá trị tại trạng thái cuối cùng của ω. Hành vi của MDP $\mathcal{M}$ với một lập lịch đã chọn trước có thể được mô tả bằng một xích Markov rời rạc DTMC P^A, với các trạng thái là các hành trình hữu hạn của $\mathcal{M}$ và xác suất chuyển được cho bởi phân bổ xác suất của A: Với hai hành trình hữu hạn ω, ω', ta có $P^A(\omega, \omega') = A(\omega)(s)$ nếu ω' có dạng $\omega \xrightarrow{A(\omega)} s$ và trong các trường hợp khác thì $P^A(\omega, \omega') = 0$. Vì vậy ta có thể định nghĩa phân bố xác suất Pr_s^A trên tập các hành trình $Path_s^A$ của lập lịch A.

Các phát biểu xác suất liên quan tới MDP thường bao gồm các giá trị định lượng trên các lập lịch của MDP, như tính xác suất lớn nhất hoặc xác suất nhỏ nhất để quan sát được một sự kiện trong tập các lập lịch.

Kiểm chứng tính chất trên các MDP

Logic cây tính toán xác suất (PCTL - Probabilistic Computation Tree Logic) được định nghĩa cho MDPs tương tự DTMC, với sự khác biệt là ngữ nghĩa được tham số hóa bởi lớp Adv các lập lịch và toán tử xác suất chứa các định lượng tường minh.

$s \vDash Adv\ \mathcal{P}_{\bowtie\rho}(\propto)$ nếu và chỉ nếu $Pr^A\{\omega \in Path^A_s \mid \omega \vDash Adv \propto\} \bowtie \rho$ với mọi lập lịch $A \in Adv$

2.3 Xích Markov thời gian liên tục (CTMC)

Xích Markov DTMC và quá trình quyết định Markov MDP chỉ có thể mô hình hóa thời gian rời rạc. Xích Markov thời gian liên tục có các trạng thái là rời rạc, một tham số thời gian trên tập $\mathbb{R}_{\geq 0}$, nhưng không cho phép các lựa chọn bất định. Mỗi quá trình chuyển đổi có một độ trễ ngẫu nhiên phân bố theo cấp số nhân, và một cuộc đua điều kiện được sử dụng để mô tả các dịch chuyển trạng thái đồng thời kích hoạt.

Định nghĩa xích Markov thời gian rời rạc CTMC

Theo [10], Chuỗi Markov thời gian rời rạc (gắn nhãn) C là một bộ (S, s_0, R, L), trong đó:

- S là tập hữu hạn các trạng thái
- s_0 là trạng thái ban đầu
- R: S x S → $R_{\geq 0}$ là ma trận tốc độ.
- L: S → 2^{AP} là một nhãn mệnh đề logic với giá trị true tại trạng thái s

$E(s) = \sum_{s' \in S} R(s,s')$ biểu diễn xác suất thực hiện một chuyển dịch từ s trong *t* đơn vị thời gian bằng $1 - e^{-E(s).t}$. Trong trường hợp $R(s,s') > 0$ với nhiều hơn một trạng thái s', có một cuộc đua các điều kiện chuyển dịch từ s đến các s' để chọn chuyển dịch sẽ được thực hiện. Do vậy P(s,s') biểu diễn xác suất chuyển s đến s' trong một bước chuyển bằng với xác suất các độ trễ của việc chuyển s đến s' sao cho độ trễ này kết thúc trước các độ trễ đến các s' khác.

Một hành trình trong CTMC là một chuỗi không rỗng $s_0t_0s_1t_1s_2t_2\ldots$ trong đó $R(s_i, s_{i+1}) > 0$ và $t_i \in \mathbb{R}_{\geq 0}$ với mọi $i \geq 0$. Giá trị t_i biểu diễn độ dài thời gian tại trạng thái s_i.

Việc phân tích các xích CTMC thường dựa trên các trạng thái tức thời tại một thời gian cụ thể và các trạng thái kỳ vọng (trạng thái của CTMC trong thời gian đủ lớn). Xác suất tức thời $\pi_{s,t}(s')$ được định nghĩa là xác suất khi bắt đầu tại s, và ở tại s' tại thời điểm *t*. Xác suất kỳ vọng $\pi_s(s')$ được định nghĩa là giá trị $\lim_{t\to\infty}\pi_{s,t}(s')$.

Chương 3. KIỂM CHỨNG TỰ ĐỘNG CÁC PTA

Một hàm phân bổ xác suất rời rạc trên tập đếm được Q là hàm μ:Q→ [0,1] với $\sum_{q\in Q}\mu(q) = 1$.

Với hàm μ:Q→$\mathbb{R}_{\geq 0}$, định nghĩa Support(μ)={q∈Q | μ(q) > 0}

Với tập đếm được Q bất kỳ, Dist(Q) là tập các hàm μ:Q→[0,1] sao cho Support(μ) là một tập đếm được, và μ giới hạn trong Support(μ) là phân bố xác suất. Với q∈Q được gọi là một điểm phân bố tại q. Gọi AP là tập các mệnh đề nguyên tử, ta giả sử các mệnh đề này cố định trong suốt tài liệu luận văn.

3.1 Các định nghĩa cho PTA

Cấu trúc thời gian xác suất (Timed Probabilistic Systems – TPS[1])

Một cấu trúc thời gian xác suất (TPS, còn gọi là hệ thời gian xác suất) T là bộ (S, s0, Act, Steps, lab) trong đó S là tập các trạng thái (có thể vô hạn), s0 ∈S là trạng thái ban đầu, Act là tập hữu hạn các hành động, Steps:S x (Act ∪ $\mathbb{R}_{\geq 0}$) → Dist(S) là hàm xác suất chuyển và lab: S → 2^{AP} là hàm gắn nhãn.

Một TPS T bắt đầu từ trạng thái s_0, và khi đang ở s∈S, có một lựa chọn không xác định trước giữa việc thực thi một hành động hoặc để thời gian trôi qua và không hành động gì (letting time pass) a ∈ (Act ∪ $\mathbb{R}_{\geq 0}$) (là lý do Steps(s,a) được định nghĩa). Sau khi lựa chọn được thực hiện (một hành động hoặc cho một khoảng thời gian trôi), trạng thái s' tiếp theo được chọn ngẫu nhiên theo phân bổ xác suất Steps(s,a). Ta giả thiết tại mỗi s∈S, luôn có ít nhất một lựa chọn hành động hoặc để thời gian trôi. Một chuỗi Markov quyết định MDP M là trường hợp đặc biệt của TPS khi bỏ qua yếu tố thời gian trong hàm chuyển, ví dụ hàm chuyển sẽ có dạng $Steps_M$: S x Act → Dist(S).

Một đường đi của TPS thể hiện một chuỗi các hành động trên hệ thống, gồm cả các quyết định theo xác suất và quyết định không xác định.

$$\omega = s_0 \xrightarrow{(a0,\mu 0)} s_1 \xrightarrow{(a1,\mu 1)} s_2 \xrightarrow{(a2,\mu 2)} \dots$$

[1] Một số tài liệu gọi là Timed Probabilistic Structures

Trong đó $a_{2i} \in \mathbb{R}_{\geq 0}$ và $a_{2i+1} \in Act$ với $i \in \mathbb{N}$.

Ký hiệu ω(i) là trạng thái s_i thứ (i+1) của ω và tổng lũy kế thời gian đến trạng thái ω(i) được xác định bởi

$$dur_\omega(i) \stackrel{def}{=} \Sigma_{0 \leq j < i \wedge a_j \in \mathbb{R}_{\geq 0}} (a_j)$$

Một vị trí của ω là cặp $(i, t) \in \mathbb{N} \times \mathbb{R}_{\geq 0}$ sao cho $t \leq dur_\omega(i+1) - dur_\omega(i)$. Ta gọi vị trí (j, t') là vị trí đứng trước (i,t), ký hiệu $(j, t') \prec (i,t)$, khi $j < i$ hoặc $j = i$ và $t' < t$.

Để xác định hành vi của PTS T, ta sử dụng ký hiệu *adversary* (lập lịch), trong đó chỉ bao gồm các lựa chọn không xác định. Một cách hình thức, một *adversary* là một hàm từ tập hữu hạn các đường đi với các số chẵn các chuyển dịch tới các khoảng thời gian có thể, và từ tập hữu hạn các đường đi với số lẻ các chuyển dịch tới các hành động có thể thực thi. Với một adversary cố định σ và trạng thái s, ta có thể định nghĩa độ đo xác suất $Pr^{\sigma}_{T,s}$ trên tập hợp $Path^{\sigma}_{T,s}$ của các đường đi không giới hạn xuất phát từ s tương ứng với σ. Với một biến số thực ngẫu nhiên *f* trên $Path^{\sigma}_{T,s}$, ký hiệu $\mathbb{E}^{\sigma}_{T,s}(f)$ là giá trị kỳ vọng của *f* theo phân bổ $Pr^{\sigma}_{T,s}$.

Ta chỉ giới hạn phạm vi xem xét với các adversary có thời gian phân kỳ, ví dụ ta không xét việc thực thi hành động trong đó thời gian không thể vượt qua một giới hạn cụ thể, do những ràng buộc này không phù hợp với một hệ thống thực tế được mô hình hóa. Một cách hình thức, một adversary σ của một TPS T có thời gian phân kỳ nếu

$$Pr^{\sigma}_{T,s}(\{\omega \in Path^{\sigma}_{T,s} \mid \forall c \in \mathbb{N}.\ \exists i \in \mathbb{N}.\ dur_\omega(i) > c\}) = 1.$$

Với mọi trạng thái s thuộc T. Ta ký hiệu Adv_T là tập tất cả các adversary có thời gian phân kỳ của T.

Khi thực hiện các bài toán kiểm chứng với TPS, các đặc tính có thể được xét và kiểm chứng dễ hơn với các cấu trúc thưởng (reward structrure, còn được gọi là chi phí hay giá).

Cấu trúc thưởng (reward structrure)

Một cấu trúc thưởng của TPS T=(S, s0, Act, Steps, lab) là cặp $r = (r_S, r_{Act})$ trong đó r_S: $S \rightarrow \mathbb{R}_{\geq 0}$ là hàm thưởng trên trạng thái và r_{Act}: $(SxAct) \rightarrow \mathbb{R}_{\geq 0}$ là hàm thưởng trên hành động.

Với một cấu trúc thưởng $r = (r_S, r_{Act})$ và hành động s, giá trị $r_S(s)$ xác định tốc độ (theo thời gian) mà giá trị thưởng tích lũy được khi ở trạng thái

s. Mặt khác, với trạng thái s và hành động a, giá trị $r_{Act}(s,a)$ xác định giá trị phần thưởng có được khi hành động a được thực thi tại trạng thái s. Một cách hình thức, với đường vô hạn $\omega = s_0 \xrightarrow{(a0,\mu 0)} s_1 \xrightarrow{(a1,\mu 1)} \ldots$, phần thưởng tích lũy được trong quá trình dịch chuyển của ω từ trạng thái si đến si+1 được xác định bởi:

$$r(\omega, i) \stackrel{\text{def}}{=} \begin{cases} r_S(s_i).a_i \; nếu \; a_i \in \mathbb{R}_{\geq 0} (tương \; đương \; i \; mod \; 2 = 0) \\ r_{Act}(s_i, a_i) \; trong \; các \; trường \; hợp \; khác. \end{cases}$$

Một lựa chọn khác là có thể biểu diễn giá trị giải thưởng tại trạng thái bằng giá trị giải thưởng tại một thời điểm nhất định. Ví dụ có thể áp dụng biểu diễn này trong việc thể hiện số lượng bản tin đang nằm trong hàng đợi tại một thời điểm cụ thể. Khi sử dụng cách diễn dịch này, giá trị thưởng theo hành động sẽ không được xét tới.

Ô tô mát thời gian xác suất PTA

Ô tô mát thời gian xác suất (PTA) mô hình hóa thời gian theo cùng cách ô tô mát thời gian (cổ điển) thực hiện, đó là sử dụng đồng hồ. Đồng hồ là biến thuộc miền thời gian thực không âm, có giá trị tăng như giá trị thời gian thực. Trong các phần tiếp theo, ta giả định có một tập các đồng hồ $\mathcal{X}$. Một hàm v: $\mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ được gọi là một giá trị đồng hồ, và tập các giá trị của các đồng hồ là $\mathbb{R}_{\geq 0}^{\mathcal{X}}$. Với mọi $v \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$, $t \in \mathbb{R}_{\geq 0}$ và $X \subseteq \mathcal{X}$, gọi v+t chỉ các giá trị đồng hồ của X sau t thời gian kể từ, và v[X:=0] là tập các giá trị đồng hồ, trong đó các đồng hồ thuộc X được đặt về 0.

Tập các ràng buộc thời gian trên tập $\mathcal{X}$, ký hiệu là $CC(\mathcal{X})$, được định nghĩa bởi cú pháp:

$$\chi ::= true \mid x \leq d \mid c \leq x \mid x+c \leq y+d \mid \neg \chi \mid \chi \wedge \chi$$

Trong đó $x, y \in \mathcal{X}$ và $c, d \in \mathbb{N}$. Một giá trị thời gian v thỏa mãn một ràng buộc thời gian χ, ký hiệu $v \vDash \chi$, nếu thay các giá trị của v vào các biến đồng hồ tương ứng thì χ có giá trị *true*. Tập các giá trị thỏa mãn một ràng buộc thời gian được gọi là một vùng. Các ràng buộc thời gian sẽ được sử dụng để định nghĩa cú pháp các PTA và sử dụng trong đặc tả các tính chất của PTA.

Định nghĩa

Một ô tô mát thời gian xác suất (PTA) T là một bộ (L, l0, $\mathcal{X}$, Act, inv, enab, prob, $\mathscr{D}$), trong đó:

- L là tập hữu hạn các vị trí

- $l0 \in L$ là vị trí ban đầu
- $\mathcal{X}$ tập hữu hạn các đồng hồ, có giá trị thực không âm.
- Act là tập hữu hạn các hành động
- inv: $L \rightarrow CC(\mathcal{X})$ là hàm điều kiện ràng buộc
- enab: L x Act $\rightarrow$ CC(X) là tập các điều kiện thực hiện (các hành động)
- prob: L x Act $\rightarrow$ Dist(2^{χ} x L) là hàm xác suất chuyển.
- $\mathscr{L}$: $L \rightarrow 2^{AP}$ là hàm gắn nhãn, ánh xạ mỗi vị trí với một tập các mệnh đề logic.

Một trạng thái trong PTA là một cặp $(l,v) \in L \times \mathbb{R}^{X}_{\geq 0}$ sao cho $v \vDash inv(l)$. Trong trạng thái (l,v) bất kỳ, hoặc một khoảng thời gian $t \in \mathbb{R}_{\geq 0}$ trôi qua, hoặc một hành động $a \in Act$ được thực thi. Khi lựa chọn là thời gian, giá trị t phải đảm bảo ràng buộc inv(l) được thỏa mãn liên tục trong khoảng t thời gian. Trạng thái của PTA sau quá trình chuyển này sẽ là (l, v+t), và để đơn giản, có thể ký hiệu trạng thái này là (l,v) + t. Trong trường hợp lựa chọn một hành động được thực hiện, hành động a được chọn chỉ khi nó thỏa mãn điều kiện thực hiện, cụ thể là điều kiện enab(l, a) thỏa mãn tại thời điểm v. Khi một hành động a được chọn, một tập các đồng hồ được reset và các vị trí tiếp theo sẽ được chọn ngẫu nhiên theo phân bổ xác suất prob(l,a). Ta gọi mỗi thành phần $(X, l') \in (2^{\chi} x L)$.

Ta giả sử PTA luôn chuyển đến các trạng thái thỏa mãn tiêu chuẩn, tức với mỗi trạng thái (l,v) và một hành động a sao cho v thỏa mãn enab(l,a), mọi cạnh $(X, l') \in$ edges(l,a) sẽ cho kết quả chuyển đến một trạng thái hợp lệ, tức $v[X:=0] \vDash inv(l')$. Để thỏa mãn điều kiện trên, mỗi bước chuyển PTA cần kiểm tra các ràng buộc của vị trí mới thỏa mãn điều kiện chuyển của (l,v) hiện tại.

Ngữ nghĩa của PTA

Cho P = (L, l0, $\mathcal{X}$, Act, inv, enab, prob,$\mathscr{D}$) là một PTA. Ngữ nghĩa của PTA được định nghĩa là một TPS (số trạng thái không giới hạn) TPS [P] = (S, s0, Act, $Steps_P$, lab) trong đó:

- $S = \{(l,v) \in L \times \mathbb{R}^{X}_{\geq 0} \mid v \vDash inv(l)\}$ và s0 = (l0, {Ϫ0}), với {Ϫ0} là trạng thái khởi đầu với tất cả các đồng hồ thuộc Ϫ có giá trị 0;
- Với mọi $(l,v) \in S$ và $a \in Act \cup \mathbb{R}_{\geq 0}$, ta có $Steps_P((l,v),a) = \lambda$ nếu và chỉ nếu:

+ Dịch thời gian: $a \in \mathbb{R}_{\geq 0}$, $v+t' \vDash inv(l)$ với mọi $0 \leq t' \leq a$, và $\lambda = \mu_{(l,v+a)}$

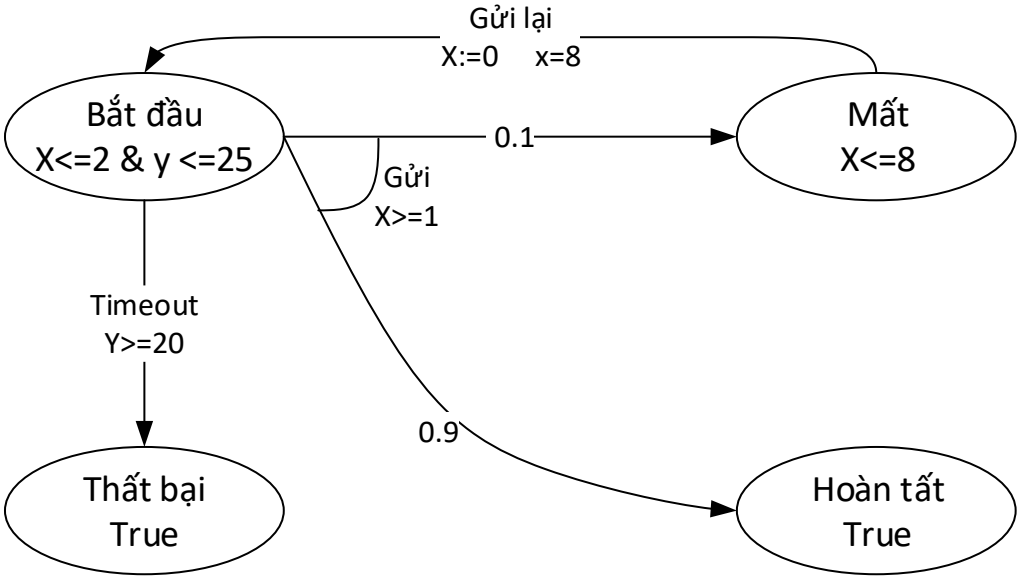
+ Thực thi hành động: $a \in Act$, $v \vDash enab(l,a)$ và với mỗi $(l',v') \in S$:

$$\lambda(l',v') = \sum\{|prob(l,a)(X,l')| \; X \in 2^{\chi} \wedge v' = v[X:=0]\}$$

- Với mọi $(l,v) \in S$ ta có $lab(l,v) = \mathscr{L}(l)$

Ví dụ về PTA

Trong Hình 3.1, một PTA thực hiện mô hình một giao thức mạng đơn giản. Các bước chuyển là các đường nối giữa các trạng thái. Phân bố xác suất được thể hiện bằng các đường nối xuất phát từ cùng một nơi, và giá trị xác suất được thể hiện trên đường nối. PTA có 2 đồng hồ x và y, được bắt đầu từ giá trị 0. Tại vị trí bắt đầu, hệ thống đợi tối thiểu 1 đơn vị thời gian (thể hiện bằng điều kiện x>=1 tại đường nối khi thực thi hành động Gửi) và tối đa 2 đơn vị thời gian (thể hiện bằng điều kiện x <=2 & y<=25) trước khi gửi một bản tin. Với khả năng 0,9; bản tin được nhận thành công (sang trạng thái Hoàn tất) và khả năng 0,1 bản tin sẽ bị mất (sang trạng thái Mất). Nếu bản tin bị mất, khi x đạt đến 8, PTA trở về trạng thái bắt đầu để chuẩn bị gửi lại bản tin. Hệ thống sẽ chuyển sang trạng thái Thất bại khi tổng thời gian từ khi bắt đầu vượt quá 20 đơn vị (nhưng không quá 25), thể hiện bằng giá trị đồng hồ y.



Hình 3.1: Minh họa một PTA

Các thưởng của PTA

Các thưởng của PTA P được định nghĩa là cặp r = (r_L, r_{Act}), trong đó r_L: L $\rightarrow \mathbb{R}_{\geq 0}$ là một hàm gán cho mỗi vị trí tốc độ tích lũy phần thưởng theo thời gian tại vị trí đó, và r_{Act}: L x Act $\rightarrow \mathbb{R}_{\geq 0}$ là một hàm gán phần thưởng cho mỗi lần thực thi hành động tại vị trí đó.

Cấu trúc thưởng tương ứng trong TPS[P] là r = (r_S, r_{Act}) trong đó $r_S(l,v) = r_L(l)$ và $r_{Act}((l,v),a) = r_{Act}(l,a)$ với $(l,v) \in L \times \mathbb{R}^X_{\geq 0}$ và $a \in Act$.

Mô hình hóa với các PTA

Các biến rời rạc

Việc mở rộng mô hình hóa các PTA với các biến rời rạc giúp thuận tiện trong quá trình mô hình hóa các hệ thống thực. Ta giới hạn việc mở rộng thêm các biến vào PTA chỉ cho số lượng hữu hạn các biến, và các biến thuộc miền giới hạn. Các điều kiện thực thi của PTA có thể tham chiếu đến các biến, và giá trị các biến có thể được cập nhật trong các hành động.

Sự khẩn cấp

Khi mô hình hóa hệ thời gian thực, có thể cần biểu diễn một hành động cần thực thi tức thì tại mộ trạng thái (không cho thời gian trôi qua tại trạng thái đó). Do vậy có thể mô hình hóa sự kiện tức thời trong hệ thống gồm một vài hành động tức thì. Một số cơ chế để mô hình hóa các tình huống này đã được giới thiệu và công bố cho ô tô mát thời gian, như trong ngôn ngữ mô tả hệ thống của công cụ UPPAAL. Dưới đây là một số cách để biểu diễn sự kiện khẩn cấp trong PTA:

- *Vị trí khẩn cấp (urgency)*: là vị trí trong đó không cho phép thời gian trôi qua. Có thể biểu diễn vị trí khẩn cấp trong PTA bằng cách thêm một đồng hồ, trong đó đồng hồ được reset khi vào vị trí, và có ràng buộc giá trị đồng hồ bằng 0 tại vị trí đó.
- *Vị trí cam kết (committed location)*: cũng là vị trí không cho phép thời gian trôi qua, nhưng phải rời khỏi vị trí này ngay khi thành phần khác của hệ thống thực hiện chuyển dịch. Mô tả vị trí cam kết trong PTA có thể thực hiện bằng cách thêm biến logic nguyên tử atom và xây dựng PTA song song. Khi PTA chuyển vào vị trí cam kết, giá trị atom được đặt là true và khi PTA rời khỏi vị trí cam kết, giá trị atom đặt là false, và mọi điều kiện thực hiện của PTA khác được thêm ràng buộc sao cho atom có giá trị false.

- *Hành động khẩn cấp*: là cách thức mô hình hóa các hành động phải được chọn ngay khi điều kiện thực thi thỏa mãn. Việc biểu diễn các hành động khẩn cấp trong cú pháp PTA bằng cách thêm tập con Act_u (của tập hành động Act) để chỉ các hành động khẩn cấp. Sự xuất hiện của các hành động khẩn cấp dẫn đến việc điều chỉnh lại định nghĩa của TPS tương ứng. Với PTA P = (L, l0, χ, Act, inv, enab, prob,$\mathscr{D}$) với các hành động khẩn cấp Act_u, PTS [P] = (S, s0, Act, $Steps_P$, lab) trong đó S, s0, lab và StepsP((l,v),a) với $(l,v) \in S$ và $a \in Act$ như trong định nghĩa Ngữ nghĩa của PTA, trong khi với $(l,v) \in L$, $t \in \mathbb{R}_{\geq 0}$, ta có StepsP((l,v),t) = $\mu_{(l,v+t')}$ nếu và chỉ nếu:
  + $v+t' \vDash inv(l)$ với mọi $0 \leq t' \leq a$
  + Với mọi $0 \leq t' \leq t$ và $a \in Act$, nếu $v+t' \vDash enab(l,a')$ thì $a' \notin Act_u$

Đặt lại đồng hồ với giá trị bất kỳ: Theo định nghĩa của PTA, các giá trị đồng hồ chỉ được thiết lập lại về 0 khi thực hiện chuyển trạng thái theo xác suất ngẫu nhiên. Tuy nhiên, trong nhiều trường hợp lại cần thiết lập lại giá trị của các đồng hồ về một số nguyên dương. Để mở rộng khái niệm PTA, bổ sung thêm việc đặt lại giá trị đồng hồ về số nguyên dương bất kỳ, định nghĩa PTA cũng được điều chỉnh tương ứng [5].

PTA với khả năng đặt lại đồng hồ bất kỳ sẽ chuyển thành PTA chuẩn với giới hạn giá trị reset của đồng hồ là 0. Tuy nhiên, khi thực hiện kiểm chứng tự động các thuộc tính của PTA, số lượng mẫu kiểm chứng (và không gian trạng thái) có thể tăng theo hàm mũ, gây khó khăn cho việc mô hình hóa và kiểm chứng. Do vậy các giải thuật kiểm chứng mô hình luôn được phát triển với khả năng đặt các đồng hồ về giá trị bất kỳ.

Tính phân kỳ của thời gian (time divergence)

Một vấn đề quan trọng liên quan đến việc kiểm chứng của các mô hình của các hệ thống thời gian thực với đó là tính phân tán của thời gian, theo đó giá trị thời gian có giá trị luôn tăng tới vô hạn. Việc biểu diễn các hành vi như vậy không phù hợp với các hệ thống thực tế, và do đó các kỹ thuật kiểm chứng được sử dụng phải có khả năng để bỏ qua hành vi như vậy trong khi phân tích mô hình.

Sử dụng các khái niệm khác nhau để biểu diễn giá trị thời gian, nhưng các nghiên cứu gần đây chỉ tập trung vào các mục tiêu ký hiệu là Adv[[P]], là những mục tiêu sao cho giá trị biết thời gian luôn vượt qua bất kỳ giá trị ràng buộc nào.

Việc xây dựng và biểu diễn các PTA có thể tạo ra các trạng thái trong đó biến thời gian không thể tiến lên bất cứ giá trị nào, các trạng thái này được gọi là khóa thời gian (timelock) trong cài đặt các ô tô mat thời gian, và được coi là lỗi trong quá trình mô hình hóa hệ thống. Các trạng thái bị khóa thời gian có thể được xác định nhờ quá trình phân tích mô hình kiểm chứng, và có thể được loại bỏ khỏi mô hình bằng cách điều chỉnh các ràng buộc về thời gian và các điều kiện chuyển.

3.2 Đặc tả tính chất cho các PTA (properties specification for PTAs)

Trong phần này, đề tài trình bày logic thời gian mô tả bằng ngôn ngữ đặc tả chính thức các tính chất định lượng của PTA. Dựa trên cơ sở cây logic tính toán xác suất PCTL, với PCTL là phần mở rộng xác suất của logic CTL đã được đề xuất cho các đặc tả tính chất cho cả chuỗi Markov xác định MDP và chuỗi Markov thời gian rời rạc. Các biểu diễn logic được bổ sung thêm các toán tử cho các phần thưởng, theo cùng cách biểu diễn logic trong MDP, và được sử dụng trong các mô hình kiểm chứng xác suất PRISM.

Định nghĩa:

Cú pháp logic mô tả tính chất PTA được cho bằng các văn phạm sau:

$$\phi \quad ::= \quad \text{true} \mid a \mid \chi \mid \phi \wedge \phi \mid \neg\phi \mid P_{\bowtie q}\,[\Psi] \mid R^{r}_{\bowtie q}\,[\rho]$$

$$\Psi \quad ::= \quad \phi\ U^{\leq k}\ \phi \mid \phi\ U\ \phi$$

$$\rho \quad ::= \quad I^{=k} \mid C^{\leq k} \mid F\ \phi$$

Trong đó $a \in AP$ là mệnh đề nguyên tử, $\chi \in CC(\chi)$ là một ràng buộc thời gian, toán tử so sánh $\bowtie \in \{\leq, <, \geq, >\}$, $p \in \mathbb{Q} \cap [0,1]$, $q \in \mathbb{Q}_{\geq 0}$, r là cấu trúc thưởng và $k \in \mathbb{N}$.

Đây là các mệnh đề logic mở rộng với các toán tử logic xác suất (P) và các toán tử phần thưởng (R). Diễn giải không chính thức, tính chất được biểu diễn dạng $P_{\bowtie q}\,[\Psi]$ cho biết xác suất để công thức Ψ là đúng luôn thỏa mãn ràng buộc $\bowtie q$. Tính chất biểu diễn dạng $R^{r}_{\bowtie q}\,[\rho]$ cho biết giá trị kỳ vọng của hàm giải thưởng ρ trong cấu trúc thưởng r thỏa mãn ràng buộc $\bowtie q$.

Các công thức trong logic luôn là công thức tại trạng thái. Các công thức này được tính toán trên các trạng thái của PTA P (hay chính xác hơn là trên các trạng thái của TPS [P] biểu diễn ngữ nghĩa của PTA P). Với trạng thái s và công thức ϕ, ta viết $s \vDash \phi$ để biểu diễn ϕ được thỏa mãn tại trạng thái s. Ngữ pháp của logic cũng bao gồm công thức trên đường Ψ và toán tử giải thưởng (ρ), được xuất hiện như tập con của các toán tử P và R.

Đặc tả tính chất của PTA được bổ sung thêm 2 công thức: ràng buộc theo thời gian (công thức có dạng $\phi_1 \, U^{\leq k} \, \phi_2$) và không ràng buộc thời gian (công thức có dạng $\phi_1 \, U \, \phi_2$). $\phi_1 \, U \, \phi_2$ có nghĩa là một trạng thái s thỏa mãn ϕ_2 và mọi thời gian trước đó thì ϕ_1 được thỏa mãn. $\phi_1 \, U^{\leq k} \, \phi_2$ có ý nghĩa tương tự, nhưng yêu cầu sự kiện ϕ_2 thỏa mãn phải trước thời điểm k. Một số toán tử hữu ích cũng có thể được bổ sung, bao gồm $F \, \phi \equiv true \, U \, \phi$, có nghĩa ϕ sẽ được thỏa mãn, và $F^{\leq k} \, \phi \equiv true \, U^{\leq k} \, \phi$ cho biết ϕ sẽ được thỏa mãn trước thời điểm k. Ta cũng có $G \, \phi \equiv \neg(F \, \neg\phi)$, cho biết ϕ luôn thỏa mãn, và $G^{\leq k} \, \phi \equiv \neg(F^{\leq k} \, \neg\phi)$ cho biết ϕ được thỏa mãn liên tục đến thời gian k.

Toán tử giải thưởng $R^r_{\bowtie q}[\rho]$, tương ứng ρ với các giá trị I=k tham chiếu đến giá trị phần thưởng tại thời điểm tức thời k, C≤k tham chiếu đến tổng phần thưởng tích lũy đến thời điểm k, và $F \, \phi$ tham chiếu đến tổng giải thưởng tích lũy đến khi đạt đến một trạng thái thỏa mãn ϕ. Một cách hình thức, các ngữ nghĩa của logic được định nghĩa như sau:

Định nghĩa

Cho P là một PTA, [P] = (S, s0, Act, $Steps_P$, lab) là một ngữ nghĩa của P, và r là cấu trúc phần thưởng trên [P], tương ứng với cấu trúc phần thưởng trên P. Cho trạng thái s=(l,v) ∈ S, quan hệ thỏa mãn ⊨ được định nghĩa quy nạp như sau:

$s \models true$ — s luôn luôn đúng

$s \models a \iff a \in lab(s)$

$s \models \chi \iff v \models \chi$

$s \models \phi 1 \wedge \phi 2 \iff s \models \phi 1 \wedge s \models \phi 2$

$s \models \neg\phi \iff s \not\models \phi$ (s không thỏa mãn ϕ)

$s \models P_{\bowtie q}[\Psi] \iff Pro^{\sigma}_{[P],s}(\{\omega \in Path^{\sigma}_{[P],s} \mid \omega \models \Psi\}) \bowtie \rho$

với mọi $\sigma \in Adv[P]$

$s \models R^r_{\bowtie q}[\rho] \iff \mathbb{E}^{\sigma}_{[P],s}(rew(r, \rho)) \bowtie q$ với mọi $\sigma \in Adv[P]$

trong đó:

$\omega \models \phi_1 \, U^{\leq k} \, \phi_2 \iff$ tồn tại một vị trí (i,t) của ω sao cho $\omega(i) + t \models \phi_2$ và $dur_\omega(i)+t \leq k$ và $\omega(i) + t' \models \phi_1 \vee \phi_2$ với mọi vị trí (j,t') trước (i,t) của

ω

$\omega \vDash \phi_1 \, U \, \phi_2 \quad \Leftrightarrow$ tồn tại một vị trí (i,t) của ω sao cho $\omega(i) + t \vDash \phi_2$ và $\omega(i) + t' \vDash \phi_1 \vee \phi_2$ với mọi vị trí (j,t') trước (i,t) của ω

Một số ví dụ về các tính chất thường được kiểm chứng của PTA có dạng thức logic được thể hiện như sau:

- $P_{\geq 0,8}[F^{\leq k} \, ack_n]$ – "Xác suất bên gửi nhận được n ack trong k đơn vị thời gian tối thiểu là 0.8"
- Trigger $\rightarrow P_{<0.0001}[G^{\leq 20} \neg deploy]$ – "Xác suất để túi khí không bung ra trong vòng 20 mili giây chắc chắn nhỏ hơn 0.0001"
- Pmax = ?[¬sent U fail] – "Xác suất tối đa khi lỗi xảy ra trước khi bản tin gửi thành công là bao nhiêu"
- R^{time}_{max} = ? [F end] – "Xác suất tối đa giao thức có thể hoàn tất"
- $R^{pwr}_{<q}$ [$C^{\leq 60}$] – "Tổng mức năng lượng tiêu thụ trong 60 giây đầu tiên <q"

Giản lược các thuộc tính

Ta cùng xét cách thức thực hiện giản lược các thuộc tính nhằm giảm mức độ phức tạp của các biểu thức logic để kiểm chứng trong các công thức đơn giản hơn trên các PTA mở rộng. Xét thuộc tính có giới hạn thời gian $P_{\bowtie p}$ [$\phi_1 \, U^{\leq k} \, \phi_2$] trên PTA P. Nếu mở rộng P bằng cách thêm vào đồng hồ z, khi đó một trạng thái (l,v) thuộc P thỏa mãn công thức $P_{\bowtie p}$ [$\phi_1 \, U^{\leq k} \, \phi_2$] nếu và chỉ nếu trạng thái (l,v') của PTA mở rộng có v'(x) = v(x) với mọi đồng hồ x của P và v'(z) = 0 thỏa mãn $P_{\bowtie p}$ [$\phi_1 \, U^{\leq k} \, \phi_2$]. Với thuộc tính không giới hạn thời gian $P_{\bowtie p}$ [$\phi_1 \, U \, \phi_2$], ta có thể chỉnh sửa PTA sao cho khi đạt đến một trạng thái không thỏa mãn ϕ_1, chỉ các bước chuyển tới các trạng thái không thỏa mãn được thực hiện, khi đó một trạng thái trong PTA thỏa mãn $P_{\bowtie p}$ [$\phi_1 \, U \, \phi_2$] nếu và chỉ nếu trạng thái của PTA mới thỏa mãn $P_{\bowtie p}$ [$F \, \phi_2$].

Tiếp theo, ta cùng xem xét cách thức làm giảm mức độ phức tạp của kiểm chứng dạng $P_{\geq p}$ [$F^{\leq k} \, \phi$] thành công thức dạng $P_{\leq 1-p}$ [$F \, a_{exec}$], do việc xác định giá trị xác suất lớn nhất trong nhiều trường hợp dễ thực hiện hơn việc tính xác suất nhỏ nhất. Ta mở rộng P bằng cách thêm đồng hồ z và thay đổi sao cho các trạng thái thỏa mãn ϕ bắt buộc phải thực hiện các phép chuyển tới trạng thái không thỏa mãn, và trong mọi trạng thái khác, ta bổ sung thêm trạng thái mới không thỏa mãn ϕ, exceeded, được thực hiện khi z>k. Do đó $P_{\geq p}$ [$F^{\leq k} \, \phi$] là đúng trong trạng thái được mô tả ban đầu nếu và chỉ nếu các

trạng thái tương ứng của PTA mở rộng với z=0 phải thỏa mãn $P_{\leq 1-p}$ [F a_{exec}], trong đó a_{exec} chỉ đúng tại vị trí exceeded.

Cuối cùng, ta cùng xem xét các phương pháp để giảm các thuộc tính dạng $R^r_{\bowtie q}$ [$C^{\leq k}$] hoặc $R^r_{\bowtie q}$ [$I^{=k}$] về dạng $R^r_{\bowtie q}$ [F ϕ]. Trong cả hai trường hợp, ta bổ sung thêm đồng hồ z vào PTA P. Với $R^r_{\bowtie q}$ [$C^{\leq k}$], chỉ cần kiểm tra $R^r_{\bowtie q}$ [F (z=k)] trên PTA mở rộng. Với $R^r_{\bowtie q}$ [$I^{=k}$], ta bổ sung thêm điều kiện z≤k vào tất cả các ràng buộc và chuyển chúng tới các vị trí không thỏa mãn và thêm bước z=k vào tất cả các vị trí trong khi thay đổi các điều kiện thực hiện chuyển sao cho nó không kích hoạt việc chuyển khi z=k. Khi đó chỉ cần kiểm chứng tính chất $R^r_{\bowtie q}$ [F a_{done}].

3.3 Các phương pháp kiểm chứng tự động PTA

Với bài toán tổng quát về kiểm chứng các tính chất Φ của PTA P, ví dụ xác định Sat(Φ) = {s∈S| s⊨ Φ }: tập các trạng thái S của P thỏa mãn tính chất P. Hiện đã có nhiều kỹ thuật kiểm chứng PTA được công bố, và các kỹ thuật này hỗ trợ các loại biểu thức logic khác nhau. Ta cùng xem các kỹ thuật kiểm chứng mô hình với PTA đã được nghiên cứu và công bố.

- Đồ thị miền
- Đồ thị miền biên
- Phương pháp đồng hồ số
- Phương pháp đạt được lùi
- Trừu tượng hóa và làm mịn với trò chơi ngẫu nhiên

Hai kỹ thuật đầu tiên nêu trên dựa trên khái niệm Đồ thị miền, ban đầu được xây dựng cho việc đánh giá khả năng quyết định và mức độ phức tạp của bài toán kiểm chứng mô hình hơn là để triển khai thực hiện kiểm chứng. Các phương pháp khác khá hiệu quả trong việc kiểm chứng mô hình cho một số loại biểu thức logic cụ thể. Bảng 3.1 thể hiện khả năng kiểm chứng của các phương pháp khác nhau với các loại biểu thức logic, chi tiết được nêu trong tài liệu [3].

Bảng 3.1 : Quy mô tính toán khi DATA = 10..30; RETRY = 0..4

Loại biểu thức	Đồ thị miền	Đồ thị miền biên	Đồng hồ số (với các PTA đóng)	Phương pháp đạt được lùi	Trò chơi ngẫu nhiên
Biểu thức đơn $P_{\bowtie q}$ [Ψ]	✓	✓	✓	✓	✓

Biểu thức đơn $R^r_{\bowtie q}$ [ρ]	X	✓	✓	Đang mở	Đang mở
Biểu thức không chứa $R^r_{\sim q}$[ρ]	✓	✓	X	✓	Đang mở
Logic đầy đủ	X	X	X	Đang mở	Đang mở

Trừ phi được nêu cụ thể các ngoại lệ, ta giả thiết PTA P không có khóa thời gian và được cấu trúc sao cho đảm bảo thời gian không giới hạn.

3.3.1 Xây dựng đồ thị miền (region graph contruction)

Đồ thị miền cho một PTA được xây dựng dựa trên việc xây dựng ô tô mát thời gian truyền thống. Kỹ thuật này cung cấp một phương pháp kiểm chứng mô hình cho các mệnh đề logic không chứa mệnh đề logic $R^r_{\bowtie q}$ [ρ]. Đồ thị miền của một PTA P và công thức ϕ có dạng một MDP hữu hạn trạng thái, trong đó các trạng thái là các vùng dạng (l, α), với l là vị trí và α là một lớp tương đương với các giá trị đồng hồ theo công thức tương đương như định nghĩa phía dưới. Cho c là giá trị hằng số lớn nhất mà mọi đồng hồ của P phải so sánh trong các ràng buộc thời gian của P và ϕ. Khi đó các giá trị đồng hồ v và v' là tương đương khi và chỉ khi chúng thỏa mãn các điều kiện sau:

- Với mọi x ∈ $\mathcal{X}$, thỏa mãn hoặc v(x) > c và v'(x) > c, hoặc v(x) và v'(x) có giá trị phần nguyên bằng nhau.
- Với mọi x, x' ∈ $\mathcal{X}$, hoặc v(x) - v(x') > c và v'(x) - v'(x')>c, hoặc v(x) - v(x') và v'(x) – v'(x') có giá trị phần nguyên bằng nhau.

Trong đó hai biến q, q'∈$\mathbb{R}_{\geq 0}$ có giá trị phần nguyên bằng nhau khi $\lfloor q \rfloor = \lfloor q' \rfloor$ và $\lfloor q \rfloor$ – q = 0 nếu và chỉ nếu $\lfloor q' \rfloor$ - q' = 0. Lưu ý là các giá trị đồng hồ tương đương sẽ cùng thỏa mãn các ràng buộc thời gian trong PTA. Tập các vùng cần cho Đồ thị miền, ký hiệu R, là tập các vùng (l, α) sao cho tồn tại v ∈ α sao cho v ⊨ inv(l). Kích thước của R bị giới hạn bởi giá trị $|L|.(2c+2)^{(|\mathcal{X}|+1)^2}$ (theo tài liệu tham khảo số 23 trong [3], nêu tại mục Region graph contruction).

Một vùng (l, α) ∈ R có thể có điểm thời gian kế nhiệm, được định nghĩa như sau. Nếu v+t ∈ α với mọi v ∈ α và t ∈ $\mathbb{R}_{\geq 0}$, khi đó điểm thời gian kế nhiệm của (l, α) là chính (l, α). Ngược lại, tồn tại một vùng duy nhất (l, β) ≠ (l, α) sao cho tồn tại v ∈ α và t ∈ $\mathbb{R}_{\geq 0}$ sao cho v+t ∈ β và v+t' ∈ α ∪ β với

mọi $0 \le t' \le t$. Ngoài ra, ta có $v+t' \vDash inv(l)$ với mọi $\le t' \le t$ thì (l, β) là kế nhiệm thời gian của (l, α), ngược lại (l, α) không có kế nhiệm thời gian.

Đồ thị miền cho P và ϕ là một MDP $Reg(P, \phi) = (R, (l0, [\mathbf{0}], Act \cup \{\tau\}, Steps, lab)$, trong đó với mỗi $(l, \alpha) \in R$ và $a \in Act \cup \{\tau\}$, ta có $Steps((l, \alpha), a) = \lambda$ nếu và chỉ nếu một trong các khả năng sau xảy ra:

- Dịch chuyển thời gian: $a = \tau$, $\lambda = \mu_{(l, \beta)}$ và (l, β) là kế nhiệm thời gian của (l, α)
- Thực hiện hành động: $a \in Act$, tồn tại $v \in \alpha$ với $v \vDash enab(l, a)$ và, với mỗi $(l', \beta) \in R$:

  $$\lambda(l', \beta) = \sum\{|prob(l,a)(X,l') \mid X \in 2^{x} \wedge \beta = \alpha[X:=0]\};$$

  và $lab(l, \alpha) = L(l)$ với mọi $(l, \alpha) \in R$.

Việc kiểm chứng mô hình của $Reg(P, \phi)$ thực hiện theo phương pháp đệ quy, trong đó tính từng biểu thức thành phần ϕ' của ϕ và tính $Sat(\phi')$. Để đơn giản, ta giả thiết P có thời gian phân kỳ. Xác định các trạng thái thỏa mãn các mệnh đề nguyên tử, các ràng buộc thời gian hoặc các liên hệ logic và các thuộc tính ràng buộc thời gian được thực hiện bằng cách giản lược các thuộc tính như và tạo các PTA mở rộng. Xét các công thức dạng $P_{\bowtie p}[\phi 1 \ U \ \phi 2]$. Nếu tập các trạng thái thỏa mãn $\phi 1$ và $\phi 2$ đã được tính và các vùng tương ứng $Reg[P, \phi)$ với các tập này được gán nhãn a1, a2, khi đó các trạng thái thuộc [P] thỏa mãn $P_{\bowtie p}[\phi 1 \ U \ \phi 2]$ đơn giản là những trạng thái thuộc các vùng tương ứng với các trạng thái trong MDP $Reg[P, \phi]$ mà thỏa mãn $P_{\bowtie p}[a1 \ U \ a2]$.

Như vậy Đồ thị miền tạo ra giải thuật cho kiểm chứng mô hình PTA thời gian phân kỳ, với các biểu thức không chứa các toán tử dạng $R^{r}_{\bowtie q}[.]$. Giải thuật có thời gian thực hiện theo hàm mũ, do việc kiểm chứng các tính chất dạng $P_{\bowtie p}[a1 \ U \ a2]$ trên các MDP có dạng đa thức, và kích thước của Đồ thị miền $Reg[P, \phi]$ là hàm mũ theo kích thước của P (kích thước của P là tổng số vị trí và đồng hồ, kích thước của mã hóa nhị phân các hằng số trong các ràng buộc và điều kiện chuyển, và kích thước của mã hóa các xác suất chuyển khi biểu diễn dưới dạng tỉ số của các số nguyên).

3.3.2 Đồ thị miền biên (boundary region graph)

Phương pháp xây dựng Đồ thị miền như trên không hiệu quả khi kiểm chứng các tính chất có chứa phần thưởng. Cụ thể, với một vùng (l, α), giá trị $\mathbb{E}^{r,min}_{P,s}(\rho)$ và $\mathbb{E}^{r,max}_{P,s}(\rho)$ nói chung không thuộc cùng một trạng thái $s \in (l, \alpha)$. Ta mô tả ngắn ngọn việc tổng quát hóa Đồ thị miền, gọi là Đồ thị miền biên, là một MDP hữu hạn trạng thái với cấu trúc thưởng trong đó ta có thể quyết định xem liệu một trạng thái s của một PTA có thể thỏa mãn biểu thức $R^{r}_{\bowtie q}[F$

ϕ], với phạm vi giới hạn bài toán là không có biểu thức logic chứa trong $R^r_{\bowtie q}[.]$ và giới hạn các giá trị của ⋈ là [≤, ≥]. Với các tính chất $R^r_{\bowtie q}[C^{\leq k}]$ và $R^r_{\bowtie q}[I^{=k}]$, phương pháp giản lược có thể sử dụng với các phạm vi giới hạn tương tự.

Tư tưởng của phương pháp Đồ thị miền biên là các hành vi tối ưu (tương ứng với giá trị giải thưởng lớn nhất hoặc nhỏ nhất) tương ứng với các tình huống chuyển dịch của PTA khi các lớp giá trị tương đương của đồng hồ có ít nhất một giá trị là số nguyên, hoặc gần với giới hạn của giá trị đồng hồ. Ngoài ra, giá trị giới hạn chính xác của các đồng hồ được xác định bởi giá trị của đồng hồ tại trạng thái s, và phải được thể hiện trong Đồ thị miền biên. Cấu trúc của các giải thưởng trong Đồ thị miền biên cũng được thừa kế trực tiếp từ các cấu trúc giải thưởng của PTA. Từ Đồ thị miền biên (là một MDP hữu hạn trạng thái), ta có thể tính toán các giá trị lớn nhất hoặc nhỏ nhất của các giải thưởng tích lũy trong tập mục tiêu xác định. Ta có thể tính các giá trị $\mathbb{E}^{r,min}_{P,s}(F\ \phi)$ và $\mathbb{E}^{r,max}_{P,s}(F\ \phi)$ và từ đó quyết định xem s có thỏa mãn $R^r_{\bowtie q}[F\ \phi]$ hay không.

3.3.3 Phương pháp đồng hồ số (digital clock method)

Các phương pháp Đồ thị miền không được cài đặt trong các ứng dụng thực tế do kích thước miền không gian quá lớn. Đã có nhiều kỹ thuật kiểm chứng PTA được phát triển. Ta xét phương pháp đồng hồ số, trong đó giới hạn các ngữ nghĩa thời gian liên tục của PTA sao cho mọi dịch chuyển thời gian luôn có giá trị là 1 đơn vị. Do vậy các giá trị của đồng hồ là các số nguyên (thay vì số thực). Từ ràng buộc này và giá trị hằng số c_x đã biết, là giá trị hằng số lớn nhất mà tất cả các đồng hồ phải so sánh trong P và các thuộc tính ϕ, ta có thể xây dựng một MDP hữu hạn trạng thái để biểu diễn PTA.

Phương pháp đồng hồ số có thể áp dụng để kiểm chứng các tính chất dạng $P_{\bowtie p}[\Psi]$ và $R^r_{\bowtie q}[\rho]$ mà không có chứa các biểu thức dạng $P_{\bowtie p}[\Psi]$ và $R^r_{\bowtie q}[\rho]$ trong các biểu diễn công thức của Ψ và ρ. Nó có thể sử dụng để kiểm chứng trong các trạng thái với các giá trị đồng hồ so sánh với các số nguyên. Tính chính xác của phương pháp đồng hồ số cũng dựa trên giả thiết P và ϕ là đóng, theo nghĩa tất cả cả các ràng buộc rạng x≤d hoặc d≤x được chứa trong một số chẵn các phép phủ định. Ngoài ra, tất cả các ràng buộc thời gian và điều kiện chuyển của P được giả định không có liên hệ đường chéo, theo nghĩa các ràng buộc rạng x+c≤y+d không được chấp nhận.

Với một giá trị của đồng hồ $v \in \mathbb{N}^{\mathcal{X}}$, ký hiệu v⊕1 là giá trị đồng hồ sao cho $(v \oplus 1)(x) = \min\{v(x) + 1, c_x + 1\}$ với mọi $x \in \mathcal{X}$. Ngữ nghĩa đồng hồ của

P và ϕ được định nghĩa như ngữ nghĩa chuẩn, ngoại trừ giới hạn thời gian chuyển có giá trị là 1, và mỗi giá trị đồng hồ x có thể tăng tới giá trị tối đa c_x+1. Một cách hình thức, ngữ nghĩa đồng hồ số của một PTA P đóng được định nghĩa là một MDP hữu hạn trạng thái Dgt(P, ϕ) = (S, (l0, 0), Act $\cup$ {l}, Steps, lab) trong đó:

- $S = \{(l,v) \in L \times \mathbb{N}^{\mathcal{X}} \mid v \vDash inv(l) \wedge (\forall x \in \mathcal{X}.\ v(x) \leq c_x + 1)\}$;
- Steps((l,v),a) = λ nếu và chỉ nếu:
  + Dịch chuyển thời gian: a=1, $v \oplus 1 \vDash inv(l)$ và $\lambda = \mu(l, v \oplus 1)$;
  + Thực thi hành động: $a \in Act$, $v \vDash inv(l, a)$ và, với mọi $(l', v') \in S$
    $\lambda(l',v') = \Sigma\{|prob(l,a)(X, l') \mid X \in 2^{\mathcal{X}} \wedge v' = v[X:=0]\ |\}$;
- $lab(l,v) = \mathcal{L}(l)$ với mọi $(l,v) \in S$.

Số lượng các trạng thái trong ngữ nghĩa đồng hồ số được giới hạn bởi giá trị $|L|.\Pi_{x \in \mathcal{X}}(c_x+1)$.

Việc kiểm chứng các tính chất có dạng $P_{\bowtie p}[\Psi]$ và $R^{r}_{\bowtie q}[\rho]$ thực hiện trực tiếp trên MDP hữu hạn trạng thái. Với công thức $P_{\bowtie p}[\Psi]$, việc tính toán thực hiện tương tự như Đồ thị miền đã nêu. Với công thức $R^{r}_{\bowtie q}[F\ \phi]$ và cấu trúc thưởng $r=(r_{Act}, r_L)$ của PTA, ta xây dựng cấu trúc thưởng mới $r'=(r'_S, r'_{Act})$ trong đó $r'_S(l,v) = 0$, $r'_{Act}((l,v),l) = r_L(l)$ và $r'_{Act}((l,v),a) = r_{Act}(l,a)$ với mọi $(l,,v) \in S$ và $a \in Act$. Sau đó sử dụng các giải thuật của MDP để tính giá trị kỳ vọng lớn nhất và nhỏ nhất của giải thưởng khi đạt đến tập thỏa mãn Sat(ϕ). Trường hợp $\rho = C^{\leq k}$ và $\rho = I^{=k}$, sử dụng phương pháp giản lược thuộc tính để tính.

3.3.4 Phương pháp đạt được lùi (backward reachability)

Phương pháp đạt được lùi không áp dụng cho các tính chất có dạng $R^{r}_{\bowtie q}[\rho]$. Phương pháp này dựa trên thủ tục tìm vị trí tiền nhiệm, từ tập các trạng thái S', liệt kê danh sách các trạng thái từ đó có thể đến S' bằng cách để thời gian trôi hoặc thực thi hành động. Tập các trạng thái được biểu diễn là các trạng thái biểu tượng, $z = (l, \zeta)$, bao gồm vị trí l và một ràng buộc thời gian ζ trên tập $\mathcal{X}$, biểu diễn tập các trạng thái $\{(l,v) \mid v \vDash \zeta\}$.

Trước hết, thực thi thủ tục tìm vị trí tiền nhiệm thực hiện tham số hóa các hành động và các cạnh của PTA. Khi duyệt qua các nhánh, biểu đồ về các node được xây dựng và các node tạo nên các trạng thái biểu tượng, và mỗi cạnh được bổ sung từ trạng thái biểu tượng z tới trạng thái biểu tượng z' nếu z được sinh ra từ z' bằng thủ tục tìm vị trí tiền nhiệm. Cạnh (z, z') được đánh nhãn tương ứng với hành động trong cạnh PTA. Các trạng thái biểu tượng được sinh ra không phải là thành phần của không gian trạng thái PTA.

Sau khi việc duyệt thủ tục tìm tiền nhiệm kết thúc, biểu đồ thu được có thể được sử dụng để xây dựng nên một MDP hữu hạn trạng thái. Để xây dựng các hàm chuyển trạng thái xác suất, thông tin phải được tổng hợp từ nhiều trạng thái biểu tượng để có thông tin chính xác về các cạnh PTA (ứng với một hành động cụ thể của PTA). Việc này thực hiện bằng cách tính các điểm giao của các trạng thái biểu tượng có ít nhất một cạnh đi ra khỏi trạng thái có chung nhãn, khi đó bổ sung cạnh tương ứng vào trạng thái biểu tượng mới tạo.

Cách tiếp cận như trên chỉ áp dụng được với các thuộc tính dạng $P_{\bowtie p}[\Psi]$ trong đó $\bowtie \in \{\leq, <\}$, chẳng hạn như phép tính xác suất lớn nhất trong PTA. Với các trường hợp $\bowtie \in \{\geq, >\}$, phương pháp tính cần phải điều chỉnh, và chỉ áp dụng cho các PTA thời gian không phân kỳ. Chi tiết trong tài liệu tham khảo [6].

3.3.5 Làm mịn trừu tượng với trò chơi ngẫu nhiên (abstraction refinement with stochastic games)

Phương pháp làm mịn trừu tượng với trò chơi ngẫu nhiên có thể kiểm chứng các tính chất dạng $P_{\bowtie p}[\Psi]$ bằng cách tính xác suất đạt đến trạng thái nào đó. Phương pháp này sử dụng khái niệm trò chơi dựa trên trừu tượng hóa các đầu vào của MDP và áp dụng các kỹ thuật làm mịn. Phương pháp này có thể áp dụng với các PTA có các ràng buộc không liên hệ đường chéo, do một trong các thủ tục thực hiện là thăm dò khả năng đạt đến phía trước, và các thủ tục này chỉ có thể thực hiện khi PTA không có ràng buộc đường chéo.

Ý tưởng triển khai là xây dựng một MDP trừu tượng, với dạng trò chơi hai người. việc khái quát hóa MDP để thể hiện có hai dạng không xác định khác nhau, mỗi dạng được điều khiển bởi một người chơi khác nhau. Trong phương pháp này, người chơi một điều khiển các quyết định trong MDP trừu tượng, và người chơi hai điều khiển các quyết định trong MDP ban đầu. Một giải thuật tối ưu xác suất trong trò chơi ngẫu nhiên (ví dụ xác suất tối đa người chơi 1 đạt mục tiêu, trong khi người chơi 2 đang cố hạn chế nó) có thể tạo ra các giới hạn dưới và giới hạn trên cho xác suất đạt đến của MDP ban đầu.

Cơ chế làm mịn trừu tượng được giới thiệu trong [9] cung cấp cách thức tạo ra trò chơi trừu tượng tự động, bằng cách duyệt qua các lớp trừu tượng đến khi giới hạn trên và giới hạn dưới có độ lệch dưới ngưỡng ε xác định. Kỹ thuật này có thể giúp xác định xác suất (lớn nhất hoặc nhỏ nhất) của PTA với trạng thái nào đó. Trước hết, xây dựng biểu đồ chạm tới cho các trạng thái

(reachability graph), dựa trên các thủ tục duyệt các trạng thái tiếp theo (có thể chạm tới được từ trạng thái hiện tại) bằng cách thực thi hành động hoặc để thời gian trôi. Từ đây, một trò chơi ngẫu nhiên trừu tượng đã được tạo ra thông qua các trạng thái biểu tượng cho đến khi các xác suất cần tính toán đạt đến giá trị yêu cầu.

3.3.6 So sánh các phương pháp kiểm chứng

Phần này tóm tắt việc triển khai ứng dụng trong thực tế các phương pháp kiểm chứng mô hình nêu trên. Về khả năng áp dụng, hiện chỉ có phương pháp đồng hồ số là có thể tính toán giá trị giải thưởng, nhưng chỉ áp dụng cho các PTA không chứa ràng buộc đường chéo và áp dụng với các ràng buộc đóng. Phương pháp trừu tượng hóa với trò chơi ngẫu nhiên chỉ có thể áp dụng với các PTA không chứa ràng buộc đường chéo, và Phương pháp đạt được lùi đòi hỏi việc hiệu chỉnh (tương đối nhiều) để tính toán xác suất nhỏ nhất.

Về hiệu suất và khả năng mở rộng, phương pháp đồng hồ số đã được chứng minh hoạt động tốt trong thực tế, nhưng hiệu năng hạn chế khi có số lượng lớn các đồng hồ hoặc có nhiều hằng số xuất hiện trong ràng buộc thời gian; trong những tình huống này, các phương pháp còn lại đã cho thấy hoạt động hiệu quả hơn. Các kỹ thuật tổng hợp thông số như trong [3] có thể sử dụng để giảm số lượng các hằng số trong các ràng buộc thời gian của các lớp con của PTA, và cho các tính chất xác suất không có ràng buộc thời gian. Một kỹ thuật khác cũng được áp dụng để tăng hiệu năng là sử dụng cài đặt biểu tượng (dựa trên biểu đồ quyết định nhị phân). Cài đặt ban đầu của Phương pháp đạt được lùi cho thấy nó tạo ra MDP tương đối nhỏ, tuy nhiên giải thuật này có chi phí triển khai lớn. Kết quả thực tế của phương pháp làm mịn trừu tượng với trò chơi ngẫu nhiên cho thấy hiệu năng tốt hơn cả. Tuy nhiên, các phương án tối ưu sau đó được đưa ra cho Phương pháp đạt được lùi giúp cải thiện hiệu năng đáng kể, và cải thiện cho phương pháp làm mịn trừu tượng với trò chơi ngẫu nhiên trong rất nhiều trường hợp.

3.3.7 Các cài đặt thực tế và công cụ hỗ trợ

Với sự quan tâm tới việc kiểm chứng các hệ thời gian thực xác suất ngày càng tăng trong thời gian gần đây, đã có một số phần mềm khác nhau đã được phát triển. Công cụ kiểm chứng mô hình xác suất PRISM [11], xuất phát từ công cụ kiểm chứng xác suất xích Markov và MDP, hiện đã hỗ trợ mô hình PTA với các phương pháp kiểm chứng đồng hồ số, Phương pháp đạt được lùi và phương pháp trừu tượng hóa dùng trò chơi ngẫu nhiên. Công cụ thứ hai là MCPTA, áp dụng phương pháp đồng hồ số để dịch từ tập một tập con của ngôn ngữ mô hình hóa Modest trực tiếp sang ngôn ngữ PRISM.

Fortuna là một công cụ tập trung vào PTAs mở rộng với các chi phí (được gọi là các phần thưởng trong một số tài liệu nghiên cứu và trong tài liệu luận văn này). Nó triển khai giải thuật tính xác suất lớn nhất để đạt tới một đích trong khi tính lũy kế các giải thưởng trong phạm vi một ngưỡng định trước. Do đây là giải thuật tổng quát hóa từ Phương pháp đạt được lùi, việc tính toán xác suất đạt tới lớn nhất cũng được hỗ trợ. Một số phương án tối ưu cho giải thuật ban đầu đã được cài đặt. Cuối cùng, công cụ nổi tiếng kiểm chứng ô tô mát thời gian UPPAAL đã có bản mở rộng UPPAAL PRO, bằng cách phân tách dần dần các không gian trạng thái, xây dựng và giải quyết các MDP hữu hạn theo từng bước.

3.4 Công cụ kiểm chứng mô hình PRISM

3.4.1 Giới thiệu công cụ PRISM

PRISM là một bộ công cụ kiểm tra mô hình xác suất, công cụ để mô hình hóa và phân tích hệ thống thể hiện hành vi ngẫu nhiên hoặc xác suất chính quy, đã được sử dụng để phân tích các hệ thống từ nhiều lĩnh vực ứng dụng khác nhau, bao gồm cả giao thức truyền thông đa phương tiện, thuật toán phân phối ngẫu nhiên, các giao thức bảo mật, các hệ thống sinh học và nhiều loại khác.

PRISM có thể xây dựng và phân tích các dạng của mô hình xác suất, bao gồm DTMC, CTMC, MDP, và PTA.

Ngoài ra, PRISM có thể bổ sung các phần mở rộng để kiểm chứng các mô hình này với bộ đặc tính chi phí / lợi ích, được gọi chung là giải thưởng (từ khóa sử dụng trong PRISM là reward – giải thưởng).

Các mô hình được mô tả bằng ngôn ngữ PRISM, là ngôn ngữ dựa trên trạng thái đơn giản. PRISM cung cấp hỗ trợ cho việc phân tích tự động của một loạt các đặc tính định lượng của các mô hình này, ví dụ "Xác suất của một thất bại gây ra hệ thống để đóng cửa trong vòng 4 giờ là gì?", "Xác suất xảy ra trường hợp xấu nhất của giao thức bị ngắt do lỗi, trên tất cả các cấu hình ban đầu có thể?", "Kích thước dự kiến của hàng đợi tin nhắn sau 30 phút là bao nhiêu?", hoặc "Dự kiến thời gian thực hiện và chấm dứt các thuật toán trong trường hợp xấu nhất? ".

Các ngôn ngữ được sử dụng để đặc tả thuộc tính kết hợp logic PCTL, CSL, LTL và PCTL*, cũng như mở rộng cho thông số kỹ thuật định lượng và các cấu trúc giải thưởng.

PRISM là sự kết hợp giữa cấu trúc dữ liệu mang tính biểu tượng và các thuật toán, dựa trên BDDs (Binary Dicision Diagram - Biểu đồ Quyết định nhị phân) và MTBDDs (Multi-Terminal Binary Dicision Diagram). Nó cũng

bao gồm một cơ cấu mô phỏng các sự kiện rời rạc, cung cấp công cụ hỗ trợ cho việc kiểm tra mô hình thống kê và ước lượng gần đúng, và triển khai các kỹ thuật phân tích khác nhau, chẳng hạn như định lượng tính trừu tượng và tính giảm đối xứng.

PRISM là bộ công cụ miễn phí mã nguồn mở, lưu hành theo giấy phép General Public License (GPL).

3.4.2 Sử dụng PRISM kiểm chứng các tính chất của PTA

Từ phiên bản 4.0 (Tháng 5/2011), công cụ PRISM bắt đầu hỗ trợ kiểm chứng mô hình PTA, bên cạnh các mô hình DTMC, CTMC, MDP đã có trước đó. Tính đến tháng 9/2016, phiên bản PRISM mới nhất là 4.3. Các phương pháp kiểm chứng các tính chất của PTA được cài đặt trong PRISM từ phiên bản 4.0 tới nay là phương pháp đồng hồ số, Phương pháp đạt được lùi và phương pháp trừu tượng hóa với trò chơi ngẫu nhiên.

Mô hình hóa PTA

PRISM sử dụng ngôn ngữ mô hình hóa đồng nhất cho tất cả các mô hình xác suất được hỗ trợ, gồm cả PTA. Để hỗ trợ PTA, PRISM từ phiên bản 4.0 trở đi đã bổ sung loại dữ liệu mới là **clock** cho các biến đồng hồ. Các biến đồng hồ có thể xuất hiện trong các biểu thức điều kiện, trong các vế trái của lệnh, và có thể được reset như các biến thông thường khác. Từ khóa mới **invariant** được bổ sung để biểu diễn càng ràng buộc thời gian tại trạng thái. Hình [] là ví dụ đoạn mã PRISM dùng để biểu diễn PTA tương ứng, trong đó có chứa cấu trúc thưởng với nhãn là energy, được sử dụng để thể hiện tốc độ tích lũy giải thưởng là 2.5 tại trạng thái s=0.

Các kĩ thuật kiểm chứng trong PRISM

PRISM phân tích hai lớp tính chất chính của PTA: (1) Xác suất lớn nhất, nhỏ nhất để đạt tới một mục tiêu nào đó, có thể kèm ràng buộc thời gian (ví dụ: “xác suất lớn nhất của túi khí không bung trong vòng 0.02 giây”); và (2) Giá trị kỳ vọng lớn nhất/nhỏ nhất của giải thưởng tích lũy được khi đạt tới một mục tiêu nào đó (ví dụ: “Kỳ vọng thời gian tối đa để giao thức hoàn tất”). Hiện nay các phương pháp kiểm chứng đồng hồ số, Phương pháp đạt được lùi và phương pháp trò chơi ngẫu nhiên đã được cài đặt, trong đó phương thức trò chơi ngẫu nhiên là cấu hình mặc định của PRISM cho các mô hình PTA.

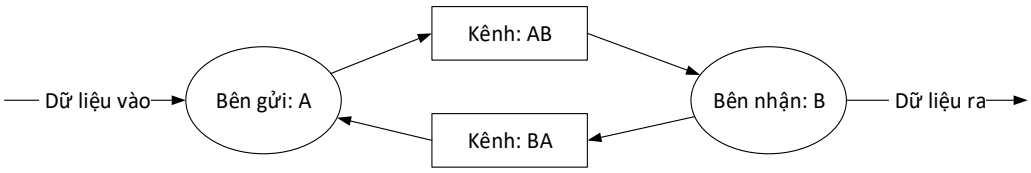
Chương 4. KIỂM CHỨNG MỘT SỐ PTA BẰNG PRISM

4.1 Kiểm chứng giao thức ABP

4.1.1 Giới thiệu giao thức bít luân phiên

Giao thức bít luân phiên (ABP) là giao thức mạng đơn giản hoạt động tại tầng data link để thực hiện truyền lại các bản tin bị mất hoặc bị lỗi. ABP được sử dụng nhiều trong các tình huống test, mô phỏng và kiểm chứng logic hoạt động của các hệ thống đồng thời. Dù mô tả giao thức là khá đơn giản, ABP có nhiều thuộc tính để giải quyết các lý thuyết về hệ thống đồng thời.

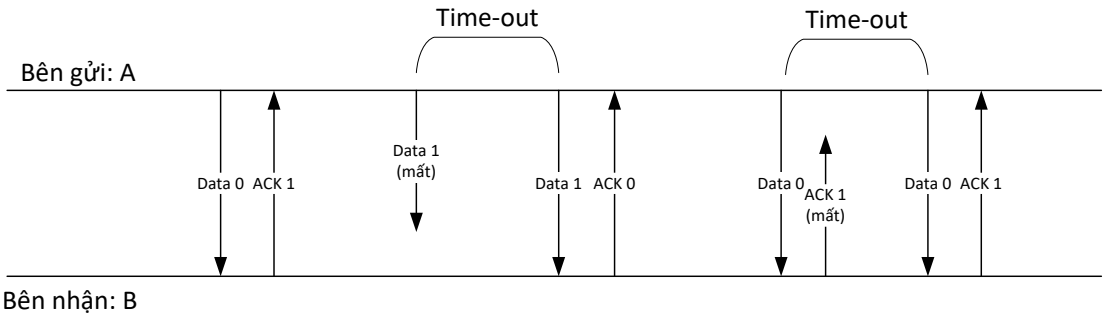
Có thể minh họa giao thức ABP bằng Hình 4.1, có 4 thành phần: Bên gửi A, bên nhận B, Kênh AB truyền dữ liệu từ A đến B và kênh BA truyền dữ liệu từ B đến A. Dữ liệu vào được nhận từ hệ thống ngoài gửi đến A, và khi B nhận được dữ liệu, dữ liệu sẽ được ghi vào dòng Dữ liệu ra, cũng là hệ thống bên ngoài.



Hình 4.1: Các thành phần của một hệ thực thi giao thức bit luân phiên

Bản tin gửi từ A đến B gồm phần dữ liệu và phần Alternating Bit với độ dài 1 bit. B cũng có hai giá trị báo nhận (ACK) gửi lại cho A: ACK0 và ACK1.

Khi có bản tin từ dòng Dữ liệu vào gửi đến A, A thực hiện đọc bản tin và bổ sung thêm Alternating Bit rồi gửi vào kênh AB. Giá trị Alternating Bit chỉ thay đổi giữa 0 và 1, tương ứng với các giá trị ACK1 và ACK0 từ bên nhận. Sau khi gửi bản tin Data 0, A sẽ chờ nhận ACK1 trước khi gửi Data1. Nếu quá thời gian Time-out mà không nhận được ACK 1, A sẽ gửi lại Data 0 và chờ. Quá trình gửi lại bản tin của A cứ tiếp tục cho đến khi A nhận được ACK1 từ kênh BA hoặc time-out xảy ra đối với việc gửi tin. Khi đã nhận ACK phù hợp (vd: ACK1), A thực hiện đảo bit của Alternating Bit và gắn vào bản tin tiếp theo để truyền sang B.



Hình 5.2: Hoạt động của Bên gửi/Bên nhận trong ABP

Khi B nhận bản tin Data 0 từ kênh AB, B kiểm tra giá trị Alternating Bit. Giá trị Alternating Bit khớp với giá trị nội tại của B, bản tin nhận được coi là hợp lệ và gửi ra dòng Dữ liệu ra, đồng thời gửi bản tin báo nhận cho A qua kênh B, và đảo giá trị sequence number để chờ nhận bản tin tiếp theo. Khi nhận được bản tin Data 0, B sẽ gửi ACK1 và A sẽ gửi bản tin Data 1 tiếp theo.

Kênh BA và kênh BA

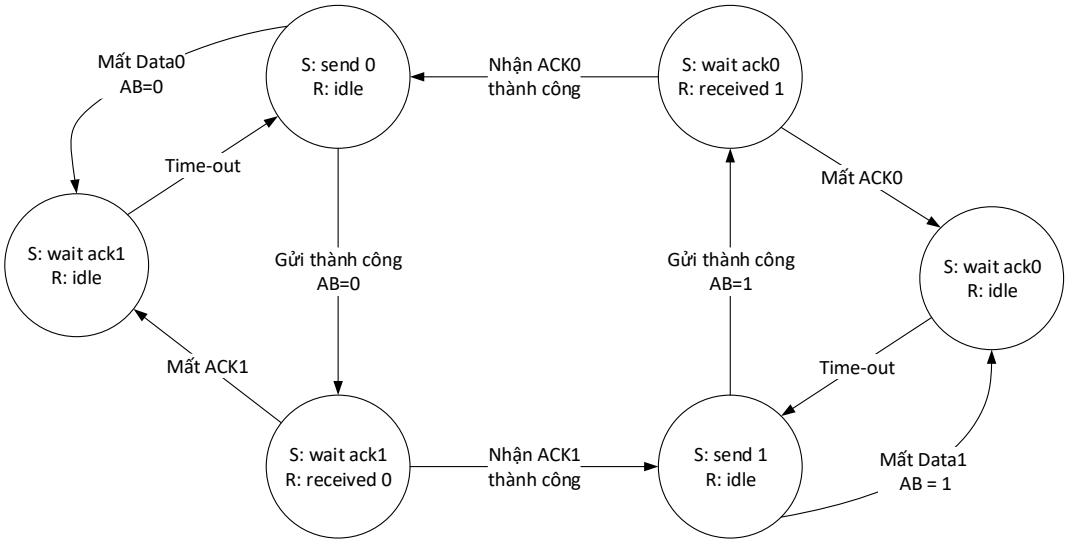
Các kênh được sử dụng để gửi bản tin hoặc ACK từ A đến B và ngược lại. Khi bản tin truyền trên kênh, bản tin có thể bị mất, hoặc bị hỏng trong quá trình truyền.

Giao thức được bắt đầu bằng cách gửi 1 bản tin giả bất kỳ Alternating Bit = 0 (Data 0) và ACK 1. Bản tin đầu tiên có Alternating Bit = 1 là bản tin thực sự được truyền

Giao thức được bắt đầu bằng cách gửi 1 bản tin giả bất kỳ Alternating Bit = 0 (Data 0) và ACK 1. Bản tin đầu tiên có Alternating Bit = 1 là bản tin thực sự được truyền.

Xử lý time-out: mỗi khi chờ ACK, nếu time-out xảy ra A sẽ gửi lại bản tin. A chỉ gửi lại các bản tin được gửi gần nhất nếu nó chưa nhận được ACK tương ứng, và việc gửi lại chỉ thực hiện sau thời gian time-out nhất định. Giá trị time-out phải đủ lớn để tăng hiệu quả xử lý của A và B.

Biểu đồ chuyển trạng thái của hệ chỉ gồm Bên gửi, Bên nhận (chưa gồm các ràng buộc thời gian) được biểu diễn như trong Hình 4.3.



Hình 6.3: Biểu đồ mô tả trạng thái Bên gửi, Bên nhận

4.1.2 Mô hình hóa giao thức ABP bằng PTA

Một số giả thiết đối với hệ thống

Không làm mất tính tổng quát của việc thực hiện mô hình hóa hệ thống, ta có một số giả thiết như sau:

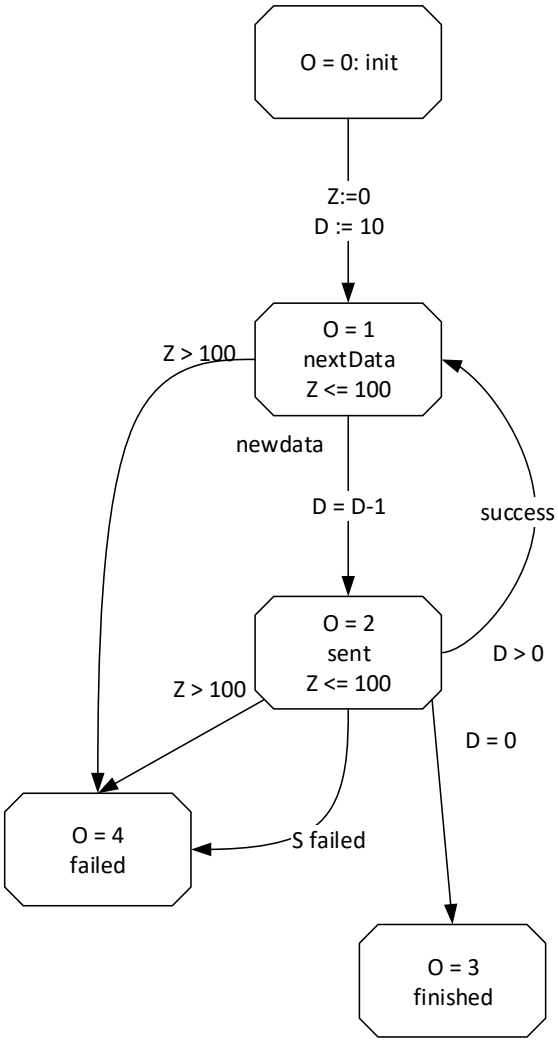
- Tin nhắn được gửi tức thời trên đường truyền (không có độ trễ).
- Bên nhận có cơ chế kiểm tra để biết bản tin được gửi có bị hỏng hay không (chẳng hạn dùng check sum), và với các bản tin hỏng, bên nhận sẽ bỏ qua, không xử lý gì. Do vậy có thể coi bản tin bị hỏng, bị sai lệch như bản tin bị mất trên đường truyền, và khi nói bản tin gửi đến bên nhận sẽ là các bản tin chính xác từ nguồn gửi.
- Phía gửi luôn tin tưởng bản tin đã được chuyển tới đầu kia chính xác (và chuyển sang trạng thái chờ), dù thực tế bản tin có thể bị hỏng, mất trên đường truyền. Tuy nhiên phía chờ có giới hạn thời gian nhận ack tương ứng với một bản tin đã gửi (timeout) để xác định bản tin cần truyền lại.
- Phía nhận tin tưởng bản tin ACK đã được gửi chính xác tới người gửi (và chuyển sang trạng thái nghỉ), dù thực tế có thể bản tin đã mất, hỏng.
- Phía nhận không gửi lại hai bản tin giống nhau liên tục cho dòng Dữ liệu ra nếu không nhận được một bản tin khác ở giữa, ví dụ B sẽ không gửi 2 bản tin Data ứng với ACK1 liên tiếp nếu giữa 2 lần gửi không có 1 bản tin Data ứng với ACK0. Cơ chế này giúp dòng dữ liệu ra không bị nhận các bản tin lặp do cơ chế gửi lại.

Sử dụng các ô tô mát thời gian xác suất để mô hình một hệ thống gửi tin gồm:

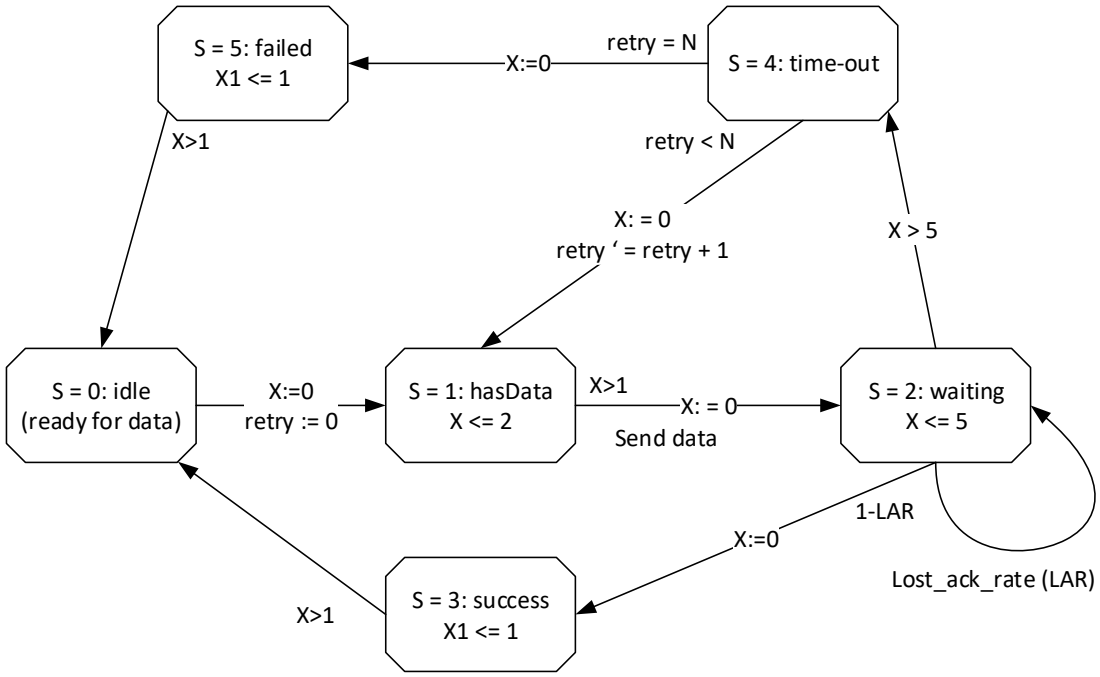
- Nguồn dữ liệu (O): Là nguồn gửi và cần truyền thông điệp sang bên nhận. Thông điệp là khói dữ liệu gồm D bản tin (ví dụ: D = 10), lần lượt các bản tin được chuyển đến Bên gửi để gửi sang Bên nhận. Các bản tin chỉ được gửi nếu bản tin trước đó có kết quả là Thành công. Việc gửi được coi là Kết thúc (thành công) nếu tất cả các bản tin được gửi thành công trong thời gian TIMEOUT cho trước, ngược lại nếu một trong các bản tin không thành công hoặc tổng thời gian gửi tin vượt quá giới hạn định trước (ví dụ: TIMEOUT > 100), hệ thống sẽ ngừng gửi và việc gửi tin là thất bại.
- Bên gửi (S): Bên gửi nhận lần lượt từng gói dữ liệu vào và gửi sang Bên nhận theo giao thức ABP. Mỗi lần gửi tin, có một số bản tin không đến được Bên nhận, hoặc đến nhưng giá trị alternating bit không khớp, gọi là bản tin không thành công. Tỉ lệ các bản tin không thành công khi gửi được thể hiện bằng tham số LOST_DATA_RATE. Sau khi nhận gói tin từ Nguồn dữ liệu, thời gian S cần xử lý và gửi tin trong thời gian 1-2 (1 <= x <= 2). Sau khi gửi tin, S sẽ chờ nhận ACK từ R. Nếu sau khoảng thời gian ACK_TIMEOUT định trước (ví dụ: ACK_TIMEOUT = 5) không nhận được ACK thì S sẽ:
    - (i) hoặc gửi lại bản tin nếu số lần gửi lại còn nhỏ hơn số RETRY cho trước;
    - (ii) hoặc coi việc gửi bản tin là thất bại và không gửi lại bản tin đó nữa.
    - Khi xác định trạng thái gửi tin là Thành công hay Thất bại, S có khoảng thời gian nghỉ 1-2 đơn vị trước khi chuyển về trạng thái sẵn sàng để gửi bản tin tiếp theo.
- Bên nhận (R): Mỗi khi nhận được bản tin từ Bên gửi, Bên nhận mất 1 đến 2 đơn vị thời gian để xử lý bản tin (1 <= y <=2), sau đó chuyển sang trạng thái gửi ACK về bên nhận. Việc gửi ACK cũng mất 1-2 đơn vị thời gian. Sau khi đã gửi ACK, R chuyển về trạng thái nghỉ để sẵn sàng nhận bản tin tiếp theo. Khi R gửi ACK, có một tỷ lệ nhất định bản tin sẽ mất trên đường truyền hoặc bị sai lệch khi đến đích. Tỉ lệ này được thể hiện bởi tham số LOST_ACK_RATE.

Từ các đặc tả trên, ta có các thành phần trao đổi bản tin của hệ thống là các ô tô mát thời gian xác suất, và cả hệ thống gồm các PTA được biểu diễn bằng các biểu đồ chuyển trạng thái như biểu diễn trong Hình 4.4, Hình 4.5

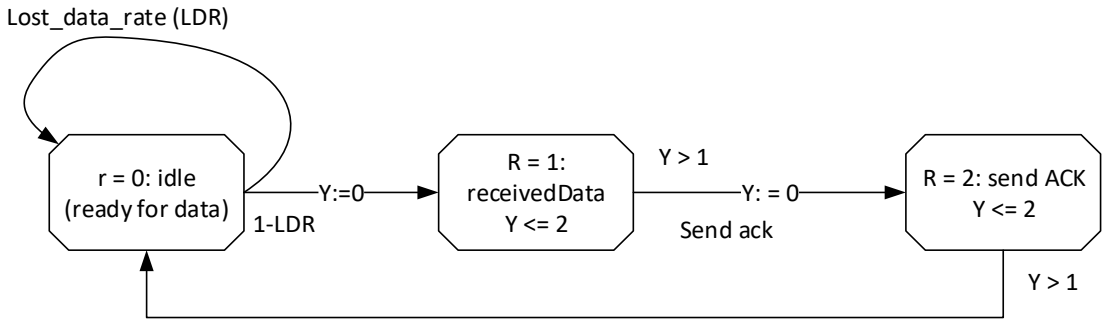
và Hình 4.6 (trong biểu diễn các hình này, giả thiết: TIMEOUT = 100; ACK_TIMEOUT = 5; DATA = 10):



Hình 7.4: Biểu đồ trạng thái của Nguồn gửi trong quá trình truyền tin



Hình 8.5: Biểu đồ trạng thái của Bên gửi trong quá trình truyền tin



Hình 9.6: Biểu đồ trạng thái của Bên nhận trong quá trình truyền tin

Các đặc điểm xác định trước của hệ thống:

- Xác suất mất gói tin của đường truyền: LOST_DATA_RATE;
- Xác suất mất bản tin ack của đường truyền: LOST_ACK_RATE;
- Số lần retry của một gói: RETRY;
- Thời gian time-out của cả hệ thống: TIMEOUT;
- Thời gian time-out chờ nhận ACK_TIMEOUT;
- Số lượng bản tin cần truyền DATA;

4.2 Cài đặt hệ truyền tin ABP bằng công cụ PRISM

Mã cài đặt PTA của hệ truyền tin bằng giao thức bít luân phiên trong công cụ PRISM như sau:

Bảng 1.1 : Cài đặt hệ thực thi ABP trong PRISM

```
// PTA model checking for Alternating Bit Protocol
// Coding for the Thesis: Automatic Verification for probability timed automata
// Author: Nguyen Duc Tho
// Year: 2016
// Intructor: Dang Van Hung
// Assumption: the signal send instantaneous over channel.
// This code is written for BACKWARD REACHABILITY, and need some modification for other checking engines
// Digital clock: can not accept strict comparation for clock variable, i.e x > 1 is unacceptable.

pta

const double LOST_DATA_RATE;
const double LOST_ACK_RATE;
const int RETRY; //so lan retry can thuc hien
const int DATA; // so ban tin can gui tu nguon du lieu den dich
const int TIMEOUT; // thoi gian gui khong qua 100
const int ACK_TIMEOUT = 5; //thoi gian cho toi da cua Sender de nhan ack
module sender
	s:[0..5] init 0;
	// 0: idle (ready for data)
	// 1: has data
	// 2: data sent, waiting
	// 3: get ack, message sent success
	// 4: time-out, urgency location
	// 5: message lost
	x: clock;
	retries:[0..RETRY] init 0;
	invariant
		(s = 1 => x <= 2) &
		(s = 2 => x <= ACK_TIMEOUT + 1) &
		(s = 3 => x <= 2) &
		(s = 4 => x = 0) &
		(s = 5 => x <= 2)
	endinvariant
	[newdata] (s = 0) -> (s' = 1) & (x' = 0) & (retries' = 0);
	[transmit_data] (s = 1) & (x > 1) -> (s' = 2);
	[transmit_ack] (s = 2) & (x > 1) -> (1 - LOST_ACK_RATE): (s' = 3) & (x' = 0) + LOST_ACK_RATE: (s' = s);
	[success] (s = 3) & (x > 1) -> (s' = 0) & (x' = 0);
	[timeout] (s = 2) & (x > ACK_TIMEOUT)  -> (s' = 4) & (x' = 0);
```

```
	[retry] (s = 4) & (retries < RETRY) -> (s' = 1) & (x' = 0) &
(retries' = retries + 1);
	[msg_lost] (s = 4) & (retries = RETRY) -> (s' = 5);
	[lost] (s = 5) & (x > 1) -> (s' = 0);
endmodule

module receiver
	r: [0..2] init 0;
	// 0: idle, ready for data
	// 1: received data
	// 2: send ack
	y: clock;
	invariant
		(r = 1 => y <= 2) &
		(r = 2 => y <= 2)
	endinvariant
	[transmit_data] (r = 0) -> (1 - LOST_DATA_RATE): (r' = 1)
& (y' = 0) + LOST_DATA_RATE: (r' = r);
	[] (r = 1) & (y > 1) -> (r' = 2) & (y' = 0);
	[transmit_ack] (r = 2) & (y > 1) -> (r' = 0);
endmodule

module datasource
	ds:[0..5] init 0;
	// 0: init
	// 1: nextData
	// 2: waiting for sender to send message
	// 3: success, urgent location
	// 4: failed
	// 5: finished
	z: clock;
	z1: clock;
	data:[0..DATA];
	invariant
		(ds = 1 => z <= TIMEOUT+1) &
		(ds = 2 => z <= TIMEOUT+1) &
		(ds = 3 => z1 = 0) &
		(ds = 4 => z1 <= 2) &
		(ds = 5 => z1 <= 2) &
		(ds = 0 => z1 <= 2)
	endinvariant
	[](ds = 0) & (z1 > 1) -> (ds' = 1) & (z' = 0) & (data' = DATA);
	//gui ban tin xuong sender
	[newdata] (ds = 1) -> (ds' = 2) & (data' = data - 1);
	//trang thai gui tin thanh cong
	[success] (ds = 2) -> (ds' = 3) & (z1' = 0);
	//gui thanh cong het cac ban tin
```

```
        [] (ds = 3) & (data = 0) -> (ds' = 5) & (z1' = 0);
        //van con ban tin, chuyen ban tin tiep theo
        [] (ds = 3) & (data > 0) -> (ds' = 1);
        //mat 1 ban tin nao day
        [lost] (ds = 2) -> (ds' = 4) & (z1' = 0);
        //time out
        [](ds = 1 | ds = 2) & (z > TIMEOUT) -> (ds' = 4) & (z1' = 0);
        // change to init, for next data
        //[](ds = 4 | ds = 5)  & (z1 > 0)-> (ds' = 0);
endmodule
label "finished" = ds = 5;
label "failed" = ds = 4;
label "lost" = s = 5;
```

Các tính chất của hệ sẽ được kiểm chứng với các đặc điểm xác định trước của hệ thống:

- Xác suất lớn nhất để hoàn thành việc truyền dữ liệu thành công là bao nhiêu?
  Pmax = ? [F “finished”]
- Xác suất lớn nhất khi truyền dữ liệu thành công trong T thời gian là bao nhiêu?
  Pmin = ? [F <= T “finished”]
- Xác suất lớn nhất bị mất gói tin là bao nhiêu?
  Pmax = ? [F “lost”]
- Xác suất lớn nhất bị mất gói tin trong thời gian T là bao nhiêu?
  Pmax = ? [F <=T “lost”]

4.2.1 Kết quả kiểm chứng và các đánh giá

4.2.1.1 Pmax = ? [F “finished”]

Kết quả kiểm chứng của PRISM với các tham số:

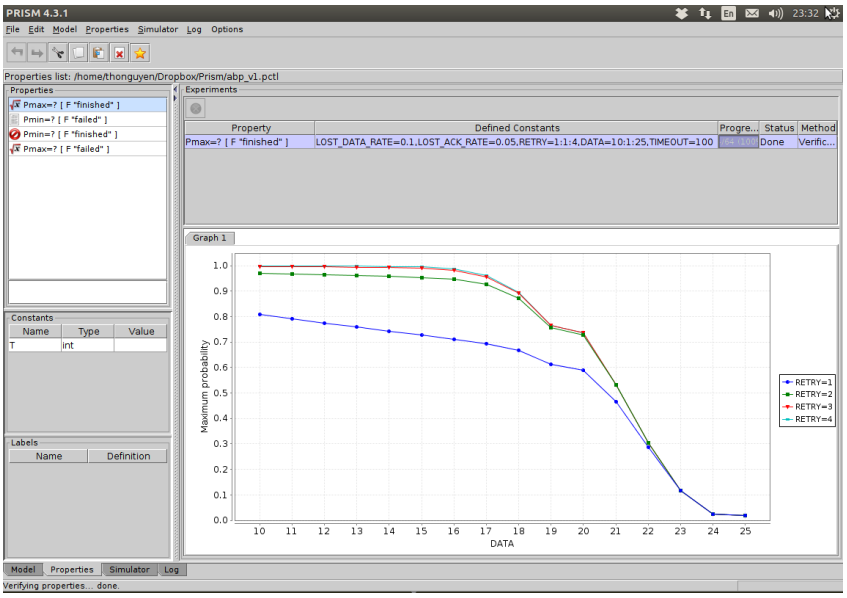
+ LOST_DATA_RATE = 0.1;

+ LOST_ACK_RATE = 0.05;

+ RETRY = 1..4

+ DATA = 10..24

Biểu đồ kết quả kiểm chứng được ghi nhận tại hình 10.



Hình 10.7: Pmax = ? [F "finished"]

Nhận xét:

- Xác suất gửi bản tin thành công tăng thêm khi số lần RETRY tăng thêm, tuy nhiên các giá trị RETRY >=3 không có khác biệt đáng kể, do giới hạn TIMEOUT = 100 của hệ thống.

4.2.1.2 Pmax = ? [F "lost"]

Kết quả kiểm chứng của PRISM với các tham số:

+ LOST_DATA_RATE = 0.1;

+ LOST_ACK_RATE = 0.1;

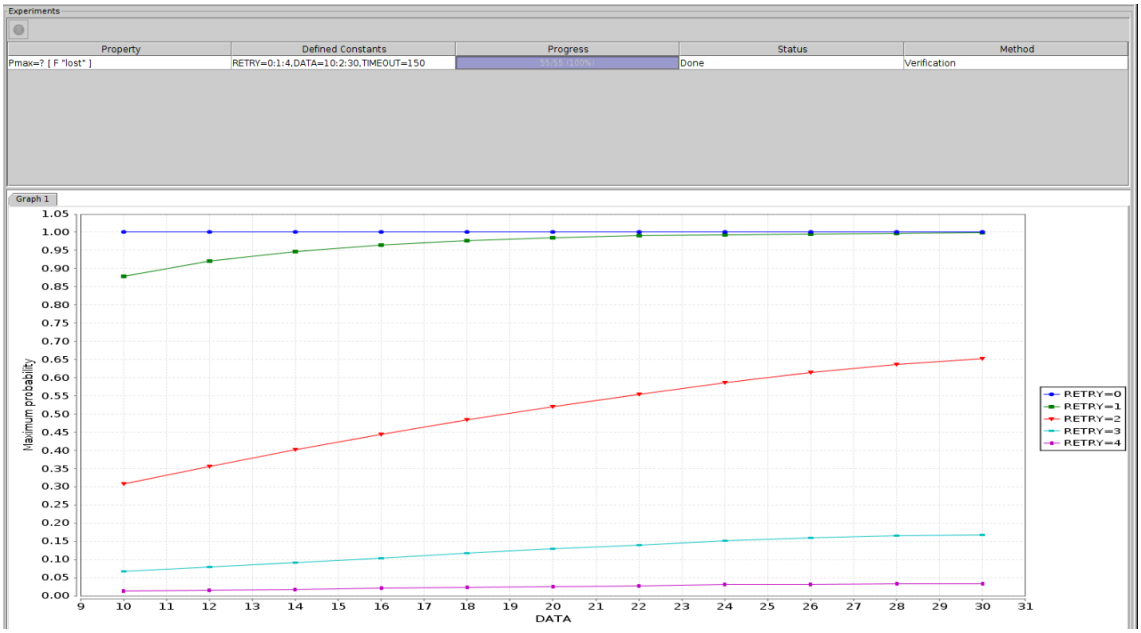
+ TIMEOUT = 150;

+ ACK_TIMEOUT = 5;

+ RETRY = [0..4]

+ DATA = [10..30]

Biểu đồ kết quả kiểm chứng được ghi nhận tại Hình 11.



Hình 11.8: Pmax = ? [F "lost"]

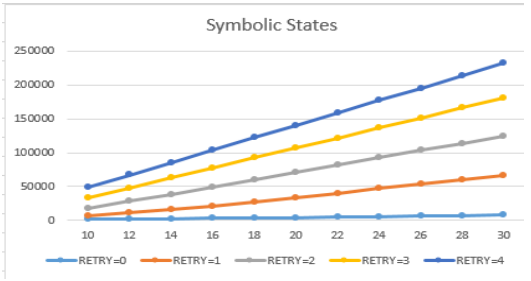
- Việc tăng số lượng retry giúp giảm nhanh khả năng mất gói tin trong quá trình truyền tin.

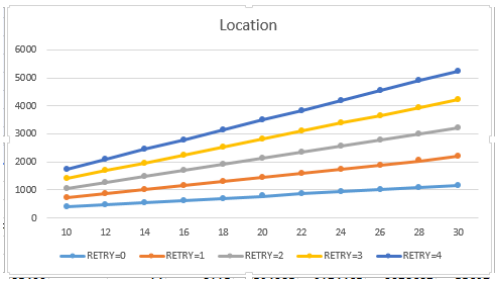
Sử dụng phương thức kiểm chứng đạt được lùi, khi xây dựng MDP, dữ liệu Bảng 3 ghi nhận kích thước của MDP (số lượng trạng thái, số lượng dịch chuyển và số lượng lựa chọn của MDP) đều tăng tuyến tính theo tất cả các giá trị hằng số.

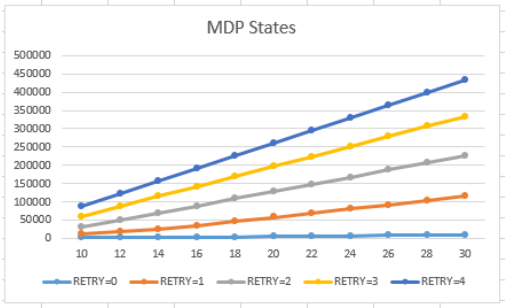
Bảng 2.2 : Quy mô tính toán khi DATA = 10..30; RETRY = 0..4

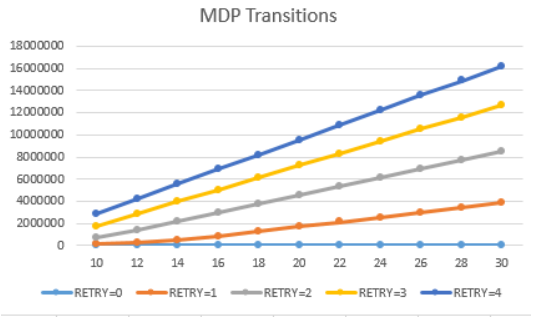
DATA	RETRY	LOCATION	SYMBOLIC STATES	MDP STATES	MDP TRANSITIONS	MDP CHOICES
10	0	393	1499	1929	4155	5204
12	0	471	1943	2507	5907	7238
14	0	549	2435	3149	8115	9752
16	0	627	2975	3855	10843	12810
18	0	705	3563	4625	14155	16476
20	0	783	4199	5459	18115	20814
22	0	861	4883	6357	22787	25888
24	0	939	5615	7319	28235	31762
26	0	1017	6395	8345	34523	38500
28	0	1095	7223	9435	41715	46166
30	0	1173	8099	10589	49875	54824
10	1	731	7313	12093	151531	155739
12	1	877	11106	18644	300176	306444
14	1	1023	15727	26671	524385	533145
16	1	1169	21176	36174	839710	851394

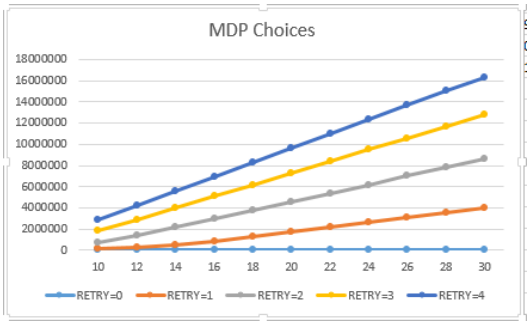
18	1	1315	27387	47042	1250853	1265854
20	1	1461	33874	58409	1696183	1714641
22	1	1607	40366	69783	2141521	2163439
24	1	1753	46858	81157	2586859	2612237
26	1	1899	53350	92531	3032197	3061035
28	1	2045	59842	103905	3477535	3509833
30	1	2191	66334	115279	3922873	3958631
10	2	1069	18160	32082	709856	719914
12	2	1283	28012	49965	1382508	1397847
14	2	1497	38754	69535	2174465	2195536
16	2	1711	49528	89175	2970649	2997468
18	2	1925	60302	108815	3766833	3799400
20	2	2139	71076	128455	4563017	4601332
22	2	2353	81850	148095	5359201	5403264
24	2	2567	92624	167735	6155385	6205196
26	2	2781	103398	187375	6951569	7007128
28	2	2995	114172	207015	7747753	7809060
30	2	3209	124946	226655	8543937	8610992
10	3	1407	33408	60587	1787875	1806138
12	3	1689	48152	87875	2879275	2905414
14	3	1971	62896	115169	3970687	4004702
16	3	2253	77640	142463	5062099	5103990
18	3	2535	92384	169757	6153511	6203278
20	3	2817	107128	197051	7244923	7302566
22	3	3099	121872	224345	8336335	8401854
24	3	3381	136616	251639	9427747	9501142
26	3	3663	151360	278933	10519159	10600430
28	3	3945	166104	306227	11610571	11699718
30	3	4227	180848	333521	12701983	12799006
10	4	1745	48799	89680	2894619	2921125
12	4	2095	67149	123922	4228697	4265011
14	4	2445	85499	158164	5562775	5608897
16	4	2795	103849	192406	6896853	6952783
18	4	3145	122199	226648	8230931	8296669
20	4	3495	140549	260890	9565009	9640555
22	4	3845	158899	295132	10899087	10984441
24	4	4195	177249	329374	12233165	12328327
26	4	4545	195599	363616	13567243	13672213
28	4	4895	213949	397858	14901321	15016099
30	4	5245	232299	432100	16235399	16359985



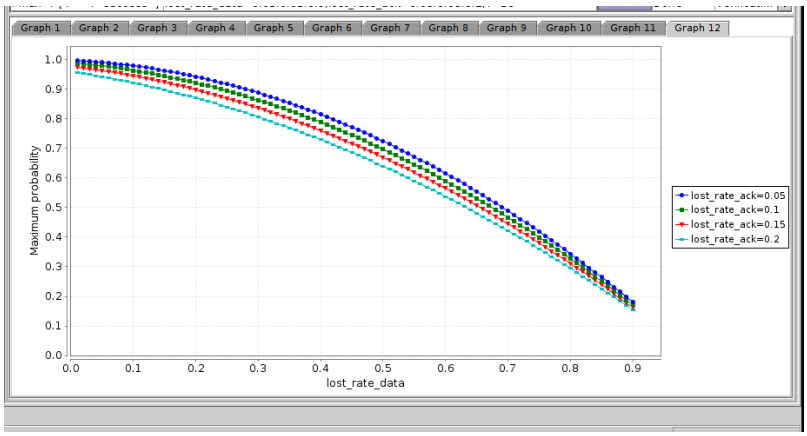




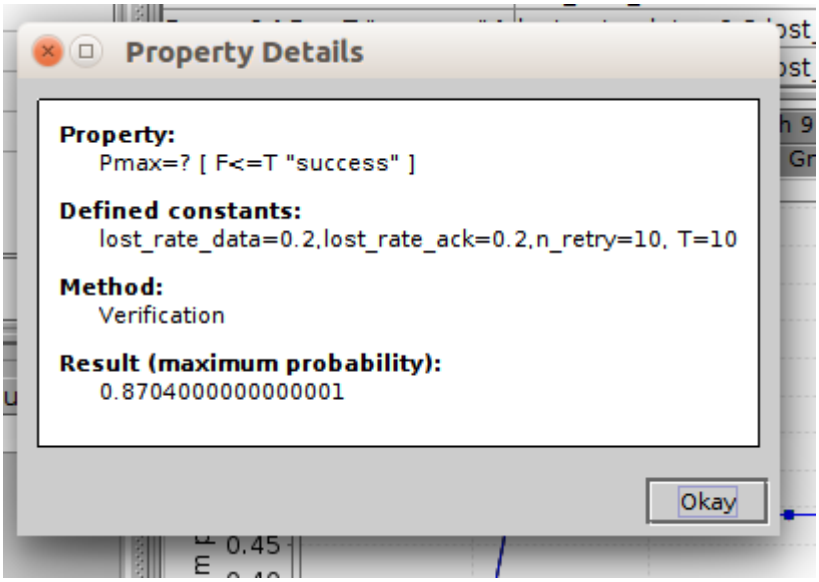




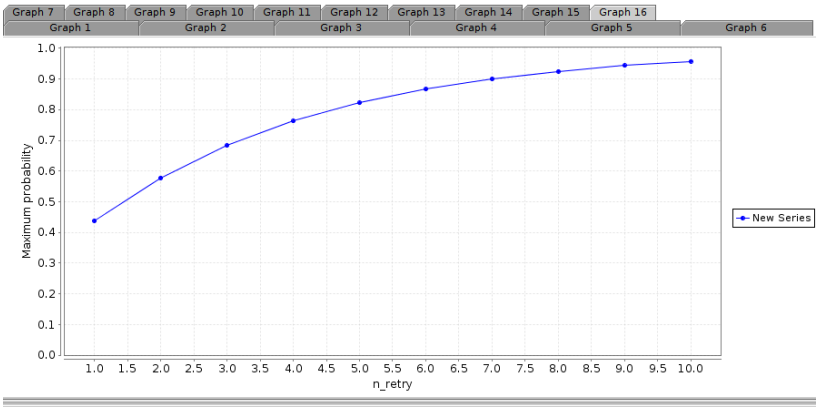
Mô hình hóa hệ đơn giản, chỉ bao gồm phần gửi và phần nhận.



Hình 12.9: Pmax =? [F <=T “success”] theo lost_rate_data (T=10)



Hình 13.10: Pmax = ? [F <= T "success"]



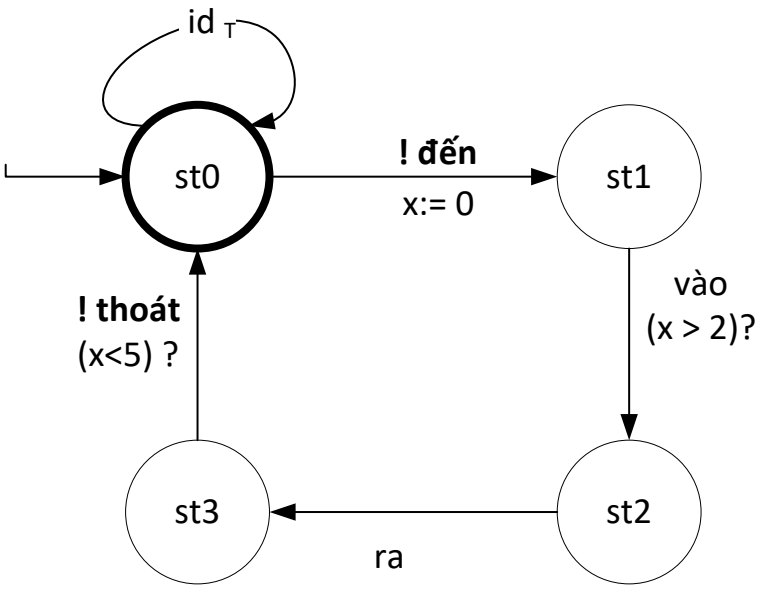
Hình 14.11: Pmax = ?[F “success”] thay đổi theo số lần retry

4.3 Hệ điều khiển tự động đường ngang

Phần này sẽ sử dụng công cụ PRISM để kiểm chứng tự động các tính chất xác suất của hệ điều khiển tự động đường ngang, là ví dụ được đề cập trong [7] và bổ sung thêm các yếu tố xác suất tại các thành phần điều khiển. Nhắc lại về ví dụ được nêu trong [7], theo đó hệ gồm 3 thành phần: TRAIN, GATE và CONTROLLER.

4.3.1 Mô hình hóa bằng PTA

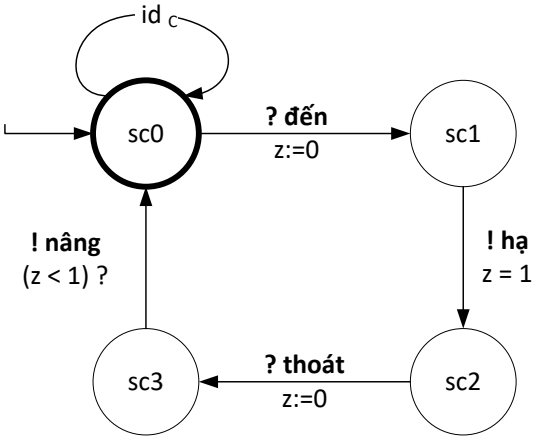
Hệ thống được biểu diễn bằng các module ô tô mát tương ứng với TRAIN, GATE, CONTROLLER.



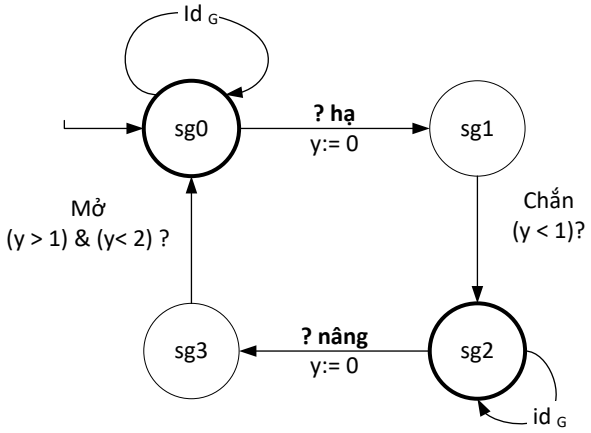
Hình 15.12: TRAIN

Ô tô mát thể hiện mô hình đoàn tàu TRAIN như Hình 4.12. Tập các sự kiện của ô tô mát là {đến, vào, ra, thoát, nghỉ}. Đoàn tầu bắt đầu từ trạng thái st0. Sự kiện nghỉ (id T) thể hiện không có tàu, hoặc đoàn tàu không cần vào gác chắn. Khi có đoàn tàu đến gần gác chắn, cảm biến chuyển động sẽ báo tín hiệu "đến" cho bộ điều khiển, và khi đoàn tàu rời xa gác chắn, tín hiệu "thoát" cũng được gửi tới bộ điều khiển. Các sự kiện "vào" và "ra" thể hiện việc đoàn tàu tiến vào và đi ra khỏi gác chắn. Đoàn tàu phải gửi tín hiệu "đến" ít nhất 2 phút trước khi vào gác chắn, do vậy độ trễ tối thiểu giữa trạng thái "vào" và "đến" là 2 đơn vị thời gian. Ngoài ra, đoàn tàu mất không quá 5 phút để vượt qua gác chắn, do vậy thời gian từ khi gửi tín hiệu "đến" và "thoát" không quá 5 phút.

Ô tô mát mô hình bộ điều khiển CONTROLLER như Hình 4.13. Tập các sự kiện hành động bao gồm {đến, hạ, thoát, nâng, nghỉ}, CONTROLLER bắt đầu từ trạng thái sc0. Khi nhận tín hiệu "đến" từ sensor, CONTROLLER thực hiện gửi lệnh "hạ" đến GATE, thời gian thực hiện lệnh không quá 1 phút. Khi nhận tín hiệu "thoát", nó tiếp tục gửi lệnh "nâng" đến GATE.



Hình 16.13: CONTROLLER

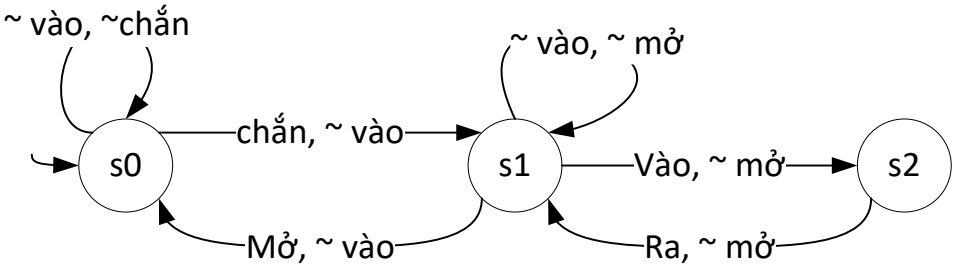


Hình 17.14: GATE

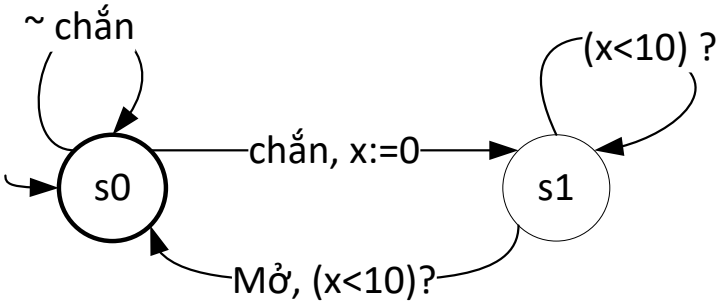
Ô tô mát mô hình gác chắn GATE như Hình 4.14. Tập các sự kiện hành động bao gồm {hạ, nâng, đóng, mở, nghỉ}, GATE bắt đầu từ trạng thái sg0. Khi nhận lệnh "hạ" từ CONTROLLER, GATE thực hiện chuyển đóng chắn trong vòng 1 phút, và chuyển sang trạng thái đóng. Khi nhận lệnh "nâng", GATE mất 1 đến 2 phút để mở chắn và chuyển sang trạng thái mở. Khi không có lệnh từ CONTROLLER, GATE ở nguyên trạng thái "đóng" hoặc "mở" đang có.

- Tính chất an toàn của đường ngang: Đường ngang chỉ an toàn khi GATE đã đóng trước khi tàu đi qua, và chỉ mở khi tàu đã đi qua hẳn.
- Tính chất đảm bảo lưu thông: Đường ngang đạt mục tiêu đảm bảo lưu thông khi thời gian đóng chắn không quá 10 phút.

Hình 4.15 và Hình 4.16 là các ô tô mát biểu diễn các tính chất an toàn và tính chất đảm bảo lưu thông của đường ngang.



Hình 18.15: Trạng thái an toàn đường ngang



Hình 19.16: Đảm bảo tính lưu thông

Bổ sung yếu tố xác suất trong đường ngang: Trong thực tế hoạt động với một hệ như đường ngang, có xác suất xảy ra lỗi đối với các thành phần trong đường ngang. Việc mô hình hóa hệ điều khiển đường ngang trong đề tài sẽ quan tâm tới các yếu tố xác suất sau:

- Xác suất lỗi không gửi tín hiệu "đến" của sensor cho CONTROLLER, biểu diễn bằng SFR (sensor failed rate) trong cài đặt.
- Xác suất lỗi thực hiện lệnh của CONTROLLER, biểu diễn bằng tham số CFR (controller failed rate) trong cài đặt.
- Xác suất hỏng của GATE khi nhận lệnh từ CONTROLLER, biểu diễn bằng tham số GFR (gate failed rate) trong cài đặt.

Hệ điều khiển đường ngang có đầy đủ các tính chất xác suất và thời gian, do vậy là ví dụ đầy đủ về PTA và có thể dùng PRISM để mô hình hóa và kiểm chứng các tính chất xác suất của hệ.

Các tính chất có thể kiểm chứng:

- Xác suất gây ra tắc nghẽn giao thông, khi GATE ở trạng thái đóng quá thời gian cho phép, Pmax = ? [F "jam"]
- Xác suất đảm bảo an toàn cho đường ngang, tức khi tàu vào đường ngang thì GATE đang ở trạng thái đóng, Pmax = ? [F "safe"]

4.3.2 Cài đặt trong PRISM

Mã cài đặt trong PRISM của hệ điều khiển đường ngang như sau:

Bảng 3.3 : Cài đặt hệ điều khiển đường ngang trong PRISM

```
// Mo phong lai vi du gac chan cua trong bai Timed Automata
// Tac gia: Nguyen Duc Tho
// Giao vien huong dan: Dang Van Hung
// Version: 1
// Verification engine: stochastic game and backward reachability (tested,
ok)

pta

const double SFR; //sensor failed rate, for sending sigal to controller
const double CFR; // controller failed rate, for sending command to gate
const double GFR; //gate failed rate, unable to execute command to
change gate state

//const int N = 5;

module train
	st:[0..3] init 0;
	//0 : ko co tau
	//1 : tau vua den
	//2 : tau vua vao gac
	//3 : tau vua roi khoi gac

	x: clock;
	invariant
		(st = 2 => x <= 5)
	endinvariant
	[approach] (st = 0) -> (st' = 1) & (x' = 0); // tau den
	[in] (st = 1) & (x > 2) -> (st' = 2); // tau vao gac chan
	[out] (st = 2) & (x > 4) -> (st' = 3); //roi khoi gac chan
	[exit] (st = 3) -> true; // thoat
endmodule

module controller
	sc:[0..3] init 0;
	// 0: ko co tau, idle
	// 1: nhan tin hieu tau vua den
	// 2: Vua gui lenh ha gac
	// 3: Nhan tin hieu tau vua thoat
	z: clock;
	//invariant
```

```
        //      (sc = 3 => z < 1)
        //endinvariant
        [approach] (sc = 0) -> (1-SFR): (sc' = 1) & (z' = 0) + SFR: (z'= 0);
        [lower] (sc = 1) & (z < 1) -> (sc' = 2);
        [exit] (sc = 2) -> (1-SFR): (sc' = 3) & (z' = 0) + SFR: (sc' = 2);
        [raise] (sc = 3) & (z < 1) -> (sc' = 0);
//      [raise] (sc = 3) & (z = 1) -> (sc' = 0);

endmodule

module gate
        sg:[0..3] init 0;
        //0: khong co tau, gac mo
        //1: vua nhan lenh Ha tu controller
        //2: Da thuc hien chan gac, gac dong
        //3: Da nhan lenh nang gac tu controller
        //4: Gac tau mo thanh cong, ket thuc qua trinh tau di qua

        jam: bool init false;
        y: clock;
        [lower] (sg = 0) -> (1-CFR): (sg' = 1) & (y' = 0) + CFR: (y' = 0);//
nhan lenh ha chan
        [down] (sg = 1) & (y < 1) -> (1 - GFR): (sg' = 2) + GFR: (sg' = 1);
//chan duoc ha xuong trong thoi gian <1
        [raise] (sg = 2) -> (1-CFR):(sg' = 3) & (y' = 0) & (jam' = false) +
CFR: (sg' = 2) & (jam' = true); //nhan lenh nang chan, xac suat loi
Controller ko gui dc lenh
        [up] (sg = 3) & (y < 2) & (y > 1) -> (1-GFR): (sg' = 0) + GFR: (sg'
= 3) & (jam' = true); //chan dc nang trong thoi gian 1<y<2
endmodule

label "danger" = ((st = 2) & (sg = 1) = true);
label "safe" = ((st = 2) & (sg = 2) = true);
label "jam" = jam = true;
label "success" = (jam = false) & ((st = 2) & (sg = 2) = true);
```

4.3.3 Kết quả kiểm chứng

SFR, CFR = 0.01, 0.03, 0.07, 0.10

“Success”:

- Tàu đi qua
- Không gây tắc đường

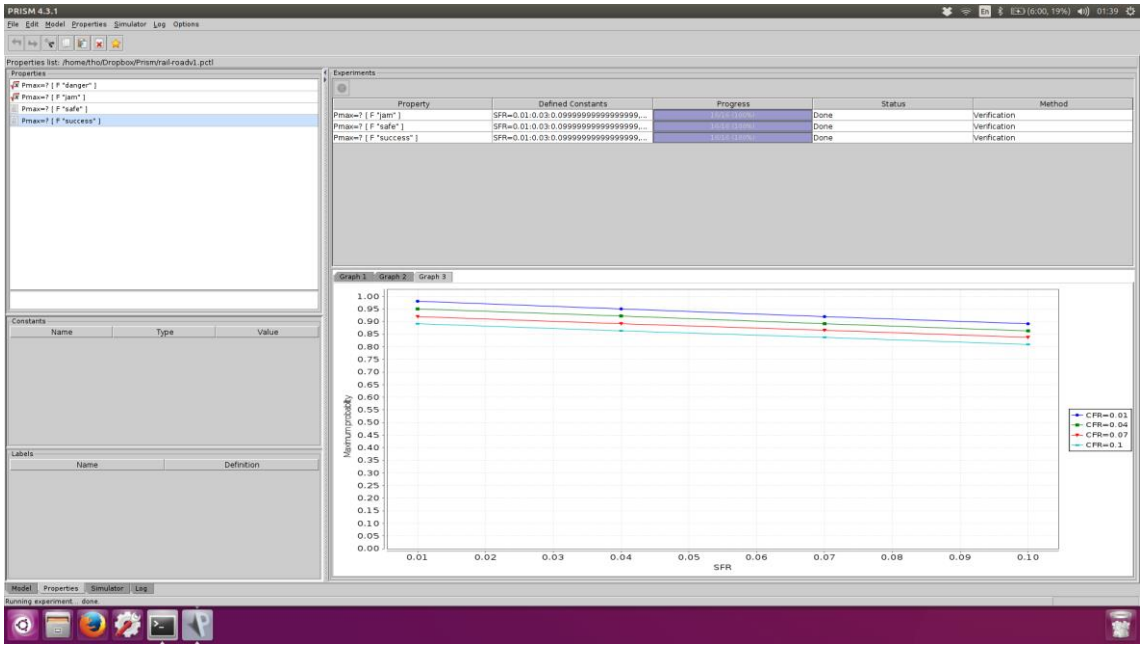
“Safe”:

- Tàu đi qua gác thì gác phải đang đóng

"jam":

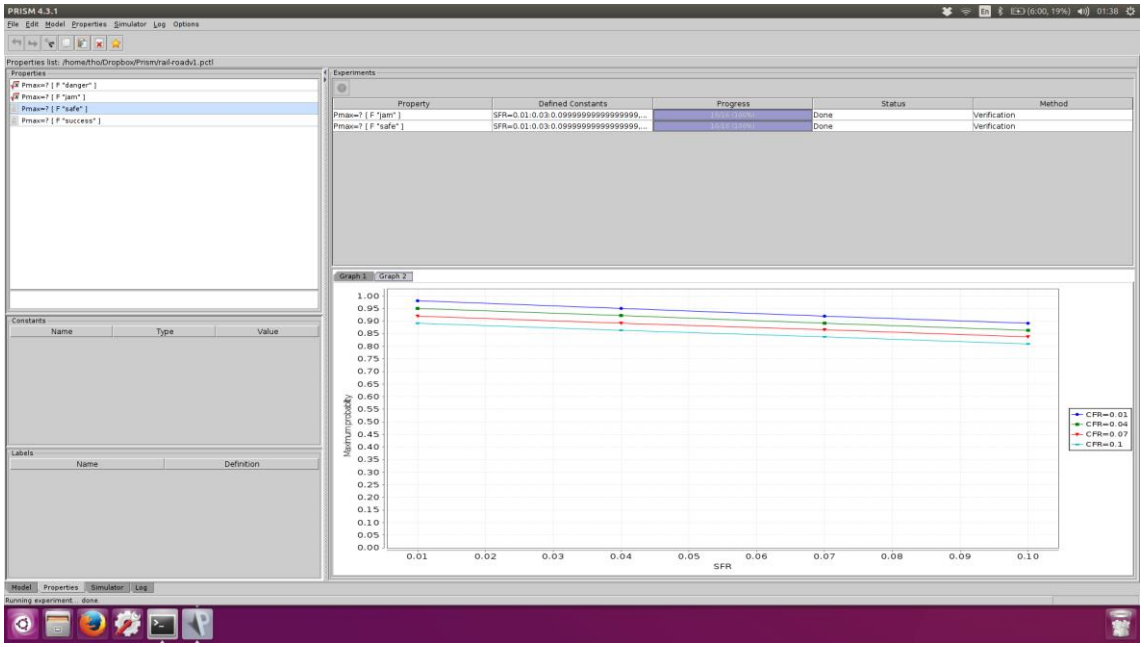
- Gác chắn bị lỗi, không nâng lên được (lỗi gate hoặc ko nhận tín hiệu nâng từ controller)

4.3.3.1 Kiểm chứng Pmax = ?[F "success"]



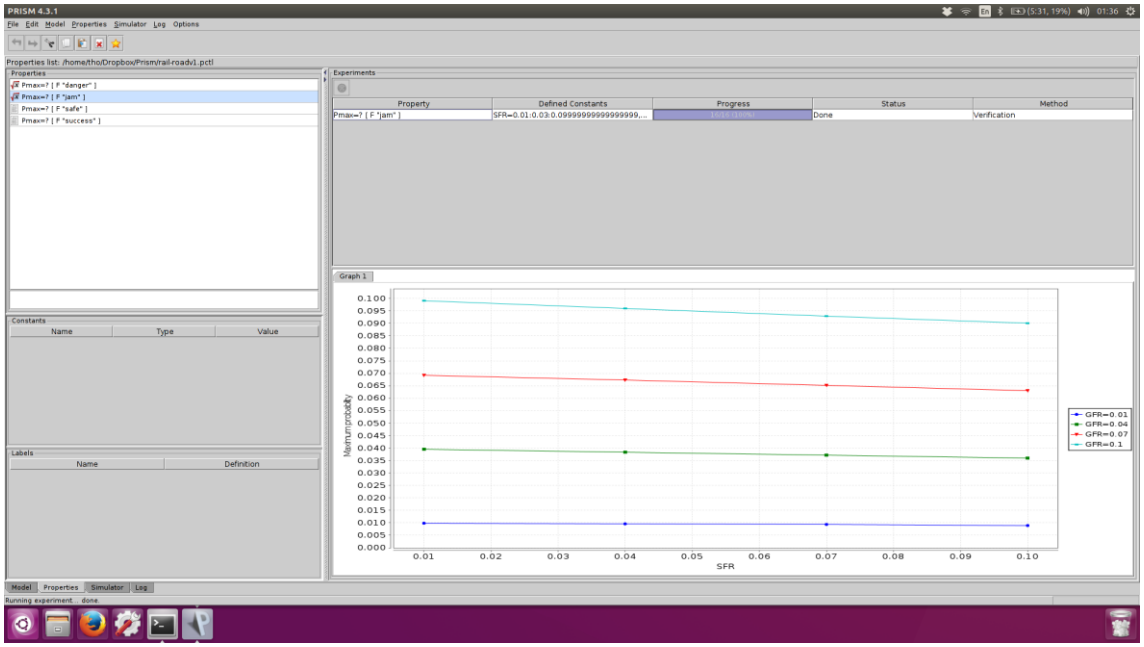
Hình 20.17: Kiểm chứng Pmax = ? [F "success"]

4.3.3.2 Kiểm chứng Pmax = ?[F "safe"]



Hình 21.18: Kiểm chứng Pmax = ? [F "safe"]

4.3.3.3 Kiểm chứng Pmax = ?[F “jam”]



Hình 22.19: Kiểm chứng Pmax = ? [F "jam"]

KẾT LUẬN

Việc kiểm chứng tự động các hệ thống với yếu tố xác suất có nhiều áp dụng trong các tình huống thực tế, và là một lĩnh vực nghiên cứu mới. Công cụ PRISM kiểm chứng các PTA mới có khả năng hỗ trợ kiểm chứng hạn chế các đặc tính xác suất của hệ thống, bao gồm các tính chất xác suất biểu diễn dạng PTCTL không chứa biểu thức R. Một phần giới hạn do số lượng trạng thái các MDP được xây dựng bùng nổ tuyến tính đồng thời theo tất cả các giá trị hằng số được ràng buộc trong hệ mô hình hóa. Các nghiên cứu lý thuyết về khả năng kiểm chứng các tính chất xác suất của PTA vẫn đang được tiến hành nhằm mục đích cài đặt các công cụ kiểm chứng tự động (như PRISM).

Trong phạm vi đề tài, giao thức bít luân phiên đã có thể được mô hình hóa và kiểm chứng định lượng các tính chất xác suất lớn nhất. Đề tài có thể tiếp tục phát triển và thực hiện so sánh hiệu năng, kích cỡ các MDP khi kiểm chứng giao thức bít luân phiên bằng các phương thức kiểm chứng khác nhau.

TÀI LIỆU THAM KHẢO

1. Charles M. Grinstead (Swarthmore College, USA), J. Laurie Snell (Dartmouth College, USA) (2003), *Introduction to Probability* (2nd edition), American Mathematical Society, Chapter 11.Markov Chains, pp.405-470. http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/book.html , freely redistributed under the terms of the GNU Free Documentation License (FDL).

2. D.V.Hung, M.Zhang, On verification of probabilistic timed automata against probabilistic duration properties, in: 13th IEEE International Conference on Embedded and Real-time Computing Systems and Applications (RTCSA 2007), 21-24 August 2007, Daegu, Korea, 2007, p.165-172
3. Gethin Norman, David Parker and Jeremy Sproston (2013), Springer, Model Checking for Probabilistic Timed Automata. *Formal Methods in System Design*, 43(2), pp.164-190
4. Sheldon M.Ross (University of Southern California, USA) (2010), *Introduction to Probability Models (Tenth Edition)*, Elsevier, Chapter 4.Markov Chains, p.191-291; Chapter 6. Continuous-Time Markov Chains, pp.371-420.
5. Marta Kwiatkowska, Gethin Norman (University of Birmingham, UK), Roberto Segala (Università di Verona, Italy), Jeremy Sproston (University of Birmingham) (2002), *Theoretical Computer Science*, Elsevier, Automatic verification of real-time systems with discrete probability distributions, pp.101-150
6. Kwiatkowska M., Norman G., Sproston J.,Wang F.: Symbolic model checking for probabilistic timed, automata. Information and Computation 205(7), 1027–1077 (2007)
7. Rajeev Alur and David L.Dill (Computer Science Department, Standford University, USA) (1994), *Theoretical Computer Science, A theory of timed automata*, pp.183-235.
8. Van Hung Dang, Miaomiao Zhang, Dinh Chinh Pham (2015), VNU Journal of Science, Towards Model-checking Probabilistic Timed Automata against Probabilistic Duration Properties.
9. Kattenbelt, M., Kwiatkowska, M., Norman, G., Parker, D. (2010), A game-based abstraction-refinement framework for Markov decision processes. Formal Methods in System Design 36(3), 246–280

10. Marta Kwiatkowska, 2004, *Model checking for probability and time: from theory to practice.*
11. Marta Kwiatkowska, Gethin Norman, David Parker (2010), PRISM 4.0: Verification of Probabilistic Real-time Systems, International Conference on Computer Aided Verification (CAV'11)