Tạp chí Tin học và Điều khiển học, T.30, S.2 (2014), 148–162

ĐƠN ĐỊNH VÀ TỐI TIỂU HOÁ OTOMAT KHOẢNG*

BÙI VŨ ANH

Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội; vuanh@vnu.edu.vn

Tóm tắt. Các công trình trước đây chúng tôi đã đề xuất mô hình otomat khoảng và sử dụng mô hình này trong một số bài toán tối ưu [3] và lập lịch [4] cũng như các bài toán trong lĩnh vực bảo mật [5]. Trong nghiên cứu này, bài báo sẽ tập trung vào hai bài toán đơn định và tối tiểu hoá otomat khoảng. Các bài toán nhỏ hơn cũng được giải quyết là: tách/ghép các khoảng trên các cung của otomat mà không làm thay đổi ngôn ngữ được đoán nhận, loại các trạng thái không đạt được (có và không có yếu tố khoảng). Những bài toán này được dùng trong việc giải bài toán chính: đơn định hoá và tối tiểu hoá otomat khoảng.

Từ khóa. Khoảng, otomat khoảng, đơn định hoá, tối tiểu hoá.

Abstract. In the previous papers, we have proposed duration automata model and used the model in optimization [3], schedule problems [4] and in problems of the security area [5]. In this paper, we study problem of determining and minimizing the duration automata. Some additional problems are also considered to show how separate and merge durations on the arcs of the automata without changing accepted languages, remove unreachable states (with and without duration meaning). These problems are applied to solve the main problem: determining and minimizing duration automata.

Key words. Duration, duration automata, deterministic, minimization.

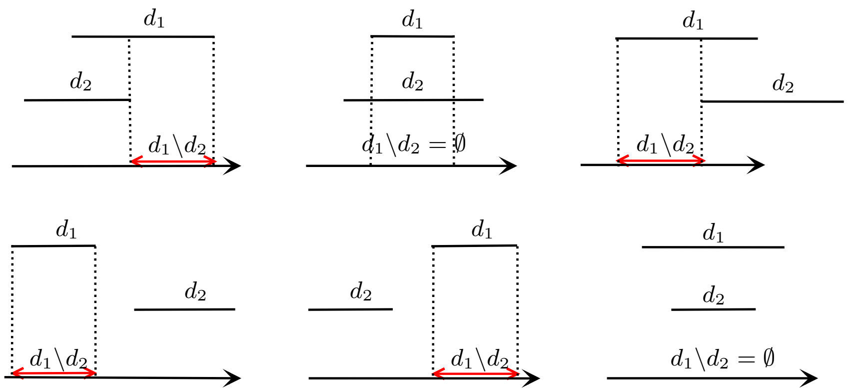
1. CÁC PHÉP TOÁN TRÊN KHOẢNG

Gọi $\varepsilon = [0, +\infty)$, $\mathcal{D} = \{\emptyset, \varepsilon\} \cup \{[l,u], [l,u), (l,u], (l,u) \mid l, u \in \mathbb{Q}\}$ là tập các khoảng. Ta nói giá trị $t \in \mathbb{R}$ thuộc khoảng đóng $d = [l,u]$ nếu $l \le t \le u$, viết là $t \in d$. Nếu khoảng d là mở thì dấu bằng không xảy ra và thay dấu ngoặc vuông ("[,]") bằng ngoặc tròn ("(,)") tương ứng. Nếu không tồn tại giá trị t để $t \in d$ thì d là khoảng trống, ký hiệu là $\emptyset$.
Cho $d_1, d_2 \in \mathcal{D},\ d_1 = [l_1, u_1], d_2 = [l_2, u_2]$, các phép toán trên khoảng được định nghĩa:
- Phép giao: $d_1 \cap d_2 = \{t\ :\ t \in d_1 \text{ và } t \in d_2\}$.
- Đoạn con: $[l_1, u_1] \subseteq [l_2, u_2] \ \Leftrightarrow\ l_2 \le l_1 \le u_1 \le u_2$.
- Phép hiệu liên tục:

$$d_1 \backslash_m d_2 = \begin{cases} d_1 \backslash\ d_2 & \text{nếu } d_1 \neq \emptyset,\ \max\{l_1, l_2\} \le \min\{u_1, u_2\} \text{ và } \ d_2 \not\subseteq d_1 \\ \emptyset & \text{(không xác định) nếu ngược lại.} \end{cases}$$

Hiệu của hai khoảng là phần còn lại *liên tục* của khoảng thứ nhất sau khi bớt khoảng thứ hai. Như vậy trong trường hợp khoảng thứ nhất chứa khoảng thứ hai hoặc khoảng thứ nhất là rỗng, hiệu sẽ là khoảng rỗng. Hiệu hai khoảng được minh hoạ ở Hình 1.

**Bài báo được thực hiện dưới sự hỗ trợ từ đề tài TN 13-02.*



Hình 1. Hiệu liên tục của hai khoảng

- Phép hợp liên tục: Hợp liên tục của hai khoảng là khoảng lớn nhất chứa cả hai khoảng trong trường hợp hai khoảng có giao khác rỗng, trái lại, hợp liên tục không xác định.

$$d_1 \sqcup d_2 = \begin{cases} [\min\{l_1, l_2\}, \max\{u_1, u_2\}] & \text{nếu } d_1 \cap d_2 \neq \emptyset \\ \emptyset & \text{(không xác định) nếu ngược lại.} \end{cases}$$

2. OTOMAT KHOẢNG

2.1. Định nghĩa

Trong một mạng giao thông, khi đi trên một cung đường, phương tiện đi lại thường bị ràng buộc về tốc độ tối thiểu và tốc độ tối đa hay thời gian cho phép xe cộ đi có dạng từ giờ nào đến giờ nào. Chẳng hạn chỉ cho xe tải chạy qua từ 22h ngày hôm trước đến 6h sáng ngày hôm sau (tương đương với cấm xe tải từ 6h-22h). Trong lập lịch cho các công việc, một công việc thường được gắn với thời gian có thể hoàn thành nó, hoặc ước lượng thời gian tối thiểu một công việc sẽ hoàn thành và ước lượng chắc chắn thực hiện công việc đó không vượt quá một giá trị ngưỡng. Trong kinh tế, khi thực hiện một dự án, mức kinh phí sử dụng có thể nằm trong một khoảng (mức tối thiểu để có thể triển khai được và mức tối đa không được phép vượt qua)... Như vậy có thể thấy ràng buộc dạng khoảng rất thường gặp trong thực tế, có thể mang ý nghĩa thời gian hay phi thời gian. Trường hợp riêng của khoảng là khi nó thu chuyển về dạng điểm, tức là hai mốc của khoảng trùng nhau. Trong phần này ta sẽ xây dựng một mô hình mà dùng nó, ta có thể mô hình hoá các hệ thống có ràng buộc dạng khoảng như nêu trên.

Định nghĩa 2.1. Otomat khoảng (DA) M là bộ 7: $\langle S, \Sigma, \triangle, \nabla, q, R, F\rangle$ trong đó:

- S là tập hữu hạn các trạng thái của otomat, $q \in S$ là trạng thái khởi đầu.
- $\Sigma, \triangle, \nabla$ lần lượt là các bảng chữ cái hành động *internal*, *input* và *output*. Ký hiệu $\mathcal{A} = \Sigma \cup \triangle \cup \nabla$.
- $R \subseteq S \times \mathcal{A} \times \mathcal{D} \times S$ là quan hệ chuyển gắn thời gian, trong đó $\mathcal{D}$ là tập hợp các khoảng. Với mỗi phép chuyển $e = (s, a, d, s') \in R$, a là hành động *output* của s và cũng là hành động *input* của s', $d \in \mathcal{D}$ là ràng buộc khoảng. Nếu $s = s'$ thì e là khuyên (vòng).

- $F \subseteq S$ là tập các trạng thái đoán nhận (trạng thái kết).

Gọi $M = \langle S, \Sigma, \triangle, \nabla, q, R, F\rangle$ là DA. Xét dãy chuyển r

$$r : s_0 \xrightarrow{(a_1,d_1)} s_1 \xrightarrow{(a_2,d_2)} \dots s_{n-1} \xrightarrow{(a_n,d_n)} s_n.$$

Nếu s_0 là trạng thái khởi đầu và với mỗi $i: \ 1 \leq i \leq n, \ \exists e_j = (s_{i-1}, a_i, d_i, s_i) \in R$, thì $\widetilde{p} = (s_0, d_1)(s_1, d_2)\dots \ (s_{n-1}, d_n)$ được gọi là một *d-đường* trong M và dãy

$$\widetilde{w} = (a_1, d_1)(a_2, d_2)(a_3, d_3)\dots(a_n, d_n)$$

được gọi là một d-từ ứng với d-đường nói trên. Nếu tồn tại dãy $t_i \in d_i$ với mọi $i = 1..n$ thì dãy $\overline{w} = (a_1, t_1)(a_2, t_2)\dots(a_n, t_n)$ được gọi là một t-từ thoả $\widetilde{w}$, khi đó $\widetilde{w}$ được gọi là d-từ thoả được. Trường hợp $s_n \in F$ thì $\widetilde{p}$ được gọi là một d-đường của M và $\widetilde{w}$ tương ứng được gọi là d-từ của M. Tập hợp các d-từ của M được gọi là ngôn ngữ do M đoán nhận, ký hiệu là $L(M)$. Mỗi (a_i, d_i) và (a_i, t_i) lần lượt được gọi là một d-nhãn và t-nhãn. d-nhãn rỗng được định nghĩa là $\varepsilon_\infty = (\wedge, \varepsilon)$ trong đó $\wedge$ là từ rỗng trong otomat truyền thống.

Mô hình otomat với cách tiếp cận dùng khoảng như ở trên đã được nghiên cứu trong một số công trình như [10] của Dang Van Hung và Bui Vu Anh, [7] của Deepak D'Souza và P. S. Thiagarajan. So sánh với mô hình trong [7], otomat khoảng IA (interval automata) được xây dựng dựa trên otomat Büchi với bảng chữ bổ sung thêm thành phần ràng buộc khoảng có hai đầu mút là hai số thực. Bản chất ngôn ngữ do otomat khoảng IA đoán nhận sẽ là ngôn ngữ vô hạn và hành vi của các hệ thống được mô hình hoá bởi IA sẽ là các hệ có hành vi vô hạn. Trong [7], tác giả nghiên cứu khía cạnh ngôn ngữ của IA và chỉ ra ngôn ngữ được đoán nhận bởi IA là một tập con của ngôn ngữ thời gian được đoán nhận bởi otomat thời gian [1] (Timed Automata - TA). Khác với IA, DA sử dụng các đầu mút khoảng là các số nguyên và gán các ngữ nghĩa khoảng khác nhau cho d-từ tùy theo các tình huống trong thực tế. Có thể nói ngôn ngữ được đoán nhận bởi DA cũng không nằm ngoài lớp các ngôn ngữ được đoán nhận bởi TA. Do các mốc thời gian khoảng được sử dụng là các số nguyên nên DA và IA đều có thể mô tả các hệ thống thời gian thực với các mốc ràng buộc thời gian khoảng là các số nguyên. Ngoài ra, do các đầu mút thời gian của IA là các số thực nên không tận dụng được các kết quả đã có của TA như trong [1] khi nghiên cứu về tính đạt được và tính đóng của các phép toán trên ngôn ngữ... Thực tiễn, đặc biệt là khi xử lý mô hình trên máy tính, ít khi mốc là số thực được sử dụng, hoặc nếu có thì cũng có cách để chuyển sang mốc nguyên.

Cho dãy các phép chuyển r, có một số cách gán ngữ nghĩa cho $d-$từ tương ứng với r được xác định như sau:

- *Ngữ nghĩa 1*: Gọi t_i là thời gian tiêu tốn để hoàn thành hành động a_i trong dãy r. Nếu $t_i \in d_i$ thì dãy $\overline{w} = (a_1, t_1)(a_2, t_2)\dots(a_n, t_n)$ được gọi là một t-từ thoả $\widetilde{w}$. Ngữ nghĩa này được gọi là ngữ nghĩa (theo) thời gian khoảng, lấy mốc tính thời gian là khi hành động bắt đầu thực hiện.
- *Ngữ nghĩa 2*: Gọi t_i là thời điểm hoàn thành hành động a_i trong dãy r. Xuất phát tại s_0 với t_0, nếu $t_i \in d_i$ với mọi $i = 1..n$ thì dãy $\overline{w} = (a_1, t_1)(a_2, t_2)\dots(a_n, t_n)$ được gọi là một t-từ thoả $\widetilde{w}$. Ngữ nghĩa này được gọi là ngữ nghĩa (theo) thời gian điểm, lấy mốc tính thời gian là khi hành động đầu tiên của dãy bắt đầu thực hiện.
- *Ngữ nghĩa 3*: Gọi t_i là thời điểm hệ thống thực hiện xong hành động a_i và chuyển sang trạng thái s_i trong dãy r. Nếu $t_i \in d_i$ với mọi $i = 1..n$ thì dãy $\overline{w} = (a_1, t_1)(a_2, t_2)\dots(a_n, t_n)$ được gọi là một t-từ thoả $\widetilde{w}$. Ngữ nghĩa này được gọi là ngữ nghĩa (theo) thời điểm chuyển trạng thái. Như vậy, hệ thống thực hiện xong hành động a_i nhưng có thể chưa chuyển sang trạng

thái mới; ngược lại, để có thể chuyển trạng thái, hành động a_i phải được thực hiện xong. Nói cách khác, ngữ nghĩa này quan tâm đến thời điểm chuyển trạng thái sau thời điểm hoàn thành hành động. Otomat sau khi hoàn thành hành động có thể được chấp nhận một thời gian trễ trước khi nó chuyển trạng thái.

Cho một d-từ, ngữ nghĩa theo thời gian khoảng luôn tồn tại (ngoại trừ khoảng ràng buộc là rỗng) hoặc công việc không hoàn thành trong khoảng thời gian ràng buộc. Nhưng khi xét với ngữ nghĩa thời gian điểm, một số trường hợp tồn tại ngữ nghĩa, một số trường hợp không và điều này cũng tương tự đối với ngữ nghĩa theo thời điểm chuyển trạng thái. Tuỳ theo yêu cầu, mục đích mà có thể chọn một loại ngữ nghĩa khi mô tả các hệ thống trong thực tiễn. Ngữ nghĩa 2 sẽ là trường hợp riêng của ngữ nghĩa 3 nếu thời điểm thực hiện xong hành động và thời điểm chuyển trạng thái là trùng nhau (không có độ trễ). Với ngữ nghĩa 3, ràng buộc d_i đối với một cung đi vào một trạng thái sẽ mang ý nghĩa đó là thời điểm sớm nhất cần đạt đến trạng thái đó và thời điểm muộn nhất phải rời khỏi trạng thái đó nếu đi vào theo cung được gắn với ràng buộc khoảng d_i. Otomat thay đổi trạng thái nếu nó thoả điều kiện rời khỏi một trạng thái và thoả cả điều kiện đến được một trạng thái mới. Như vậy, ràng buộc khoảng trên cung nó đã đến một trạng thái (và đang ở trạng thái đó) cùng ràng buộc khoảng trên cung nó sẽ đi đến phải đồng thời được thoả mãn. Để có được điều đó, ràng buộc khoảng trên hai cung đã đến và sẽ đi ra khỏi một trạng thái phải giao nhau. Do đó, nếu t là thời điểm chuyển trạng thái từ s sang s' thì trước thời điểm t, hệ thống ở trạng thái s, sau thời điểm t hệ thống sẽ ở trạng thái s'. Ở đây mỗi hành động chỉ có một trong hai khả năng là nếu thực hiện thì sẽ thành công và không thực hiện được.

Otomat khoảng được định nghĩa trên ràng buộc khoảng tổng quát, yêu cầu các khoảng liên tiếp trên một đường là giao nhau hoặc không giao nhau. Mô hình mà bài báo nghiên cứu xét trường hợp các ràng buộc khoảng trên cung đi vào và đi ra khỏi một trạng thái là giao nhau nếu muốn đến/rời khỏi trạng thái đó, trái lại thì sẽ rơi vào trạng thái treo NULL (tắc nghẽn). Đây là trạng thái đặc biệt nằm trong không gian các trạng thái của otomat.

Có thể nhận thấy nếu dãy số thực $t_i,\ i = 1..n$ diễn tả khoảng thời gian tiêu tốn để hoàn thành hành động tương ứng a_i thì dãy này không có tính chất đơn điệu tăng. Nhưng với một dãy số thực $t_i,\ i = 1..n$ diễn tả thời điểm hoàn thành hành động tương ứng a_i, hoặc diễn tả thời điểm chuyển trạng thái thì các dãy này sẽ là dãy đơn điệu tăng. Nếu quan sát thực hiện của một dãy hành động thì luôn có ngữ nghĩa 1. Với ngữ nghĩa 2, tại mọi vị trí i của d-từ cần có $l_i \le u_{i+1}$ và với ngữ nghĩa 3, điều kiện sẽ là $\max\{l_i, l_{l+1}\} \le \max\{u_i, u_{i+1}\}$ với $i = 1..n-1$.

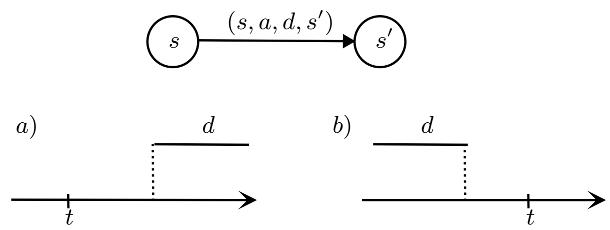
Với các hệ thời gian thực, ta chỉ có thể quan sát hoạt động của hệ theo thời gian. Do đó, t-từ chính là một dãy các quan sát (một thể hiện) tại các thời điểm của hệ. Các cách quan sát khác nhau sẽ dẫn đến các t-từ khác nhau ứng với các ý nghĩa được gắn cho t-từ. Một d-từ có thể xác định nhiều t-từ thoả nó. Trường hợp tồn tại t-từ thoả d-từ (theo một trong các ngữ nghĩa) thì d-từ là thoả được (theo ngữ nghĩa đó). Nếu không tồn tại t-từ như vậy, d-từ là không thoả được (ứng với một ngữ nghĩa, nhưng vẫn có thể là một t-từ ứng với một ngữ nghĩa khác). Ngữ nghĩa ở đây chính là giá trị quan sát của dãy t_i và cách thức otomat vận hành để tạo ra dãy này. Với ngữ nghĩa 2 và 3, otomat rơi vào trạng thái treo (NULL) nếu otomat đang ở trạng thái s tại thời điểm t, mà không tồn tại giá trị $t' > t$ nào trên các cung ra khỏi s để ràng buộc khoảng d của nó được thoả (tức là $t' \notin d$ với mọi $t' > t$).

Có thể thấy với một d-từ thoả được theo ngữ nghĩa 2 thì có thể xác định một dãy t-từ tương ứng theo ngữ nghĩa 1 khi chọn $t'_0 = 0,\ t'_i = t_i - t_{i-1}$ với $i = 1..n$. Tất nhiên ràng buộc khoảng d_i cũng phải thay đổi để phù hợp với điều kiện $t'_i \in d_i$ thay cho $t_i \in d_i$.

Trong [10] và [7], để theo dõi được thời gian thực hiện của hành động, giá trị (biến) đồng hồ được khởi tạo lại bằng 0. Như vậy, giá trị thời gian t_i ở mỗi phép chuyển chính là thời gian thực hiện hành động a_i trong dãy các phép chuyển. Việc đặt lại giá trị t_i về 0 trước khi mỗi hành động bắt đầu sẽ phù hợp với việc gán ngữ nghĩa 1. Trong ứng dụng lập lịch công việc trên máy tính, thời gian triển khai công việc được tính từ khi công việc đó bắt đầu được triển khai nên có thể sử dụng ngữ nghĩa 1. Trong quan sát vận hành của một mạng giao thông, ngữ nghĩa sử dụng lại là ngữ nghĩa 2. Trong phần nghiên cứu về khía cạnh ngôn ngữ của otomat khoảng ở đây, bài báo tập trung vào nghiên cứu ngôn ngữ theo ngữ nghĩa thời điểm chuyển trạng thái, tức là yêu cầu mô hình có các khoảng liên tiếp phải giao nhau thì mới *đi qua* được.

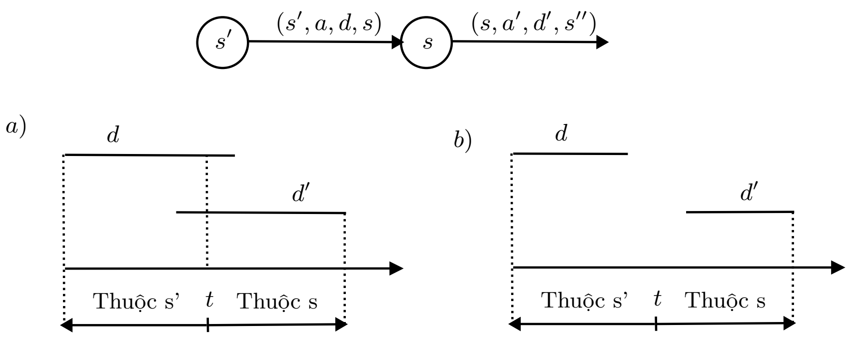
Nhận xét: Trong mô hình otomat khoảng, nếu mọi ràng buộc khoảng đều là ε, tức là ràng buộc luôn có thể thỏa mãn và không phân biệt loại chữ cái hành động thì mô hình trở thành mô hình otomat truyền thống. Nói cách khác, mô hình otomat truyền thống là trường hợp riêng của mô hình otomat khoảng. Đồng thời, cũng với ràng buộc khoảng là ε trên mọi cung thì nó trở thành mô hình otomat IO [11] và otomat IO cũng trở thành trường hợp riêng của DA. Nếu trên hai cung liên tiếp của một đường đi giao nhau bằng rỗng nhưng biên phải cung trước nhỏ hơn biên trái cung sau thì có thể bổ sung thêm một hành động chờ cho otomat để giá trị thời gian trôi đến khi gặp biên trái của ràng buộc khoảng trên cung tiếp theo và chuyển sang cung đó. Tuy nhiên, nếu giao các khoảng là rỗng nhưng biên trái khoảng sau nhỏ hơn biên phải của khoảng trước thì otomat sẽ không chuyển trạng thái tiếp được. Các phần sau đây nếu không có lưu ý gì thì ngữ nghĩa sử dụng sẽ là ngữ nghĩa theo thời điểm chuyển trạng thái.

Hoạt động của otomat được mô tả như sau. Otomat xuất phát từ trạng thái khởi đầu q tại thời điểm t_0 (thường được gán bằng 0). Cấu hình của M tại thời điểm t là cặp (s, t), trong đó $s \in S$ và $t \in \mathbb{R}^+$ chỉ ra M đang ở trạng thái s tại thời điểm t và lân cận ngay trước t (trừ cấu hình xuất phát). Cấu hình khởi đầu của M là (q, t_0) và cấu hình đoán nhận của M là (s, t) với $s \in F$ và $t \geq 0$.



Hình 2. Trạng thái di chuyển được a) và trạng thái treo b) theo ngữ nghĩa 2.

Hình 2 minh hoạ otomat ở một trạng thái s với ràng buộc thời gian trên cung đi ra d và cấu hình hiện tại là (s, t). Ở hình 2a), otomat không bị rơi vào trạng thái tắc nghẽn (giá trị thời gian thực hiện hành động trước nhỏ hơn thời điểm cho phép thực hiện hành động tiếp theo). Trong Hình 2b) minh hoạ tình huống otomat ở trạng thái mà ràng buộc thời gian trên cung đi ra d không thể thoả (giá trị thời gian hoàn thành hành động đã vượt quá ràng buộc cho phép thực hiện hành động kế tiếp). Khi đó otomat không thể chuyển sang trạng thái tiếp mà rơi vào trạng thái tắc nghẽn. Như vậy, việc otomat có chuyển được đến trạng thái tiếp theo hay không phụ thuộc vào ràng buộc khoảng trên cung đi ra và thời điểm hoàn thành hành động.



Hình 3. Trạng thái di chuyển được và trạng thái treo theo ngữ nghĩa 3

Hình 3 minh hoạ ngữ nghĩa 3. Trường hợp 3a) là có thể thay đổi trạng thái được còn 3b) là không thay đổi trạng thái được (hay bị treo, tắc nghẽn).

Trong mô hình tổng quát, do không quan tâm riêng đến một loại hành động trong hay ngoài nên có thể giả thiết các hành động được gộp chung vào một tập hợp, gọi là tập nhãn các hành động $\mathcal{A} = \Sigma \cup \triangle \cup \nabla$. Các cặp $(a_i, d_i) \in (\mathcal{A}, \mathcal{D})$ và $(a_i, t_i) \in (\mathcal{A}, \mathbb{R}^+)$ được gọi lần lượt là một d-nhãn và t-nhãn. Mỗi phần tử $x = (a_1 \dots a_k, d) \in (\mathcal{A}^*, \mathcal{D})$ và $y = (a_1 \dots a_k, t) \in (\mathcal{A}^*, \mathbb{R}^+)$ được gọi theo thứ tự là d-xâu và t-xâu, trong đó $\mathcal{A}^*$ là tập các xâu tạo thành từ dãy liên tiếp các ký hiệu trong $\mathcal{A}$, $\mathbb{R}^+$ là tập các số thực không âm.

Định nghĩa 2.2.
- Hai d-từ $w_1,\ w_2$ là bằng nhau nếu mỗi t-từ thoả w_1 đều thoả w_2 và ngược lại.
- Hai d-ngôn ngữ $L_1,\ L_2$ là bằng nhau nếu với mỗi d-từ thuộc L_1 đều thuộc L_2 và ngược lại.

Định nghĩa 2.3. Hai DA được gọi là tương đương nếu chúng đoán nhận cùng một ngôn ngữ khoảng, hay hai ngôn ngữ khoảng do chúng đoán nhận là bằng nhau.

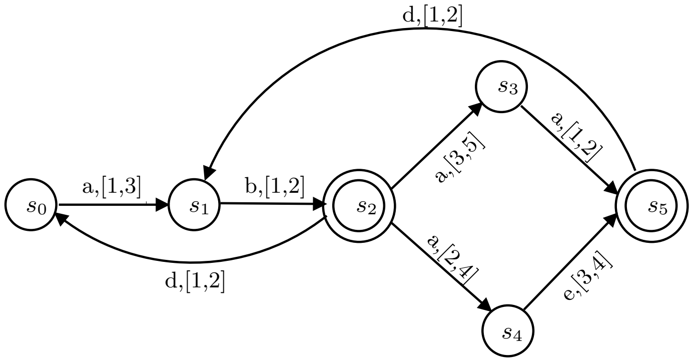
2.2. Otomat đơn định và đa định

Định nghĩa 2.4. Một DA $M = \langle S, \Sigma, \triangle, \nabla, q, R, F\rangle$ được gọi là đơn định khi $\forall s \in S,\ a \in \mathcal{A}$, nếu (s, a, d', s'), $(s, a, d'', s'') \in R$ và $d' \cap d'' \neq \emptyset$ thì $s'' \equiv s'$. Trái lại, DA được gọi là đa định hay không đơn định.

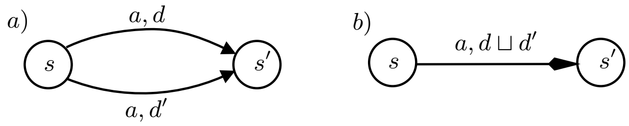
Theo Định nghĩa 2.4, nếu DA có hai cung đi ra từ một trạng thái trên đó ghi cùng ký hiệu mà có giao các ràng buộc khoảng khác rỗng thì chúng phải đi đến cùng một trạng thái. Điều này sẽ đảm bảo cho việc với cùng một t-từ là đầu vào cho otomat thì chỉ dẫn đến duy nhất một trạng thái cấu hình.

Hình 4, ở trạng thái s_2 có hai cung $(s_2, a, [2, 4], s_4)$ và $(s_2, a, [3, 5], s_3)$ ghi cùng ký hiệu chữ cái và giao của ràng buộc thời gian khác rỗng nhưng đi đến hai trạng thái khác nhau. DA này là đa định. Có thể nhận thấy t-từ $(a, 1.2)(b, 3.3)(a, 3.5)$ dẫn đến hai trạng thái khác nhau là s_3 và s_4.

Ta đưa ra thêm một cách định nghĩa khác về quan hệ chuyển. Quan hệ chuyển có thể được xem như là một hàm chuyển $\delta : S \times (A, \mathcal{D}) \to S$ trong đó với mỗi cung chuyển $e = (s, a, d, s') \in R$ xác định một quan hệ chuyển $\delta(s, (a, d)) = s'$. Hàm chuyển mở rộng $\widehat{\delta} : S \times (A, \mathcal{D})^* \to S$ được định nghĩa tương tự otomat truyền thống trong đó với từ wa, $a \in (\mathcal{A} \times \mathcal{D})$, $w \in (\mathcal{A} \times \mathcal{D})^*$,



Hình 4. Otomat khoảng đa định



Hình 5. Ghép cung

$\widehat{\delta}(s, wa) = \delta(\widehat{\delta}(s, w), a)$ với $\delta(s, \varepsilon_\infty) = s$. Khi cho trước otomat khoảng, một trong hai dạng này sẽ đều được sử dụng và xem là tương đương nhau. Giống như otomat truyền thống, $\widehat{\delta}(q, w)$ sẽ bao gồm tất cả các trạng thái p sao cho có thể đi đến được theo từ w, bao gồm cả việc đi qua các cung ε_∞.

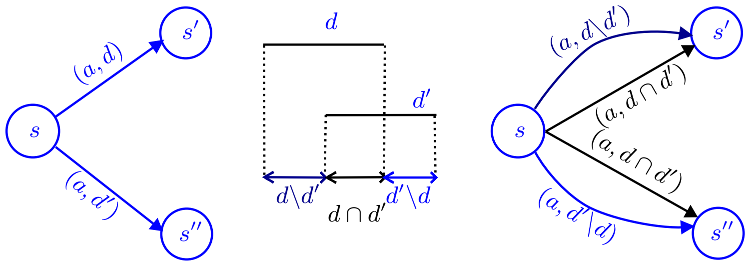
Như vậy, với otomat đơn định, hàm chuyển xác định không quá một giá trị trạng thái, tức là $\delta : S \times (A, \mathcal{D}) \to S$, trong khi với otomat đa định, hàm chuyển tương ứng này sẽ là $\delta_N : S \times (A, \mathcal{D}) \to 2^S$. Hàm chuyển mở rộng cho otomat đa định cũng được định nghĩa tương tự: $\widehat{\delta_N}(s, wa) = \widehat{\delta_N}(\widehat{\delta_N}(s, w), a)$.

Hệ quả 2.1. *Cho otomat M với hàm chuyển trạng thái δ. Nếu $\delta(s, (a, d')) = s'$ và $\delta(s, (a, d'')) = s'$ mà $d' \subseteq d''$ thì việc loại bỏ quan hệ chuyển $\delta(s, (a, d')) = s'$ không thay đổi ngôn ngữ được M đoán nhận.*

Chứng minh. Do do mọi t-từ thoả d-từ chứa (a, d') đều đã được chứa trong d-từ chứa (a, d'') nên việc loại bỏ không làm mất những t-từ được đoán nhận. Ngược lại, việc loại bỏ cung không làm tăng các từ được đoán nhận. Vậy ngôn ngữ được M đoán nhận là không đổi. ■

Hệ quả 2.2. *Cho otomat M với hàm chuyển trạng thái δ. Nếu thay cặp quan hệ chuyển $\delta_N(s, (a, d)) = s'$ và $\delta_N(s, (a, d')) = s'$, với $d \cap d' \neq \emptyset$ bằng quan hệ $\delta_N(s, (a, d \sqcup d')) = s'$ thì ngôn ngữ do otomat đoán nhận không thay đổi.*

Chứng minh. Giả sử $d = [l, u]$, $d' = [l', u']$. Nếu khoảng thời gian ở cung hợp được tách ra làm 3 khoảng nhỏ: $d_1 = d \backslash d'$ tức là khoảng gồm các giá trị thuộc d nhưng không thuộc d', $d_2 = d \cap d'$ là khoảng gồm các giá trị thuộc cả d và d', và $d_3 = d' \backslash d$ là khoảng gồm các giá trị thuộc d' nhưng không thuộc d. Như vậy, mỗi khi đi từ s đến s' thì giá trị t của t-từ tương ứng sẽ rơi vào một trong 3 khoảng vừa chia. Vậy việc hợp cung chỉ thu gọn số lượng cung của otomat mà không làm thay đổi ngôn ngữ do otomat đoán nhận. ■



Hình 6. Tách khoảng

Hệ quả 2.3. *Cho otomat M với hàm chuyển trạng thái δ. Nếu thay mọi cặp quan hệ $\delta_N(s,(a,d)) = s'$ và $\delta_N(s,(a,d')) = s''$ với $d \cap d' \neq \emptyset$ và $s' \neq s''$ bằng các quan hệ:*

$$e'_1 : \delta_N(s,(a,d\backslash d')) = s', \qquad e'_2 : \delta_N(s,a,d\cap d') = s',$$
$$e'_3 : \delta_N(s,(a,[d\cap d')) = s'', \qquad e'_4 : \delta_N(s,(a,d'\backslash d)) = s''$$

thì otomat thu được đoán nhận cùng ngôn ngữ với otomat ban đầu.

Chứng minh. Các khoảng giao/chứa nhau được tách thành 3 khoảng rời rạc. Việc tách này tạo ra các khoảng nhỏ hơn và mỗi t-từ thoả quan hệ chuyển trước khi tách có giá trị nhãn t tương ứng chỉ thỏa một trong ba khoảng trong các quan hệ chuyển thay thế. Đồng thời, hợp các giá trị t thỏa 3 khoảng mới tạo ra chính là $d \sqcup d'$. Do đó, việc thay thế cung cũng không làm thay đổi ngôn ngữ được đoán nhận.

Trong các quan hệ được thay vào ứng với các cặp quan hệ đa định, các quan hệ e'_2, e'_3 chính là các quan hệ làm cho otomat đa định: cùng hành động a, trùng nhãn thời gian và đi đến hai trạng thái khác nhau. Tại mọi trạng thái, nếu $\delta_N(s,a,d) = s'$ và $\delta_N(s,a',d') = s'$ thì $(a,d) \neq (a',d')$ và nếu $\delta_N(s,a,d) = s'$ và $\delta_N(s,a,d) = s''$ thì $s' \neq s''$. ■

2.3. Bài toán đơn định hoá otomat

Bài toán: Cho trước một otomat khoảng M. Đơn định hóa otomat đó.

So sánh với bài toán đơn định của otomat truyền thống, sau khi đã rời rạc hoá các khoảng và tách, gộp cung, ta có thể coi nhãn (bộ hai thành phần) trong otomat khoảng như một nhãn mới thì có thể sử dụng phương pháp như với otomat truyền thống để đơn định hoá. Trong hình 6, hai cung $(s,a,d\cap d',s')$ và $(s,a,d\cap d',s'')$ chính là hai cung làm cho otomat đa định.

Thuật toán đơn định hoá otomat khoảng.

Input: Otomat khoảng không đơn định $M_N = \langle S_N, \Sigma, \nabla, \triangle, q_N, \delta_N, F_N\rangle$.

Output: Otomat khoảng đơn định $M_D = \langle S_D, \Sigma, \nabla, \triangle, Q_0, \delta_D, F_D\rangle$ đoán nhận cùng ngôn ngữ với otomat M.

1. Kiểm tra nếu M_N đơn định, thuật toán dừng.
2. Loại bỏ các quan hệ chuyển chứa nhau: Giả sử $\delta_N(s,a,d') = s'$, $\delta_N(s,a,d'') = s'$ mà $d' \subseteq d''$ thì loại bỏ quan hệ chuyển $\delta_N(s,a,d') = s'$ (Bổ đề 2.1).
3. Ghép các quan hệ chồng lặp: Với mọi cặp quan hệ chuyển $\delta_N(s,a,d) = s'$ và $\delta_N(s,a,d') = s'$, nếu $d \cap d' \neq \emptyset$ thì thay hai quan hệ trên bằng quan hệ $\delta_N(s,a,d\sqcup d') = s'$ (Bổ đề 2.2).

4. Tách các khoảng: Với mọi cặp quan hệ $\delta_N(s, a, d') = s'$ và $\delta_N(s, a, d'') = s''$ mà $d' \cap d'' \neq \emptyset$ và $s' \neq s''$ thì thay hai quan hệ này bằng các quan hệ như trong Bổ đề 2.3. Quá trình tách này dừng khi không còn quan hệ cần tách.
5. Đơn định hoá:
   (a) Gọi D là tập tất cả các $d-$nhãn của otomat M_N sau khi tách và gộp các khoảng.
   (b) $Q_0 = \{q_N\}$ là trạng thái khởi đầu của M_D. Xây dựng hàm $T : 2^{S_N} \times D \to 2^{S_N}$ như sau:
      i. $\forall s \in S_N,\ \forall a \in D$: $T(\{s\}, a) = \{s' \in S_N \mid s' = \delta_N(s, a)\}$.
      ii. $\forall B \subseteq S_N,\ \forall a \in D$: $T(B, a) = \bigcup\limits_{s \in B} T(\{s\}, a)$.
   (c) Xây dựng các quan hệ chuyển: Đối với mỗi tập $B \subseteq S_N$ và với mọi $d-$nhãn $a \in D$, nếu $Q = T(B, a) \neq \emptyset$ thì $\delta_D(B, a) = Q$ và đưa B, Q vào tập trạng thái S_D (khởi tạo rỗng).
   (d) Trạng thái kết: $F_D = \{Q \in S_D \mid Q \cap F_N \neq \emptyset\}$.

Chứng minh. Để chứng minh tính đúng đắn của thuật toán, tức là $L(M_D) = L(M_N)$, ta sẽ chứng minh $\widetilde{\delta_D}(Q_0, w) = \widetilde{\delta_N}(q_N, w)$ với w là một $d-$từ. Hàm δ_D trả lại một tập các trạng thái trong S_N. Ta sẽ chứng minh quy nạp theo độ dài của w.
- Với độ dài $|w| = 0$, theo định nghĩa hàm chuyển mở rộng, ta có $\widehat{\delta}(q_N, w) = q_N$. Mặt khác, $\widehat{\delta_D}(Q_0, w) = \widehat{\delta_D}(\{q_N\}, w) = \widehat{\delta_N}(q_N, w) = q_N$. Vậy khẳng định đúng.
- Giả sử khẳng định đúng với $d-$từ w có độ dài n, xét $d-$từ wa, với a là $d-$nhãn. Giả sử $\widehat{\delta_N}(q_N, w) = \{p_1, \ldots, p_k\}$. Theo giả thiết quy nạp, $\widehat{\delta_D}(\{q_N\}, w) = \widehat{\delta_N}(q_N, w) = \{p_1, \ldots, p_k\}$. Lại có, $\widehat{\delta_N}(q_N, wa) = \delta_N(\widehat{\delta_N}(q_N, w), a) = \bigcup\limits_{i=1..k} \delta_N(p_i, a)$.
Mặt khác, $\widehat{\delta_D}(Q_0, wa) = \delta_D(\widehat{\delta_D}(\{q_N\}, w), a) = \delta_D(\{p_1, \ldots, p_k\}, a) = \bigcup\limits_{i=1..k} \delta_N(p_i, a)$ (theo cách xây dựng hàm T của thuật toán). Vậy $\widehat{\delta_D}(Q_0, wa) = \widehat{\delta_N}(p_N, wa)$ và khẳng định đúng với từ có độ dài $n+1$. Cả M_N và M_D chấp nhận $d-$từ w nếu tương ứng $\widehat{\delta_D}(Q_0, w)$ và $\widehat{\delta_N}(p_N, w)$ chứa ít nhất một trạng thái kết thuộc F_N. Theo quy nạp, thuật toán được chứng minh. ∎
Vậy ta có định lý.

Định lý 2.1. *Lớp ngôn ngữ đoán nhận bởi otomat khoảng đơn định và otomat khoảng đa định là trùng nhau.*

Thuật toán tìm các trạng thái của otomat mới thông qua việc duyệt các tập con của không gian các tập con của tập trạng thái của otomat ban đầu. Với mỗi tập con này, thuật toán xây dựng các quan hệ chuyển dựa trên việc duyệt lần lượt các nhãn trên các cung. Trường hợp xấu nhất phải vét tất cả các tập con của tập trạng thái S_N và tất cả các nhãn (mỗi cung một nhãn), nên độ phức tạp của thuật toán sẽ là $O(2^n \times m \times |D|)$ với n là số trạng thái và m là số nhãn của otomat ban đầu. Để tránh phải thực hiện tách cung nhiều lần, ta có thể chia các khoảng ra thành các khoảng đơn vị trong quá trình tách, nghĩa là không có khoảng nào giao với nhau khác rỗng. Tuy nhiên, thao tác tiền xử lý này không làm giảm bậc của độ phức tạp thuật toán mà chỉ thay đổi thời gian chạy của thuật toán khi cài đặt, không phải tách lại trên các khoảng đã tách.

Định nghĩa 2.5. (Thành phần không thời gian của DA) Cho otomat $M = \langle Q, \Sigma, \nabla, \triangle, q_0, R, F\rangle$.
- Otomat không thời gian tương ứng với M là $untime(M) = \langle Q, \Sigma, \nabla, \triangle, q_0, R', F\rangle$ trong đó

$R' = \{(s, a, s') \mid \exists d \in \mathcal{D} : (s, a, d, s') \in R\}$.
- Từ không gắn thời gian của từ $\widetilde{w} = (a_1, d_1)(a_2, d_2)(a_3, d_3) \ldots (s_n, d_n)$ là $a_1 a_2 a_3 \ldots a_n$.

Như vậy, trong trường hợp xét thành phần không thời gian của DA thì các khái niệm $d-$từ và t-từ trùng nhau và trở về khái niệm từ được định nghĩa cho otomat truyền thống.

2.4. Otomat khoảng tối tiểu

Định nghĩa 2.6. Hai trạng thái s và s' của otomat $M = \langle Q, \Sigma, \nabla, \triangle, q_0, \delta, F\rangle$, được gọi là tương đương, ký hiệu $s \approx s'$ nếu với mọi $d-$từ w, $\delta(s, w) \in F \Leftrightarrow \delta(s', w) \in F$. Hai trạng thái không tương đương gọi là hai trạng thái phân biệt.

Như vậy, với hai trạng thái bất kỳ, nếu $s \in F$ và $s' \notin F$ hoặc nếu $\exists$ $d-$nhãn a mà $\delta(s, a) = p,\ \delta(s', a) = \emptyset$ thì s và s' đều là không tương đương. Khái niệm trạng thái tương đương ở đây tương tự khái niệm trạng thái tương đương của otomat truyền thống.

Hệ quả 2.4. *Quan hệ giữa các trạng thái trong Định nghĩa 2.6 là một quan hệ tương đương.*

Chứng minh. Chứng minh suy ra từ Định nghĩa 2.6 về trạng thái không phân biệt.
- Phản xạ: $q \approx q$.
- Đối xứng: $q \approx p \ \Leftrightarrow \forall w \in (\mathcal{A}, \mathcal{D})^*, \widehat{\delta}(q, w) \in F \Leftrightarrow \widehat{\delta}(p, w) \in F \Leftrightarrow \forall w \in (\mathcal{A}, \mathcal{D})^*, \widehat{\delta}(p, w) \in F \Leftrightarrow \widehat{\delta}(q, w) \in F \Rightarrow p \approx q$.
- Bắc cầu:
+ $q \approx p \ \Leftrightarrow \forall w \in (\mathcal{A}, \mathcal{D})^*, \widehat{\delta}(q, w) \in F \Leftrightarrow \widehat{\delta}(p, w) \in F$.
+ $p \approx r \ \forall w' \in (\mathcal{A}, \mathcal{D})^*, \widehat{\delta}(p, w') \in F \Leftrightarrow \widehat{\delta}(r, w') \in F$
$\Rightarrow \forall w,\ w' \in (\mathcal{A}, \mathcal{D})^*, \widehat{\delta}(q, ww') \in F \Leftrightarrow \widehat{\delta}(r, ww') \in F$. ■

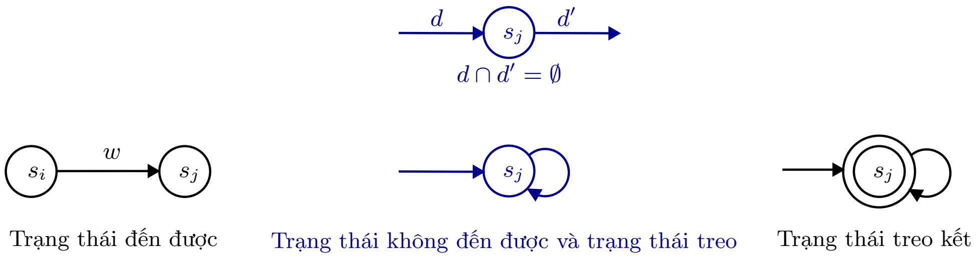
Hệ quả 2.5. *Cho $\delta(p, a) = p'$, $\delta(q, a) = q'$ với a là một $d-$nhãn. Khi đó nếu p' và q' là phân biệt thì p và q cũng phân biệt.*

Chứng minh. Do p', q' phân biệt nên tồn tại $d-$từ w: $\widehat{\delta}(q', w) \in F$ và $\widehat{\delta}(p', w) \notin F$ (hoặc ngược lại). Khi đó $\widehat{\delta}(q, aw) \in F$ và $\widehat{\delta}(p, aw) \notin F$ (hoặc ngược lại) nên q và p phân biệt. ■

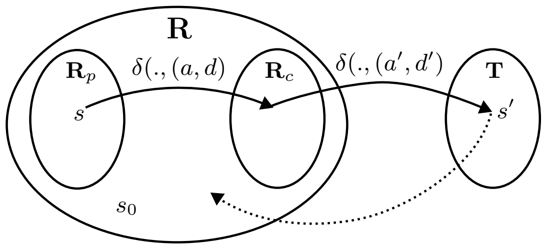
Định nghĩa 2.7. Cho $d-$ngôn ngữ $L \neq \emptyset$. Otomat khoảng có số trạng thái ít nhất đoán nhận L được gọi là otomat khoảng tối tiểu đoán nhận L.

Định nghĩa 2.8.
- Một trạng thái s của otomat khoảng $M = \langle Q, \Sigma, \nabla, \triangle, q_0, \delta, F\rangle$ được gọi là đến được nếu tồn tại một $d-$từ thoả được $\widetilde{x}$, sao cho $\widehat{\delta}(q_0, \widetilde{x}) = s$ trong đó $\widehat{\delta}$ là hàm mở rộng của hàm chuyển trạng thái. Ngược lại, trạng thái gọi là không đến được.
- Trạng thái s được gọi là trạng thái *treo* nếu nó không phải là trạng thái kết và dãy các phép chuyển theo mọi $d-$từ vào thoả được sẽ dừng ở đó, tức là $s \notin F$ và $\forall \widetilde{x} \in (\mathcal{A} \times \mathcal{D})^*,\ \delta(s, \widetilde{x}) = \emptyset$ hoặc $\delta(s, \widetilde{x}) = s$.

Dễ thấy các trạng thái không đến được (có thể là trạng thái kết) và những trạng thái treo không là trạng thái kết chỉ làm gia tăng số trạng thái của otomat nhưng không làm thay đổi ngôn ngữ do otomat đoán nhận. Do đó, ta có thể loại bỏ các trạng thái này mà không ảnh hưởng đến ngôn ngữ do otomat đoán nhận. Ta có bổ đề sau.



Hình 7. Trạng thái đến được và trạng thái treo



Hình 8. Loại trạng thái không đạt được

Hệ quả 2.6. *Otomat khoảng tối tiểu không chứa các trạng thái không đến được và trạng thái treo không kết.*

Bổ đề 2.6 cho ta điều kiện để thực hiện bước tiền xử lý trước khi tìm một otomat tối tiểu. Do đó, ta có thể giả thiết tất cả các trạng thái của otomat đều là đến được và không là trạng thái treo không là trạng thái kết khi tiến hành tối tiểu hóa otomat. Lưu ý là trạng thái kết cũng có thể là trạng thái không đến được. Trạng thái đến được và trạng thái treo được minh hoạ ở Hình 7. Đồng thời, otomat $untime(M)$ có các trạng thái không đến được và trạng thái treo không kết thì những trạng thái tương ứng này trong otomat khoảng cũng là các trạng thái không đến được và trạng thái treo không kết. Những trạng thái này cũng sẽ không tham gia trong việc xác định ngôn ngữ mà otomat khoảng đoán nhận. Do đó, trong quá trình tìm otomat tối tiểu, ta có thể sử dụng thuật toán tựa thuật toán Hopcroft [9] trên otomat $untime(M)$ xác định các trạng thái thuộc cùng lớp tương đương của otomat không thời gian.

Ý tưởng loại bỏ các trạng thái không đạt được minh hoạ trong Hình 8. Khởi đầu, q_0 là trạng thái đến được thuộc R_p và tập R_c các trạng thái đến được (có cung trực tiếp) từ q_0. Đây là các trạng thái đến được (R). Với mỗi trạng thái trong R_p có cung đến R_c và có thể đi tiếp được đến một trạng thái s' (không nằm trong tập đã đến được) thì trạng thái mới này là đến được, ta bổ sung vào R và được đưa vào phần gia tăng ở mỗi bước lặp T (xoá rỗng T sau mỗi lần thử trạng thái thuộc R_p). Sau đó ta đổi vai trò của R_c thành R_p và T thành R_c. Khi ở một bước lặp, phần gia tăng T là rỗng tức là không thể bổ sung thêm các trạng thái đến được, thuật toán dừng. Quá trình này dừng do tập trạng thái của otomat là hữu hạn và quá trình thực hiện không bị lặp lại trên các trạng thái đã được xét là đến được. Ta chỉ giữ lại các trạng thái có trong R cùng các cung nối các trạng thái trong R.

Thuật toán loại các trạng thái không đạt được.

Input: Otomat khoảng đơn định $M = \langle S, \Sigma, \nabla, \triangle, q_0, \delta, F \rangle$.
Output: Otomat khoảng chứa các trạng thái đạt được.

1. $D = \{(a, d) \ : \ \exists s, s' \in S, \delta(s, a, d) = s'\}$;
2. $R_p = \{q_0\}$;
3. $R_c = \emptyset$; for $(a, d) \in D$ $R_c = R_c \cup \{\delta(q_0, (a, d))\}$;
4. $R_c = R_c \backslash R_p$;
5. $R = R_p \cup R_c$;
6. done=1;
7. while (done)
8. { done=0;
9. for $s \in R_p$
10. { $T = \emptyset$;
11. for $(a, d) \in D$
12. if $\delta(s, (a, d)) \neq \emptyset$
13. { for $(a', d') \in D$
14. if $\delta(\delta(s, (a, d)), (a', d')) \neq \emptyset$
15. { $s' = \delta(\delta(s, (a, d)), (a', d'))$
16. if $(s' \notin S - R)$ && $(d \cap d' \neq \emptyset)$
17. { $R = R \cup \{s'\}$;
18. $T = T \cup \{s'\}$;
19. done=1; }
20. }
21. }
22. }
23. $R_p = R_c$;
24. $R_c = T$;
25. }

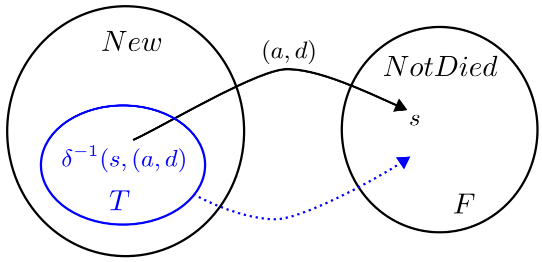
Độ phức tạp của thuật toán: $O(|D|^2 \times |S|)$ với D là tập các nhãn có trên các cung của otomat.

Tiếp theo, ta sẽ loại các trạng thái treo. Trong Hình 9, từ các trạng thái không bị treo (xuất phát từ tập trạng thái kết F) dò ngược trở lại tìm các trạng thái có cung đi đến các trạng thái này và đưa chúng vào tập trạng thái không treo (và là phần gia tăng New). Quá trình dò như vậy đến khi tìm được hết các trạng thái mà từ đó tồn tại một dãy các cung dẫn đến được trạng thái kết. Các trạng thái không được đưa vào $NotDied$ sẽ là các trạng thái treo và sẽ loại các trạng thái này cùng các cung đến đó. Thuật toán dừng khi phần bổ sung thêm New ở một bước là rỗng. Quá trình này dừng do không lặp lại các trạng thái đã bổ sung (duyệt theo cung) và tập trạng thái là hữu hạn.

Thuật toán loại bỏ các trạng thái treo.

Input: Otomat khoảng đơn định $M = \langle S, \Sigma, \nabla, \triangle, q_0, \delta, F \rangle$ có các trạng thái đều đạt được.
Output: Otomat khoảng không chứa các trạng thái treo không kết M'.

1. $NotDied = F$;
2. $New = F$;
3. $done = 1$;
4. while $(done)$



Hình 9. Loại trạng thái treo không kết

```
{   done = 0;
    T = ∅;
    for (s ∈ New)
    {   for ((a, d) ∈ D)
        if(δ^{-1}(s, (a, d))! = ∅) && δ^{-1}(s, (a, d)) ∉ NotDied
        {   T = T ∪ δ^{-1}(s, (a, d));
            done = 1; }
    }
    New = T\NotDied;
    NotDied = NotDied ∪ New;
}
```

Độ phức tạp của thuật toán: $O(|S|^2|D|)$.

Hệ quả 2.7. *Otomat khoảng đơn định có các trạng thái đến được là tối tiểu khi và chỉ khi tất cả các cặp trạng thái của nó là phân biệt.*

Chứng minh. Do otomat là đơn định nên không tồn tại $d \subseteq d', \forall s, s' \in S$ mà (s, a, d, s') và (s, a, d', s') đều thuộc quan hệ chuyển δ. Nếu ta đặt lại các nhãn trên các cung trong otomat khoảng đơn định bởi các nhãn mới thì otomat mới thu được chính là otomat truyền thống. Do đó việc chứng minh định lý này có thể dùng lại chứng minh của Hopcroft. ■

Bài toán: Cho $d-$ngôn ngữ được biểu diễn bởi DA đơn định $M = \langle Q, \Sigma, \nabla, \triangle, q_0, \delta, F\rangle$. Tối tiểu hóa M.

Nhận xét: Ta có thể sử dụng cải biên của thuật toán Hopcroft [8] để áp dụng cho otomat khoảng. Thuật toán này sẽ phân lớp các trạng thái tương đương vào cùng một lớp tương đương và xây dựng otomat tối tiểu dựa vào các lớp tương đương này. Với otomat khoảng, nhãn dùng để phân lớp tương đương tương tự như với thuật toán Hopcroft, nhưng gồm hai thành phần là nhãn và khoảng thay vì chỉ là một nhãn đơn.

Thuật toán tối tiểu hoá otomat khoảng.

Input: Otomat khoảng đơn định $M = \langle S, \Sigma, \nabla, \triangle, q_0, \delta, F\rangle$.

Output: Otomat khoảng tối tiểu $M_t = \langle S_t, \Sigma, \nabla, \triangle, s_0, \delta_t, F_t\rangle$ mà $L(M_t) = L(M)$.

1. Loại bỏ trạng thái không đến được (sử dụng một thuật toán duyệt các trạng thái).
2. Loại bỏ các trạng thái treo không là trạng thái kết.
3. Gọi D là danh sách tất cả các $d-$nhãn có trên các cung còn lại của otomat (sau bước 1, 2).

```
P = {F, Q\F}
W := {F}
while (W ≠ ∅) do
{   get(A, W) /* lấy tập A ra khỏi W * /
    for (a, d) ∈ D do
    {   X = {q : δ(q, (a, d)) ∈ A}
        for {Y ∈ P : X ∩ Y ≠ ∅, Y ⊈ X} do
        { thay thế Y trong P bởi X ∩ Y và Y\X
            if Y ∈ W
                thay Y trong W bởi X ∩ Y và Y\X
            else
                if |X ∩ Y| ≤ |Y\X| thêm X ∩ Y vào W else thêm Y\X vào W
        }
    }
}
```

Phần tối tiểu otomat có độ phức tạp tương ứng là $O(|S| \times log|S|)$, tương tự như [8, 9], tuy nhiên do cần loại trạng thái treo và trạng thái không đến được nên độ phức tạp của thuật toán tối tiểu hoá là $max\{O(|S| \times D), O(|D| \times S^2), O(|S| \times log|S|)\}$. Khi số lượng nhãn (hai thành phần) là hằng số, có thể coi độ phức tạp cỡ $O(|S| \times log|S|)$. Khi lập trình, có thể sử dụng kỹ thuật trong [2] để tránh trường hợp xấu nhất khi tìm otomat tối tiểu.

Định lý 2.2. *Thuật toán tìm otomat khoảng tối tiểu là đúng đắn.*

Chứng minh. Ngôn ngữ được đoán nhận không thay đổi khi loại bỏ trạng thái không đến được và trạng thái treo không kết vì các $d-$từ dẫn đến các trạng thái này không thể dẫn tiếp đến các trạng thái được đoán nhận. Nếu coi các nhãn kép của otomat khoảng thu được sau khi loại các trạng thái không đến được và trạng thái treo không kết là một nhãn đơn (thông qua phép đặt lại tên chẳng hạn) thì otomat khoảng trở thành otomat truyền thống và việc chứng minh otomat này là otomat tối tiểu sẽ tương tự như chứng minh cho otomat truyền thống. ■

3. KẾT LUẬN

Bài báo trình bày bài toán đơn định và tối tiểu trên otomat khoảng. Đây là hai bài toán kinh điển và được đặt lại cho mô hình otomat khoảng, đồng thời đưa ra thuật toán để có thể giải hai bài toán trên. Tương tự như [12], bài toán tìm siêu tối tiểu (hyper-minimizing) cho otomat khoảng, phương pháp làm giảm độ phức tạp của thuật toán đơn định và tối tiểu... là vấn đề thú vị có thể tiếp tục nghiên cứu. Ngoài ra, liệu ta có thể song song hoá thuật toán đơn định (có độ phức tạp cao) như trong [6], hay sử dụng các kỹ thuật đặc biệt trong lập trình [6] để có thể tối ưu về mặt thời gian thực hiện thuật toán.

TÀI LIỆU THAM KHẢO

[1] Rajeev. Alur and David L. Dill, A theory of timed automata, *Proceedings of 17th International Colloquium on Automata, Languages, and Programming*, and *Proceedings of the REX*

workshop "Realtime: theory in practice", 1991; *Theoretical Computer Science* **126** (1994), 183–235.

[2] Andrei Păun, Mihaela Păun, Alfonso Rodríguez-Patón, *Hopcroft's Minimization Technique: Queues or Stacks?*, CIAA 2008, LNCS 5148, Springer-Verlag Berlin Heidelberg, 2008, 78–91.

[3] Bui Vu Anh, Nondeterministic duration automata in modeling priority network, *Proceedings of National Conference on Discovering Knowledge from Data*, Vietnam, August 5-6, 2009, 210156B00, 315–325.

[4] Bui Vu Anh, Duration automaton in scheduling programs for a cluster computer system, *Journal of Computer Science and Cybernetics* **27**(3) (2011), 218–228.

[5] Bui Vu Anh, Phan Trung Huy, Ứng dụng một tích khoảng mới trong bảo mật, *Journal of Computer Science and Cybernetics* **29**(2) (2013), 149–158.

[6] Ambuj Tewari, Utkarsh Srivastava, and P. Gupta,*A Parallel DFA Minimization Algorithm*, LNCS 2552, Springer-Verlag Berlin Heidelberg, 2002, 34–40.

[7] Deepak D'Souza, P.S. Thiagarajan, "Distributed Interval Automata: a subclass of Time Automata", Internal Report, Theoretical Computer Science TCS-98-3 (1998).

[8] J.E. Hopcroft, An n log n algorithm for minimizing states in a finite automaton, *Theory of Machines and Computations* **35** (1971), 189–196 (Academic Press).

[9] J.E. Hopcroft, J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, 1979.

[10] Dang Van Hung, Bui Vu Anh, Model checking component based systems with black-box testing, *Proceeding of 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications* (RTCSA'05), United Nation University, Macao (ISBNĨSSN: 1533-2306, 0-7695-2346-3), 2005, 76–79.

[11] Nancy Lynch and Mark Tuttle, An Introduction to Input/Output automata, *CWI-Quarterly* **2**(3) (1989), 219–246.

[12] Markus Holzer, Andreas Maletti, An n log n algorithm for hyper-minimizing a (minimized) deterministic automaton, *Theoretical Computer Science* **411** (2010) 3404–3413,

Ngày nhận bài 17 - 12 - 2013
Nhận lại sau sửa ngày 01 - 04 - 2014