Sinh ngữ cảnh giả định bằng thuật toán L\*

1. Thuật toán L\*

**Thuật toán 1**: Thuật toán học L\*

**Input**: U, Σ: Với U là một ngôn ngữ chưa biết, Σ là bảng chữ cái

**Output**: M: Với M là một DFA sao cho M là một automata đơn định nhỏ nhất tương ứng với U và L(M) = U

1: Initially, S = {λ},E = {λ}

2: loop

3: update T using membership queries

4: while (S,E,T) is not closed do

5: add sa to S to make S closed, where s ∈ S and a ∈ Σ

6: update T using membership queries

7: end while

8: construct a candidate DFA M from the closed (S,E,T)

9: present an equivalence query: L(M) = U?

10: if M is correct then

11: return M

12: else

13: add e ∈ Σ∗ that witnesses the counterexample cex to E

14: end if

15: end loop

Phương pháp sử dụng thuật toán học L\* được đề xuất bởi Angluin [9] và sau đó được cải tiến bởi Rivest and Schapire [17]. Trong nội dung của luận văn này, thuật toán mà chúng ta đề cập chính là phiên bản L\* đã được cải tiến. Thuật toán L\* học một ngôn ngữ chưa biết và đưa ra một DFA chấp nhận ngôn ngữ đó.

Với Σ là tập các chữ cái, U là một ngôn ngữ chưa biết trên Σ. L\* sẽ đưa ra một DFA M sao cho M là DFA đơn định nhỏ nhất và L(M) = U. Để học U, L\* cần tương tác với một giáo viên (Minimally Adequate Teacher). Người giáo viên này phải trả lời chính xác hai loại câu hỏi. Loại câu hỏi thứ nhất là câu hỏi dạng truy vấn thành viên, nội dung câu hỏi là một chuỗi σ ∈ Σ\* có thuộc U hay không. Câu trả lời của giáo viên là True nếu σ ∈ U và False nếu σ ∉ U. Loại câu hỏi thứ hai là một phỏng đoán, một DFA M có ngôn ngữ là L(M), được L\* tin tưởng là giống hệt với U, L\* sẽ hỏi giáo viên xem là L(M) có bằng U hay không. Câu trả lời của giáo viên sẽ là True nếu L(M) = U, ngược lại giáo viên sẽ trả về một phản ví dụ là một chuỗi σ đối xứng giữa L(M) và U. Hình xxx dưới đây mô tả sự tương tác giữa L\* và giáo viên.

True/False

L\*

Chuỗi σ ∈ Σ\* ?

DFA C1, L(C1) = U?

False + Phản ví dụ/True

DFA đơn định nhỏ nhất

Hình xxx: Mô hình sự tương tác giữa L\* và giáo viên

Đi một cách chi tiết vào thuật toán, L\* sẽ cải tiến bảng T, trong đó T là một bảng để ghi lại xem chuỗi s trong Σ\* có thuộc U hay không. Để làm điều này, L\* sẽ sử dụng dạng câu hỏi thứ nhất (truy vấn thành viên) đối với giáo viên.

2. Sinh ngữ cảnh dựa trên thuật toán học L\*

Để sinh ra được ngữ cảnh hợp lí, thuật toán L\* áp dụng quy tắc ghép nối theo cách lặp đi lặp lại được minh họa trong hình 3.5. Ở mỗi bước lặp i, một ứng viên ngữ cảnh giả định Ai được sinh ra ra dựa trên một số tri thức về hệ thống và kết quả của bước lặp trước. 2 bước của quy tắc ghép nối được áp dụng. Bước 1 kiểm tra xem M1 có thỏa mãn p trong môi trường đảm bảo Ai hay không bằng cách tính toán biểu thức <Ai> M1 <p>. Nếu kết quả là false, điều này có nghĩa ứng viên ngữ cảnh quá yếu (Ai không hạn chế môi trường để p được thỏa mãn). Do đó, ứng viên Ai phải được làm mạnh hơn, điều này có nghĩa phải loại bỏ các hành vi trong Ai, phản ví dụ cex sẽ được đưa ra. Trong ngữ cảnh của ứng viên giả định tiếp theo là Ai+1, M1 nên ít nhất là không vi phạm các hành vi được phản ánh bởi phản ví dụ. Ngược lại nếu kết quả là true, điều này có nghĩa là Ai là đủ mạnh để M1 thỏa mãn thuộc tính p. Bước thứ 2 được áp dụng để kiểm tra nếu mô hình thành phần M2 thỏa mã Ai theo công thức <true> M2 <Ai>. Nếu bước này kết quả trả về là true, thì hệ thống ghép nối M1||M2 thỏa mãn thuộc tính p (M1||M2 |= p) và thuật toán kết thúc. Ngược lịa, nếu kết quả là false, thêm nữa cần có sự phân tích để kiểm tra xem thuộc tính p có thật sự bị vi phạm trong M1||M2 hay ứng viên giả định Ai là quá mạnh để M2 thỏa mãn. Phân tích như vậy là dựa vào phản ví dụ cex được trả lại bởi bước 2. Bước này sẽ kiểm tra xem phản ví dụ cex có thuộc về ngôn ngữ chưa biết U = L(Aw) (cex ϵ L(Aw ?). phân tích này kiểm xe thuộc tính p bị vi phạm bởi M1 trong ngữ cảnh của phản ví dụ cex bằng việc kiểm tra công thức [cex] || M1 không thõa mãn p, trong đó [cex] là 1 LTS được định nghĩa như sau:

LTS [cex] = <Q, α[cex], δ, q0> và phản ví dụ cex = a1a2…ak. LTS [cex] được tạo từ phản ví dụ cex như sau:

[cex] = <Q, α[cex], δ, q0>. Trong đó

* Q = {q0, q1, …, qk},
* α[cex] = {a1, a2, …, ak},
* δ = {(qi-1, ai, qi) | 1 ≤ i ≤ k}, và
* q0 = q0

Hình 3.4 minh họa LTS [cex] được tạo từ phản ví dụ cex:

**…**

Hình 3.4. LTS [cex] được tạo từ phản ví dụ cex

Nếu thuộc tính p không được giữa trong hệ thống ghép nối [cex] || M1 ([cex] || M1 ⊭p), điều này có nghãi là hệ thống ghép nối M1||M2 không thỏa mãn thuộc tính p (M1||M2 ⊭ p). Ngược lại, Ai là quá mạnh để được thỏa mãn bởi M2. Ứng viên giả định Ai cần bị làm yếu đi trong vòng lặp tiếp thoe. Kết quả của việc làm yếu đi này ít nhất hành vi thể hiện trong phản ví dụ cex sẽ được chấp nhận bởi ứng viên giả định Ai+1. Ứng viên giả định mới có thể tất nhiên quá yếu và do đó quá trình này được lặp lại.

Một câu hỏi quan trọng trong phương thức này là thuật toans học L\* làm việc như thế nào. Câu hỏi tương tự là sinh ứng viên giả định Ai như thế nào tại mỗi bước lặp i trong hình 3.5. Trong phương thức đảm bảo giả định được đề xuất trong [10], L\* học ngôn ngữ của giả định yếu nhất Aw. Ý nghĩa của điều này là L\* học ngôn ngữ chưa biết U = L(Aw) thông qua ngôn ngữ ∑ = α Aw = (αM1 ∪ α) αM2. Phương pháp sử dụng các ứng viên được sinh ra bởi thuật toán học L\* như các ứng viên ngữ cảnh Ai cho luật đảm bảo giả định (luật ghép nối). Để đưa ra mỗi ứng viên giả định Ai, đầu tiên L\* sinh ra một ứng viên DFA Mi dựa trên bảng quan sát đóng S, E, T, sau đó DFA Mi sẽ được chuyển trang LTS an toàn như các ứng viên giả định Ai bằng cách áp dụng định nghĩa được trình bày ở chương 2. Để học Aw, chúng ta cần cung cấp 1 Teacher, Teacher này có khả năng trả lời 2 loại câu hỏi mà L\* yêu cầu. Loại câu hỏi đầu tiên là 1 truy vấn thành viên, bao gồm 1 chuỗi σ thuộc ∑\*. Câu trả lời là true nếu σ thuộc U, và false nếu σ không thuộc U. Loại câu hỏi thứ 2 là một phỏng đoán, 1 ứng viên DFA Mi sở hữu ngôn ngữ thuật toán được tin tưởng giống với U. Câu trả lời là true nếu L(Mi) = U, ngược lại Teacher trả lại 1 phản ví dụ, phản ví dụ là 1 chuỗi σ đối xứng khác nhau giữa L(Mi) và U. Phương pháp tiếp cận này sử dụng kiểm chứng mô hình để cài đặt 1 Teacher.

Với kiểu câu hỏi đầu tiên, để trả lời một truy ván thành viên cho chuỗi σ = a1a2…an xem trong Σ\* = L(Aw), Teacher minh họa truy vấn trong hệ thống ghép nối M1||perr. Với chuỗi σ, đầu tiên Teacher xây dựng một LTS [σ] = <Q, α[σ], δ, q0>, trong đó Q = {q0, q1, …, qn}, α[σ] = ∑, δ = {(qi-1, ai, qi) | 1 ≤ i ≤ n}, và q0 = q0. Teacher sau đó kiểm tra công thức <[σ]> M1 <p> bằng cách tính toán hệ thống ghép nối [σ]||M1||perr. Nếu không có trạng thái nào trong hệ thống ghép nối đến được trạng thái lỗi π (tức là công thức trả về True), điều đó có nghĩa là σ thuộc L(AW). Trong trường hợp này, Teacher trả về true bởi vì M1 không vi phạm thuộc tính p trong ngữ cảnh của σ. Ngược lại câu trả lời để truy vấn thành viên là false.

Với kiểu câu hỏi thứ hai, với mỗi DFA Mi được sinh ra bởi L\* từ bảng quan sát S, E, T tại mỗi vòng lặp i, Teacher phải kiểm tra xem DFA Mi có phải là môt DFA ứng viên của vòng lặp i hay không. (L(Mi) = L(AW)?). Cho mục tiêu này, Teacher đầu tiên chuyển DFA Mi sang LTS an toàn Ai. Sau đó sử dụng LTS an toàn Ai làm ứng viên giả định cho quy tắc ghép nối. Teacher áp dụng hai bước của quy tắc ghép nối và phân tích phản ví dụ để trả lời phỏng đoán như sau:

Bước 1 được minh họa như trong hình 3.5, Teacher sẽ kiểm tra biểu thức <Ai> M <p> bằng cách ghép nối Ai||M||perr. Nếu tồn tại một dẫn xuất nào đó chứa trạng thái lỗi π thì biểu thức <Ai> M <p> trả về false. Nên teacher trả về kết quả là false và 1 phản ví dụ cex. Teacher trả về cho L\* là phỏng đoán Ai không chính xác và cung cấp phản ví dụ cex phù hợp hơn. Ngược lại, Teacher trả lời là True và chuyển sang bước 2.

Bước 2 Teacher kiểm tra biểu thức <true> M2 <Ai­> như trong hình 3.5 Nếu biểu thức này trả về kết quả là True. Framework của chúng ta kết thúc việc kiểm chứng bởi vì dựa trên luật ghép nối, M1||M2 ||= p. Ngược lại bước này trả về 1 phản ví dụ cex. Teacher sẽ tiến hành phân tích để xác định xem p có thật sự vi phạm M1 || M2 hay Ai có quả quá mạnh để M2­ thỏa mãn. Việc phân tích phản ví dụ được thực hiện bởi Teacher bằng cách tương tự các sử dụng để trả lời truy vấn thành viên. Với cex là phản ví dụ được trả lại bởi bước 2. Đầu tiên, Teacher sẽ tạo 1 LTS an toàn [cex ↑ Σ] từ phản ví dụ cex được minh họa trên hình 3.4. Sau đó, Teacher kiểm tra biểu thức <[cex ↑ Σ]> M1 <p> bằng cách ghép nối <[cex ↑ Σ] || M1 || perr. Nếu không tồn tại 1 dẫn xuất nào đó chứa trạng thái lỗi π thì chứng tỏ M1||M2 không thỏa mãn p. Ngược lại, Ai quá mạnh để M2 thỏa mãn trong ngữ cảnh của cex. cex ↑ Σ được trả lại như một phản ví dụ cho phỏng đoán Ai



Hình 3.5. Mô hình sinh ngữ cảnh giả định dựa trên thuật toán học L\*

3. Ví dụ minh họa việc sinh ngữ cảnh sử dụng thuật toán học L\*

Giả sử chúng ta có 3 LTS như trên hình dưới đây

in

send

ack

in

Hình 1. LTS Input (M1)

in

out

out

in

Hình 2. LTS Order

send

out

ack

Hình 3. LTS Output (M2)

Nhiệm vụ của chúng ta là kiểm tra xem Input || Output ⊨ Perr hay không

Gọi A là LTS ngữ cảnh giả định với Σ là tập tập các hành vi. Ta có Σ = (αM1 hợp αP) ∩ αM2 = {send, out, ack}

Bước 1. Thực hiện ghép nối LTS Output và Perr ta được

in

out

out

send

in

out

in

out

ack

ack

out

in

send

out

Hình 3 : Mô hình ghép nối Input và Order

Xây dựng bảng quan sát (S, E, T), kết quả từ cột E có được là nhờ các truy vấn thành viên từ L\*, teacher sẽ kiểm tra trong mô hình ghép nối Input và Order ở hình 3 và đưa ra kết quả.

S

S. Σ

|  |  |
| --- | --- |
|  | E |
| Table T | λ |
| λ | True |
| ack | True |
| out | False |
| send | True |

Xét 1 truy vấn thành viên σ = ack, teacher sẽ minh họa σ thành 1 LTS A có dạng

Ta nhận thấy bảng quan sát này chưa đóng, đưa out lên trên tập S

S

S. Σ

|  |  |
| --- | --- |
|  | E |
| T | λ |
| λ | True |
| out | False |
| ack | True |
| out | False |
| send | True |
| out, ack | False |
| out, out | False |
| out, send | False |

Từ bảng quan sát trên LTS tương ứng được sinh ra là A1 có dạng

ack

send

Hình 4. Mô hình LTS A1 ra từ bảng quan sát

Ta tiến hành kiểm tra tính đúng đắn của biểu thức <A1> M1 <P>.

Thực hiện việc ghép nối A1, M1, và P ta được , ta thấy A1||Input ≡ Input

out

in

send

out

in

out

in

out

ack

ack

send

out

in

out

Hình 7. Mô hình ghép nối A1||Input||Order

Chúng ta thấy tồn tại dẫn xuất đến trạng thái lỗi π là <in, send, ack, out>. Chúng ta dùng luôn dẫn xuất này làm phản ví dụ. Ta có, cex = <in, send, ack, out>

Phản ví dụ mà teacher trả lại cho L\* có dạng cex↑Σ = <send, ack>

Đưa send vào tập tiền tố send và ack vào tập hậu tố ta được bảng quan sát như sau:

S

S. Σ

|  |  |
| --- | --- |
|  | E |
| T | λ ack |
| λ | True True |
| out | False False |
| send | True False |
| ack | True True |
| out | False False |
| send | True False |
| out, ack | False False |
| out, out | False False |
| out, send | False False |
| send, ack | False False |
| send, out | True True |
| send, send | True True |

Bảng này đã đóng, DFA A2 sinh ra sẽ có dạng

[9] D. Angluin: "Learning Regular Sets from Queries and Counterexamples", Information and Computation, vol. 75, no. 2, pp. 87-106 (Nov. 1987).

[17] R. L. Rivest and R. E. Schapire: \Inference of Finite Automata using Homing Sequences", Information and Computation, vol. 103, no. 2, pp. 299-347 (Apr. 1993).