Lý thuyết tính toán

Lớp: 57TH3

Báo cáo: Ôtômat đẩy xuống

Nhóm:

Đỗ Minh Hiếu

Lê Thế Anh Nhân

Nguyễn Quý Hiếu

Nguyễn Mạnh Cường

Hà Mạnh Tuyên

Mục lục

[1. Ôtômat đẩy xuống 3](#_Toc495677134)

[1.1. Định nghĩa hình thức 3](#_Toc495677135)

[1.2. Ví dụ 5](#_Toc495677136)

[1.3. Sự tương đương với văn phạm phi ngữ cảnh 14](#_Toc495677137)

[1.3.1. Chứng minh bổ đề 1 15](#_Toc495677138)

[1.3.2. Chứng minh bổ đề 2 22](#_Toc495677139)

[2. Ngôn ngữ không phi ngữ cảnh 24](#_Toc495677140)

[2.1. Bổ đề bơm cho ngôn ngữ phi ngữ cảnh 25](#_Toc495677141)

[2.2. Sử dụng bổ đề Bơm để chứng minh một ngôn ngữ A là không phi ngữ cảnh 25](#_Toc495677142)

Đỗ Minh Hiếu

# 1. Ôtômat đẩy xuống

## 1.1. Định nghĩa hình thức

1. Khái niệm

- Ôtômat đẩy xuống (PDA) giống với Ôtômat không đơn định (NFA) ngoại trừ việc (PDA) có thêm một thành phần nữa là ngăn xếp

- Ngăn xếp cung cấp một bộ nhớ bên cạnh một lượng hữu hạn có sẵn trong sự điều khiển

- Ngăn xếp đoán nhận một số ngôn ngữ không chính quy

Mô hình tổng quát về Ôtômat đẩy xuống

2. Biểu diễn PDA

- PDA khi được biểu diễn sẽ gồm có các trạng thái dưới dạng hình tròn, các định thức để thể hiện cho các bước chuyển trạng thái hoặc các thao tác xử lý xâu đầu vào tại từng trạng thái.

Bộ kí tự vào

Ngăn xếp

…..

z

y

x

b

b

a

a

Bộ điều khiển trạng thái

|  |
| --- |
|  |
|  |
|  |
|  |

Biểu diễn hình học của PDA

Trong đó :

* A,B là các trạng thái của Ôtômat
* a là ký tự vào
* b là ký tự nằm ở đỉnh ngăn xếp, ký tự này sẽ được lấy ra (pop)
* c là ký tự được đẩy (push) vào trong ngăn xếp
* a, b, c đều có thể nhận ký tự Ɛ
* a,b c là định thức

Khi biểu diễn PDA ta có 6 trường hợp đối với định thức :

* TH 1: Ɛ, Ɛ $ định thức này thể hiện là ta không đọc kí tự nào từ xâu vào, không lấy kí tự nào từ ngăn xếp ra mà chỉ truyền vào kí tự đáy $. Ta dùng khi chuyển từ trạng thái bắt đầu sang trạng thái đầu tiên
* TH 2: a, Ɛ a định thức này thể hiện việc ta đọc kí tự từ xâu đầu vào, không lấy gì từ ngăn xếp ra và truyền kí tự vào trong ngăn xếp. Trường hợp này ta dùng khi a là kí tự đầu tiên của xâu kí tự và đọc liên tiếp các kí tự a đằng sau kí tự a đầu xâu
* TH 3: a,b Ɛ định thức này thể hiện ta đọc kí tự a từ xâu đầu vào, lấy kí tự b từ ngăn xếp ra và không truyền gì vào trong ngăn xếp. Ta dùng khi dãy kí tự đầu tiên của xâu đầu vào được đọc hết, gặp kí tự khác trong xâu và thực hiện việc đọc các kí tự còn lại giống với kí tự này
* TH 4: Ɛ, Ɛ Ɛ định thức này chỉ ra rằng ta không làm một việc gì cả, không đọc không lấy và không ghi. Ta dùng định thức này khi muốn thể hiện việc ghép nối giữa hai Ôtômat

B

A

a,b c

* TH 5: Ɛ, $ Ɛ định thức này thể hiện ta không đọc gì, lấy ra kí tự đáy hoặc đỉnh cảu ngăn xếp ra, không truyền vào trong ngăn xếp. Ta dùng định thức này khi xâu đàu vào được đọc hết, lấy hết kí tự ra khỏi ngăn xếp và chyển sang trạng thái kết thúc

3. Định nghĩa hình thức tổng quát

* Ôtômat đẩy xuống bộ 6 (6 chiều)

M = (Q, , Ɛ, δ, q0,F)

* Trong đó:
  + Q: Tập trạng thái (hữu hạn)
  + Ɛ: Bộ chữ đầu vào, tập hữu hạn các kí tự
  + : Bộ chữ của ngăn xếp, tập hữu hạn các kí tự ( khác với NFA như đã nói ở trên )
  + δ: Hàm dịch chuyển
  + Q0: Trạng thái ban đầu
  + F: Là tập các trạng thái kết thúc (F ⊆ Q)

## 1.2. Ví dụ

* Ví dụ

- Xét ngôn ngữ A = {0n1n| nta lấy cụ thể là xâu 000111

- Vẽ Ôtômat:

* Bước 1: ta quy ước $ là kí tự đỉnh hay đáy ngăn xếp để biết được trạng thái của ngăn xếp có rỗng hay không, vẽ trạng thái bắt đầu và trạng thái q0, lúc này trong ngăn xếp sẽ có kí tự $ và sử dụng định thức ở TH 1 như đã nêu trên.

S

Ɛ, Ɛ $

q0

Start

$

* Bước 2: ta bắt đầu đọc kí tự 0 của xâu đầu vào và đọc hai kí tự 0 còn lại,ta sử dụng định thức ở TH 2 và vì đọc liên tiếp các kí tự không cho nên tại trạng thái q0 ta sẽ sử dụng một vòng lặp đối với việc đọc 0 và ghi 0. Lúc này ngăn xếp sẽ được ghi các số 0 vào lần lượt.

0

Ɛ, Ɛ $

Start

q0

S

0

0

0, Ɛ 0

$

* Bước 3: sau khi đọc hết dãy 0 trong xâu đầu vào ta sẽ đọc sang dãy kí tự 1. Nhưng trước đó ta phải làm một bước rất quan trọng đó là thực hiện việc ghép nối hai trạng thái chuển từ 0 sang một hay nói một cách chính xác hơn là việc thực hiện ghép nối hai Ôtômat
  + Xét ngôn ngữ A có dạng 0n1n ta coi 0n là một Ôtômat riêng, 1n là một Ôtômat riêng và hai Ôtômat này hợp lại sẽ được A. Để biểu diễn được A ta sẽ phải sử dụng phép ghép nối để hợp hai Ôtômat này lại. Như ở các bài trước để thể hiện phép ghép nối Ôtômat ta có cách biểu diễn tổng quát sau:

Ɛ

* + Cho nên ta sẽ sử dụng định thức Ɛ, Ɛ Ɛ ở TH 4 đã nói trên gắn cho bước chuyển trạng thái từ q0 sang q1 để thể hiện việc ghép nối hai Ôtômat.

Ɛ

q1

Start

q0

S

Ɛ, Ɛ Ɛ

Ɛ, Ɛ $

0, Ɛ 0

* Bước 4: khi kí tự 1 đầu tiên được đọc, kí tự này sẽ được so sánh với kí tự ở đỉnh của ngăn xếp trong trường hợp này là 0, nhận thấy 0 và 1 khác nhau nên 0 sẽ bị lấy ra và cùng với việc lấy 0 ra ngăn xếp cũng không được đẩy thêm kí tự nào vào. Thao tác này sẽ được lặp lại với mỗi khi đọc kí tự 1 do vậy ở trạng thái q1 phải có vòng lặp đối với định thức 1,0 Ɛ (dạng TH 3 đã nêu ở trên ).

Start

Ɛ, Ɛ Ɛ

Ɛ, Ɛ $

S

q1

q0

0, Ɛ 0

1, 0 Ɛ

Đọc kí tự 1 thứ hai

Đọc kí tự 1 thứ ba

Đọc kí tự 1 thứ nhất

Lấy 0 ra

Lấy 0 ra

Lấy 0 ra

…

…

…

$

* Bước 5: sau khi dãy 1 được đọc hết , dãy 0 cũng được lấy hết ra khỏi ngăn xếp thì Ôtômat sẽ không còn kí tự nào để đọc, trong ngăn xếp lúc này chỉ còn kí tự đỉnh hay đáy. Lúc này ta thực hiện chuyển trạng thái từ q1 sang trạng thái kết thúc, trong bước chuyển này ta sẽ dùng định thức Ɛ, $ Ɛ ở TH 5. Sau khi lấy ra kí tự đáy ngăn xếp rỗng, xâu đầu vào cũng được đọc hết,kết thúc việc biểu diễn Ôtômat.

Ɛ, Ɛ Ɛ

Ɛ, Ɛ $

Start

E

q1

q2

S

0, Ɛ 0

1, 0 Ɛ

* Các bộ của Ôtômat:
* Q = { S , q1 , q2 , E }- tập trạng thái của A
* = { 0 , 1 , Ɛ }- Bộ kí tự đầu vào
* = { 0 , $ , Ɛ }- Bộ kí tự sẽ được ghi trong ngăn xếp
* F = { E }- trạng thái kết thúc
* Q0 = { S } – trạng thái đầu tiên
* δ :

{ q1, }

{ q2, }

{ q1, 0}

{ q1, $}

$

$

$

{ E, }

* Giải thích bảng dịch chuyển trạng thái δ:
* Xét ở trạng thái S ta không đọc gì nên sẽ chọn khoảng thứ 3 của bảng là , ta cũng không lấy gì nên sẽ chọn điền ở vị trí khoảng thứ 3 và điền vào {q1, } để chuyển sang trạng thái q1
* Tại trạng thái q1 ta thực hiện đọc 0, không lấy gì ra khỏi ngăn xếp và truyền 0 vào trong ngăn xếp nên ta sẽ điền vào khoảng thứ nhất trong bảng là 0, ứng với dòng q1 tại vị trí và viết { q1,0 }
* Sau khi dãy 0 được đọc xong đồng nghĩa với việc Ôtômat 0n kết thúc và chuyển sang Ôtômat 1n với dãy 1 được đọc, trạng thái q1 sẽ chuyển sang trạng thái q2 với phép ghép nối. Do vậy ở đây tại dòng q1 khoảng thứ 3 của bảng và ứng với vị trí ta điền {q2, } để thể hiện việc ghép nối
* Tại trạng thái q2 ta đọc dãy kí tự 1 lấy 0 ra khỏi ngăn xếp và không điền gì vào trong ngăn xếp, ứng với dòng trạng thái q2 tại khoảng 1 trong bảng và tại vị trí ta điền {q2, }
* Sau khi việc đọc xâu kết thúc , ngăn xếp chỉ còn lại $ ta không còn gì để đọc, lấy $ ra khỏi ngăn xếp và không điền gì để chuyển từ q2 sang trạng thái kết thúc E. Do đó vẫn dòng trạng thái q2 trong bảng tại khoảng thứ 3 tại vị trí ta điền {E,}

Lê Thế Anh Nhân

* Ví dụ 1:

Cho xâu {0n1n|n0} ta có:

Q={A,B,C,D} : Tập trạng thái

= {0,1, ε } : bộ chữ đầu vào  
={0,$, ε } : bộ chữ của ngăn xếp

F = { D}

Ta có bảng chuyển trạng thái như sau:

* (A, ε, ε) → (B,$) : ở trạng thái đầu tiên của ngăn xếp,để kiểm tra xem trong ngăn xếp lúc đó có phần tử nào hay chưa, ta không đọc gì,không lấy gì và đẩy $ vào làm đáy của ngăn xếp và chuyển sang trạng thái B
* (B, 0, ε) → (B,0) : ở trạng thái B,ta lặp các bước sau : đọc 0,không lấy gì và đẩy 0 vào trong ngăn xếp
* (B,1,0) → (C, ε) : ở trạng thái B, ta thực hiện đọc 1,lấy 0 ra khỏi ngăn xếp và không đẩy gì vào ngăn xếp, chuyển trạng thái sang C.
* (C,1,0) → (C, ε) : Ở trạng thái C, ta sử dụng vòng lặp để thực hiện các bước đọc 1, ghi 0 và không đẩy gì vào trong ngăn xếp.
* (C, ε,$) → (D, ε) : từ trạng thái C ta chuyển sang trạng thái D, không đọc gì,lấy đáy ra khỏi ngăn xếp và không đẩy gì vào ngăn xếp. → kết thúc xâu.

Bảng chuyển trạng thái δ:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | | | 1 | | | ε | | |
|  | 0 | $ | ε | 0 | $ | ε | 0 | $ | ε |
| A |  |  |  |  |  |  |  |  | (B,$) |
| B |  |  | (B,0) | (C, ε) |  |  |  |  |  |
| C |  |  |  | (C, ε) |  |  |  | (D, ε) |  |
| D |  |  |  |  |  |  |  |  |  |

Vẽ Ôtômat

ε,$ → ε

1,0 → ε

1,0 → ε

0, ε → 0

ε, ε → $

* Ví dụ 2:

{ aibjck | i,j,k và i=j hoặc i=k}

Q={A,B,C,D,E,F,G} : Tập trạng thái

= {a,b,c, ε } : bộ chữ đầu vào  
={a,$, ε } : bộ chữ của ngăn xếp

F = {D,G }

* TH1 : i= j ta có aibick
* (A, ε, ε) → (B,$): tại trạng thái ban đầu để kiểm tra ngăn xếp có phần tử nào chưa ta làm như sau không đọc gì,không lấy gì và đẩy $ vào trong ngăn xếp ($ là đáy ngăn xếp) rồi chuyển trạng thái sang B
* (B,a, ε) →(B,a) : tại B chúng ta lặp lại các bước sau đọc a,không lấy gì ra khỏi ngăn xếp và đẩy a vào trong ngăn xếp.
* (B, ε, ε) → (C, ε) : đây là phép ghép tiếp để kết nối 2 otomat bằng cách không đọc gì,không lấy gì từ ngăn xếp,không đẩy gì vào ngăn xếp và chuyển trạng thái sang C
* (C,b,a) → (C, ε) : vẫn ở trạng thái C,ta lặp lại các bước sau đọc b,lấy a ra khỏi ngăn xếp và không đẩy gì vào trong ngăn xếp.
* (C, ε,$) → (D, ε) : từ trạng thái C chuyển sang trạng thái D ta không đọc gì,đẩy đáy ra khỏi ngăn xếp và không đẩy gì vào trong ngăn xếp.
* (D,c, ε) → (D, ε) : ở trạng thái D chúng ta chỉ đọc c mà lấy hay đẩy gì vào ngăn xếp vì số mũ của c khác a và b. → Kết thúc chuỗi
* TH2:i=k ta có aibjci
* (A, ε, ε) → (B,$): tại trạng thái ban đầu để kiểm tra ngăn xếp có phần tử nào chưa ta làm như sau không đọc gì,không lấy gì và đẩy $ vào trong ngăn xếp ($ là đáy ngăn xếp) rồi chuyển trạng thái sang B
* (B,a, ε) →(B,a) : tại B chúng ta lặp lại các bước sau đọc a,không lấy gì ra khỏi ngăn xếp và đẩy a vào trong ngăn xếp.
* (B, ε, ε) → (E, ε) : đây là phép ghép tiếp để kết nối B với E khi không đọc gì,không lấy gì và không đẩy gì vào trong ngăn xếp.
* (E,b, ε) → (E, ε) : tại E chúng ta chỉ đọc b mà không lấy hay đẩy gì vào ngăn xếp vì tại trường hợp này số mũ của b khác a và c.
* (E, ε, ε) → (F, ε) : chúng ta dùng phép ghép tiếp để kết nối E và F
* (F,c,a) → (F, ε) : Tại F chúng ta lăp lại các bước sau đọc c,lấy a ra khỏi ngăn xếp và không đẩy gì vào trong ngăn xếp.
* (F, ε,$) → (G, ε) : Tại F chúng ta không đọc gì, lấy đáy của ngăn xếp ra ngoài và không ghi gì vào ngăn xếp rồi chuyển đến trạng thái kết thúc. → Kết thúc xâu
* Bảng chuyển trạng thái δ:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | | | b | | | c | | | ε | | |
|  | a | $ | ε | a | $ | ε | a | $ | ε | a | $ | ε |
| A |  |  |  |  |  |  |  |  |  |  |  | (B,$) |
| B |  |  | (B,a) |  |  |  |  |  |  |  |  | (C,ε)  (E,ε) |
| C |  |  |  | (C,ε) |  |  |  |  |  |  | (D,ε) |  |
| D |  |  |  |  |  |  |  |  | (D,ε) |  |  |  |
| E |  |  |  |  |  | (E,ε) |  |  |  |  |  | (F,ε) |
| F |  |  |  |  |  |  | (F,ε) |  |  |  | (G,ε) |  |
| G |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

c,ε → ε

b,a → ε

a,ε → ε

ε, $→ ε

ε, ε→ ε

ε, ε→ $

ε, ε→ ε

ε, $→ ε

ε, ε→ ε

c, a → ε

b, ε → ε

Vì trạng thái A và B của 2 bước giống nhau nên ta có thể kết hợp chúng với nhau

* Ví dụ 3: PDA đoán nhận ngôn ngữ {wwR| w {0,1}\*} với wR là xâu w được viết theo thứ tự đảo ngược.

Ta lấy xâu w= 01 vậy xâu wR = 10,thực chất trong trường hợp này là 2 xâu nối tiếp nhau và là nghịch đảo của nhau

Q={A,B,C,D}

= {0,1, ε }   
={0,1,$, ε }

F = {A,D }

* Đầu tiên ta đang ở trạng thái bắt đầu A,để kiểm tra xem trong ngăn xếp lúc đó có phần tử nào không ta thực hiện phép đẩy $ vào ngăn xếp làm đáy ngăn xếp mà không thực hiện đọc và lấy gì từ ngăn xếp ra,chuyển sang trạng thái B.

|  |
| --- |
|  |
|  |
| $ |

* Tiếp theo tại trạng thái B,ta sử dụng vòng lặp để thực hiện hai thao tác sau: đọc 0,không lấy gì từ ngăn xếp ra,đẩy 0 vào trong ngăn xếp và đọc 1,không lấy gì từ ngăn xếp,đẩy 1 vào trong ngăn xếp,khi đó trong ngăn xếp sẽ được biểu diễn như sau

|  |
| --- |
|  |
|  |
| 1 |
| 0 |
| $ |

* Sau đó ta dùng phép ghép tiếp để kết nối hai ngăn xếp w và wR bằng cách không đọc gì,không lấy gì và không đẩy gì vào ngăn xếp rồi chuyển trạng thái sang C.
* Tiếp theo ở trạng thái C,t thực hiện vòng lặp để đọc 1 và 0 rồi lấy 1 và 0 ra khỏi ngăn xếp và không ghi gì vào ngăn xếp và trong ngăn xếp lúc đó chỉ còn đáy($).

|  |
| --- |
|  |
|  |
|  |
| $ |

pop

|  |
| --- |
|  |
|  |
| 1 |
| 0 |
| $ |

* Chúng ta chuyển từ trạng thái C sang trạng thái D bằng cách không đọc gì,không lấy gì và đẩy đáy ra khỏi ngăn xếp,kết thúc xâu tại D.
* Một trường hợp đặc biệt khi ngay từ trạng thái A xâu đọc một kí tự rỗng ε nên A sẽ là một trạng thái kết thúc.

ε, $ → ε

0,0→ ε

1,1 → ε

0, ε → 0

1,ε→ 1

ε, ε → ε

ε, ε → $

ε, $ → ε

0,0→ ε

1,1 → ε

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | | | | 1 | | | | ε | | | |
|  | 0 | 1 | $ | ε | 0 | 1 | $ | ε | 0 | 1 | $ | ε |
| A |  |  |  |  |  |  |  |  |  |  |  | (B,$) |
| B |  |  |  | (B,0) |  |  |  | (B,1) |  |  |  | (C,ε) |
| C | (C,ε) |  |  |  |  | (C,ε) |  |  |  |  | (D,ε) |  |
| D |  |  |  |  |  |  |  |  |  |  |  |  |

Nguyễn Quý Hiếu

# 1.3. Sự tương đương với văn phạm phi ngữ cảnh

Một ngôn ngữ phi ngữ cảnh cảnh là một ngôn ngữ được biểu diễn bởi một CFG nào đó. Ta có định lí: Một ngôn ngữ là phi ngữ cảnh nếu và chỉ nếu có một PDA nào đó đoán nhận nó.(Đây là định lí 2 chiều vì thế có thể chia thành 2 bổ đề)

* Bổ Đề 1: Nếu một ngôn ngữ là phi ngữ cảnh thì sẽ có một PDA đoán nhận nó.
* Bổ Đề 2: Nếu có một PDA đoán nhận một ngôn ngữ thì ngôn ngữ đó là ngôn ngữ phi ngữ cảnh.

## 1.3.1. Chứng minh bổ đề 1

Ta chứng minh bổ đề 1 bằng phương pháp xây dựng.

* Bước 1: Đặt kí hiệu $ và biến ban đầu vào trong ngăn xếp
* Bước 2: Lặp lại các bước sau:
  + Nếu đỉnh của ngăn xếp là 1 ký hiệu biến A → Chọn một quy tắc ứng với A và thay thế A bởi phần bên phải của quy tắc đó
  + Nếu đỉnh của ngăn xếp là 1 ký hiệu kết thúc a → Đọc ký hiệu tiếp theo từ dữ liệu vào và so sánh. Nếu giống nhau thì lặp lại, khác nhau thì bỏ qua nhánh này
  + Nếu đỉnh của ngăn xếp là kí hiệu $, chuyền vào trạng thái chấp thuận. Nếu tất cả các dữ liệu vào được đọc thì ngôn ngữ được chấp thuận.

\*Giải thích từng bước:

* Bước 1: Chúng ta bắt đầu bằng việc khởi tạo ngăn xếp chứa kí hiệu $ và S

start

Bước 2:

Đầu tiên vận dụng trường hợp (a) trong đó đỉnh ngăn xếp chứa 1 biến. Đặt | trong đó A là một quy tắc thuộc R }.

Thứ 2 chúng ta vận dụng trường hợp (b) trong đó đỉnh ngăn xếp chứa một kí hiệu kết thúc. Đặt .

Cuối cùng, chúng ta vận dụng trường hợp (c) trong đó kí hiệu đánh dấu ngăn xếp rỗng $ nằm trên đỉnh ngăn xếp. Đặt:

(

Sau khi khởi tạo ngăn xếp chứa kí hiệu $ và S, đỉnh ngăn xếp lúc này là một biến. Ta tìm tất cả các quy tắc ứng với S. Với mỗi quy tắc tìm được, ta lại có một chuyển đổi hàm thực hiện vòng lặp lấy S ra khỏi ngăn xếp và đẩy phần bên phải của quy tắc vào ngăn xếp. Sau đó ta lại làm tương tự với tất cả các biến có trong văn phạm phi ngữ cảnh.

Với mỗi ký tự kết thúc có trong văn phạm phi ngữ cảnh, ta lại có một chuyển đổi hàm thực hiện vòng lặp đọc ký tự đó từ dữ liệu vào và lấy cùng ký tự đó từ ngăn xếp.

Rồi cuối cùng, ta lấy đáy khỏi ngăn xếp và chuyển sang trạng thái chấp thuận.

for rule A

a,a for a terminal a

Chú ý: Để đưa nhiều ký hiệu vào ngăn xếp, ta không thể cùng lúc đưa toàn bộ mà ta cần những bước trung gian.

Ví dụ:

D

Để thay thế A trong ngăn xếp bằng BCD, đầu tiên ta lấy A ra khỏi ngăn xếp và đẩy D vào. Sau đó, ta lần lượt đẩy C và B vào trong ngăn xếp. Bởi vì ngăn xếp hoạt động theo cơ chế vào sau ra trước nên ta khi đẩy các ký tự ta cần thực hiện từ phải qua trái.

Ví dụ xây dựng PDA từ CFG:

S→0S1

S→

Đầu tiên ta thực hiện bước 1, đẩy $ và biến bắt đầu S vào trong ngăn xếp.

Sau đó, ta thực hiện bước 2, đưa các chuyển đổi hàm sau vào PDA:

Bởi vì ta không thể đẩy một lúc 0S1 vào ngăn xếp nên cần phải thêm các bước trung gian.

Cuối cùng ta lấy $ ra khỏi ngăn xếp và chuyển sang trạng thái chấp thuận.

Ta được PDA như hình dưới:

Bây giờ ta thử cho PDA vừa xây dựng được đọc một xâu văn phạm đã cho. Ví dụ xâu 000111

Đầu tiên, PDA sẽ chuyển từ trạng thái bắt đầu q0 đến q2 và đẩy $ và S vào trong ngăn xếp. Sau đó PDA sẽ đi theo vòng lặp từ q2 đến q3 đến q4 rồi lại q2 để đẩy xâu cần đọc vào trong ngăn xếp. Đồng thời PDA thực hiện vòng lặp tại q2 với để đọc chuỗi 000, dùng để chuyển sang đọc chuỗi 1 và dùng để đọc chuỗi 111.

Mô phỏng từng bước với ngăn xếp

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
| $ |

* Đẩy $ vào ngăn xếp

000111

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| S |
| $ |

* Đẩy S vào ngăn xếp

000111

|  |
| --- |
|  |
|  |
|  |
| 0 |
| S |
| 1 |
| $ |

* Thay thế S ở đỉnh ngăn xếp bằng 0S1

000111

* Đọc kí tự 0 đầu tiên từ xâu đầu vào và lấy ra kí tự 0 ở đỉnh ngăn xếp

000111

|  |
| --- |
|  |
|  |
|  |
|  |
| S |
| 1 |
| $ |

|  |
| --- |
|  |
|  |
| 0 |
| S |
| 1 |
| 1 |
| $ |

* Thay thế S ở đỉnh ngăn xếp bằng 0S1

000111

|  |
| --- |
|  |
|  |
|  |
| S |
| 1 |
| 1 |
| $ |

* Đọc kí tự 0 thứ 2 và lấy ra kí tự 0 ở đỉnh ngăn xếp

000111

* Thay thế S ở đỉnh ngăn xếp bằng 0S1

|  |
| --- |
|  |
| 0 |
| S |
| 1 |
| 1 |
| 1 |
| $ |

000111

|  |
| --- |
|  |
|  |
| S |
| 1 |
| 1 |
| 1 |
| $ |

* Đọc kí tự 0 thứ 3 và lấy ra kí tự 0 ở đỉnh ngăn xếp

000111

|  |
| --- |
|  |
|  |
|  |
| 1 |
| 1 |
| 1 |
| $ |

* Sử dụng để lấy S ra khỏi đỉnh ngăn xếp, kết thúc việc đọc chuỗi 000, chuyển sang đọc chuỗi 1

000111

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
| $ |

* Liên tiếp đọc 1 từ xâu và lấy 1 ra từ đỉnh ngăn xếp

000111

* Đọc hết xâu, ta lấy nốt $ ra khỏi ngăn xếp

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

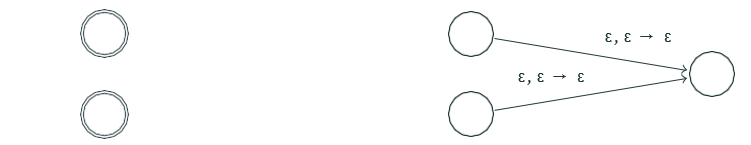
000111

Nguyễn Mạnh Cường

## 1.3.2. Chứng minh bổ đề 2

-Bước 1: Đơn giản hóa PDA sao cho có 3 đặc điểm sau

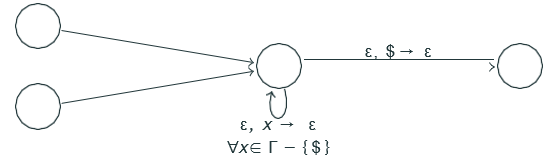
+ Có duy nhất một trạng thái chấp thuận \

 Vd:

Giải thích ý 1: Khi trong một PDA xuất hiện nhiều trạng thái kết thúc thì ta sẽ có bước chuyển các trạng thái kết thúc về thành 1 trạng thái kết thúc duy nhất.

+ Nó làm rỗng ngăn xếp trước khi chấp thuận

Giải thích ý 2: trong trường hợp chấp thuận 1 PDA mà trong khi ngăn xếp vẫn còn thì ta sẽ làm rỗng ngăn xếp



+ Không thực hiện đấy vào và lấy ra các ký hiện trong ngăn xếp cùng một lúc

a,b -> c

a, ->c

,b->

Với a là ký tự đầu vào

b là ký tự đi ra từ ngăn xếp

c là ký tự đi vào ngăn xếp

Giải thích ý 3: khi trong PDA xuất hiện những ký tự vừa lấy ra khỏi ngăn xếp và vừa đẩy vào ngăn xếp thì ta sẽ thêm một trạng thái trung gian để tách ký tự vào và ký tự ra khỏi ngăn xếp .

-Bước 2: Xây dựng CFG

Quy tắc xây dựng CFG

+Cho P= ( Q,{}) ta xây dựng CFG G với các biến là {p, q Q} biến ban đầu là

\*Chú ý : {p, q Q} là CFG sẽ có các biện với p và q là 2 trạng thái bất kỳ mà qua truy xuất từ p -> q

: là biến ban đầu từ trạng thái bắt đầu truy xuất đến trạng thái kết thúc

+ Với mỗi p, q, r, s Q, t và a, b , nếu (p, a, ) = (r, t) và (s, b, t) = (q,) thì ta đưa quy tắc vào trong G

\*Chú ý: a, b : dữ liệu đầu vào a, b có thể đọc

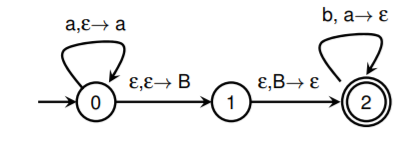
trong đó a là dữ liệu vào được đọc tại bước dịch chuyển đầu tiên, b là dữ liệu vào được đọc tại bước dịch chuyển cuối cùng, r là trạng thái sau p và s là trạng thái trước q.

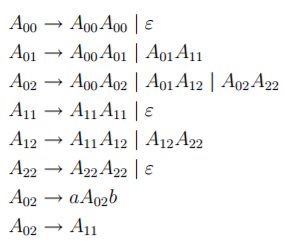
+ Với mỗi p,q,r ∈ Q đưa quy tắc A*pq* → A*pr* A*rq* vào trong G

\*Chú ý: với r là trạng thái khi ngăn xếp rỗng

+Với mỗi p ∈ Q đưa quy tắc A*pp* → ε vào trong G

VD:



Đối với cấu trúc chung sẽ có 9 biến cho p, q {0, 1, 2}. Tuy nhiên ta cần phải bao gồm những sao cho có một đường dẫn từ p đến q trong sơ đồ PDA. Điều này có nghĩa rằng chúng ta chỉ cần bao gồm các quy tắc liên quan đến , ,

Hà Mạnh Tuyên

# 2. Ngôn ngữ không phi ngữ cảnh

* Mọi ngôn ngữ phi ngữ cảnh đều có 1 giá trị đặc biệt được gọi là độ dài dẫn xuất
* Có một cách chứng minh ngôn ngữ không phi ngữ cảnh tương tự ngôn ngữ không phi ngữ cảnh

## 2.1. Bổ đề bơm cho ngôn ngữ phi ngữ cảnh

**Nếu A là một ngôn ngữ phi ngữ cảnh, thì tồn tại một số p sao cho nếu s là một xâu bất kỳ thuộc A có độ dài ít nhất là p, thì s có thể được chia ra làm 5 phần s=uvxyz thỏa mãn các điều kiện sau:**

* + - 1. **uv *i* xy*i* z ∈ A ∀ i ≥ 0**
      2. **|vy| > 0**
      3. **|vxy| ≤ p**

## 2.2. Sử dụng bổ đề Bơm để chứng minh một ngôn ngữ A là không phi ngữ cảnh

Ý TƯỞNG: (Chứng minh bằng phản chứng)

-Giả sử A là phi ngữ cảnh

-Nó có một độ dài dẫn xuất p

-Tất cả các xâu trong A có độ dài lớn hơn p (|s| ≥ p) có thể

chia làm 5 đoạn s = uvxyz

-Chọn 1 xâu như vậy trong A

-Chia nó làm 5 đoạn uvxyz

-Chỉ ra rằng uv *i* xy*i* z ∉ A bằng cách

-Xét tất cả các trường hợp mà s có thể chia thành 5 đoạn

-Chỉ ra rằng không có trường hợp nào thỏa mãn 3 điều kiện của bổ đề Bơm

→ Mâu thuẫn, do đó kết luận A không phải là phi ngữ cảnh

* Sử dụng bổ đề Bơm để chứng tỏ rằng B = {*anbncn*|n ≥ 0} là không phi ngữ cảnh
* Giả sử B là ngôn ngữ phi ngữ cảnh (CFL) → B có một độ dài dẫn xuất p
* Xâu chúng ta lựa chọn để chỉ ra phản chứng là: s = *ap bp cp*
* Xét các trường hợp có thể chia s thành 5 đoạn uvxyz

-v và y chỉ chứa 1 loại ký tự

-v và y chứa nhiều loại ký tự

* Ta có vxy ∈ anbncn , **|**xyz**| ≤** n nên xyz không thế đồng thời chứa cả ký hiệu a và c ( vì giữa a và c có n kí hiệu b ) vậy nên cũng không thể chứa đồng thời ký hiệu a và c
* Do **|**vy**| >** 0 và trong uvxyz chứa số kí hiệu a, b, c như nhau:
  + Nếu vy có chứa kí hiệu a (nên không thể chứa kí hiệu c) thì khi bơm chuỗi vy số kí hiệu c sẽ không đổi ( luôn là n) nhưng số kí hiệu a sẽ thay đổi. Tức là số kí hiệu a và số kí hiệu c luôn không bằng nhau
  + Nếu vy không chứa kí hiệu a thì khi bơm chuỗi vy số kí hiệu a không đổi nhưng số kí hiệu b hoặc c sẽ thay đổi.
* **VÍ DỤ:** Xét a4b4c4

B = **aaaabbbbcccc**

Giả sử B là văn phạm phi ngữ cảnh

Trường hợp 1:

Chọn **v=aa , y = c** thì suy ra **uv2xy2z = aaa|aaabbbbcc|ccc**

Rút gọn **uv2xy2z=** **a6b5c5**  ∉ B

Trường hợp 2:

Chọn **v=aabb , y = bc** thì suy ra **uv2xy2z = aaaabb|aabb|bbc|bcccc**

Ta thấy thứ tự của **uv2xy2z** không giống với thứ tự trong xâu B nên **uv2xy2z** cũng không thỏa mãn.

Theo điều kiện 3: **|vxy| ≤ p** ta có

TH1: **| aaabbbbcc | = 9 ≤ p=4 (** độ dài dẫn xuất của xâu B)(không thỏa mãn)

TH2: **| aabbbc | = 6 ≤ p=4** (không thỏa mãn)

* Có các mâu thuẫn nên giả thiết là sai → B là ngôn ngữ không phi ngữ cảnh