**Bài tập 1 – Máy Turing Và Hàm Đệ Quy Nguyên Thủy**

1. **Máy Turing**

## Bài 1:

Xây dựng máy turing M2 thực hiện phép trừ 1 của số nhị phân.

Lời giải:

M2 = (K, ∑, δ, s)

Trong đó:

K = {s, q, “h”}, ∑ = {0 ,1, □, ►},

s là trạng thái xuất phát.

h là trạng thái dừng.

Bảng hàm chuyển δ:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | q | σ | δ(q, σ) = (p, ρ, {→ | ← | -}) |
| 1 | s | ► | (s, ►, →) |
| 2 | s | 0 | (s, 0, →) |
| 3 | s | 1 | (s, 1, →) |
| 4 | s | □ | (q, □, ←) |
| 5 | q | 0 | (q, 1, ←) |
| 6 | q | 1 | (“h”, 0, -) |
| 7 | q | ► | (“h”, ►, -) |

## Bài 2:

Xây dựng máy Turing M3 thực hiện việc thay tất cả các số 0 trong một dãy nhị phân thành các số 1 và ngược lại. Ví dụ: 01001 ⇒ 10110.

Lời giải:

M3 = (K, ∑, δ, s)

Trong đó:

K = {s, “h”},

∑ = {0 ,1, □, ►},

s là trạng thái xuất phát,

h là trạng thái dừng.

Bảng hàm chuyển δ:

Đầu đọc chạy từ trái sang phải, cứ gặp 0 thì chuyển thành 1 và ngược lại.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | q | σ | δ(q, σ) = (p, ρ, {→ | ← | -}) |
| 1 | s | ► | (s, ►, →) |
| 2 | s | 0 | (s, 1, →) |
| 3 | s | 1 | (s, 0, →) |
| 4 | s | □ | (“h”, □, -) |

## Bài 3:

Xây dựng máy Turing M4 thực hiện việc loại bỏ các chữ số 0 trong dãy nhị phân. Sau khi bỏ cần dồn dãy lại. Ví dụ: 01001 ⇒ 11.

Lời giải:

M4 = (K, ∑, δ, s)

Trong đó:

K = {s1, s2, s3, q1, q2, “h”},

∑ = {0 ,1, □, ►, D},

s1 là trạng thái xuất phát,

D cho phép xóa kí tự trên băng

Bảng hàm chuyển δ:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Q | σ | δ(q, σ) = (p, ρ, {→ | ← | -}) |
| 1 | s1 | ► | (s1, ►, →) |
| 2 | s1 | 0 | (q1, ►, ←) |
| 3 | q1 | ► | (s1, D, →) |
| 4 | s1 | 1 | (s2, 1, →) |
| 5 | s2 | 1 | (s2, 1, →) |
| 6 | s2 | 0 | (q2, 1, ←) |
| 7 | q2 | 1 | (q2, 1, ←) |
| 8 | q2 | ► | (s3, D, →) |
| 9 | s3 | 1 | (s1, ►, →) |
| 10 | s1 | □ | (“h”, □, - ) |
| 11 | s2 | □ | (“h”, □, - ) |

Giải thích các quy tắc:

(1) s1 xuất phát

(2) s2 gặp bit 0 đầu tiên chuyển nó thành ► và quay trở lại để xóa dấu ► (trạng thái q1)

(4) nếu s1 và gặp bit 1 thì ghi nhận bên trái đã có bit 1, chuyển sang trạng thái s2

(6) nếu trong trạng thái s2 (bên trái đang có bit 1) và gặp bit 0 thì chuyển sang trạng thái q2, các quy tắc 7 đến 9 mô tả cách dồn các bit 1 sang bên phải

(10) ,(11) mô tả trạng thái dừng.

Ví dụ:

Khi cho máy chạy trên băng ►001001□

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | trạng thái q | kí tự σ | hàm chuyển  δ(q, σ) = (p, ρ, {→ | ← | -}) | dữ liệu trên băng |
| 0 | s1 | ► | (s1, ►, →) (1) | ►001001□ |
| 1 | s1 | 0 | (q1, ►, ←) (2) | ►001001□ |
| 2 | q1 | ► | (s1, D, →) (3) | ►►01001□ |
| 3 | s1 | ► | (s1, ►, →) (1) | ►01001□ |
| 4 | s1 | 0 | (q1, ►, ←) (2) | ►01001□ |
| 5 | q1 | ► | (s1, D, →) (3) | ►►1001□ |
| 6 | s1 | ► | (s1, ►, →) (1) | ►1001□ |
| 7 | s1 | 1 | (s2, 1, →) (4) | ►1001□ |
| 8 | s2 | 0 | (q2, 1, ←) (6) | ►1001□ |
| 9 | q2 | 1 | (q2, 1, ←) (7) | ►1101□ |
| 10 | q2 | ► | (s3, D, →) (8) | ►1101□ |
| 11 | s3 | 1 | (s1, ►, →) (9) | 1101□ |
| 12 | s1 | 1 | (s2, 1, →) (4) | ►101□ |
| 13 | s2 | 0 | (q2, 1, ←) (6) | ►101□ |
| 14 | q2 | 1 | (q2, 1, ←) (7) | ►111□ |
| 15 | q2 | ► | (s3, D, →) (8) | ►111□ |
| 16 | s3 | 1 | (s1, ►, →) (9) | 111□ |
| 17 | s1 | 1 | (s2, 1, →) (4) | ►11□ |
| 18 | s2 | 1 | (s2, 1, →) (5) | ►11□ |
| 19 | s2 | □ | (“h”, □, -) (11) | ►11□ |

## Bài 4:

Xây dựng máy Turing M5 thực hiện việc kiểm tra một dãy số nhị phân có

đối xứng không. Ví dụ: 010010 ⇒ “yes”, 01100 ⇒ “no”.

Lời giải:

Ý tưởng: so sánh từng cặp bit ở hai vi trí đối xứng nhau từ ngoài vào trong.

Mô tả:

* Khi mỗi bit được kiểm tra, ta chuyển nó thành kí tự \*
* Kí hiệu các trạng thái :

L0: đang so sánh bit bên phải với bit bên trái, đã biết bit bên trái

bằng 0

L1: đang so sánh bit bên phải với bit bên trái, đã biết bit bên trái

bằng 1

R0: đang so sánh bit bên phải với bit bên trái, đã biết bit bên phải

bằng 0

R1: đang so sánh bit bên phải với bit bên trái, đã biết bit bên phải

bằng 1

KLi (i = 0,1): bắt đầu lấy giá trị của bit bên trái cần so sánh với bit

bên phải đã biết (bằng i)

KRi (i = 0,1): bắt đầu lấy giá trị của bit bên phải cần so sánh với

bit bên trái đã biết (bằng i)

L: bắt đầu 1 chu trình so sánh mới (mỗi chu trình so sánh 1 cặp bit

có vị trí đôi xứng nhau) bắt đầu từ bên trái

Tương tự cho R.

M5 = (K, ∑, δ, s)

Trong đó:

K = {R, R0, R1, KRo, KR1, L, L0, L1, KLo, KL1, “yes”, “no”}

∑ = {0 ,1, □, ►, \*},

L là trạng thái xuất phát.

Bảng hàm chuyển δ:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | q | σ | δ(q, σ) = (p, ρ, {→ | ← | - }) |
| 1 | L | ► | (L, ►, →) |
| 2 | L | 0 | (L0, \*, →) |
| 3 | L | 1 | (L1, \*, →) |
| 4 | L0 | 0 | (L0, 0, →) |
| 5 | L0 | 1 | (L0, 1, →) |
| 6 | L1 | 0 | (L1, 0, →) |
| 7 | L1 | 1 | (L1, 1, →) |
| 8 | L0 | □ | (KR0, □, ←) |
| 9 | L0 | \* | (KR0, \*, ←) |
| 10 | L1 | □ | (KR1, □, ←) |
| 11 | L1 | \* | (KR1, \*, ←) |
| 12 | KR0 | 1 | (“no”, \*, -) |
| 13 | KR0 | 0 | (R, \*, ←) |
| 14 | KR1 | 1 | (R, \*, ←) |
| 15 | KR1 | 0 | (“no”, \*, -) |
| 16 | R | 0 | (R0, \*, ←) |
| 17 | R | 1 | (R1, \*, ←) |
| 18 | R0 | 0 | (R0, 0, ←) |
| 19 | R0 | 1 | (R0, 1, ←) |
| 20 | R1 | 0 | (R1, 0, ←) |
| 21 | R1 | 1 | (R1, 1, ←) |
| 22 | R0 | □ | (KL0, □, →) |
| 23 | R0 | \* | (KL0, \*, →) |
| 24 | R1 | □ | (KL1, □, →) |
| 25 | R1 | \* | (KL1, \*, →) |
| 26 | KL0 | 1 | (“no”, \*, -) |
| 27 | KL0 | 0 | (L, \*, →) |
| 28 | KL1 | 1 | (L, \*, →) |
| 29 | KL1 | 0 | (“no”, \*, -) |
| 30 | L | \* | (“yes”, \*, -) |
| 31 | L | □ | (“yes”, □, -) |
| 32 | KL0 | \* | (“yes”, \*, -) |
| 33 | KL1 | \* | (“yes”, \*, -) |
| 34 | R | \* | (“yes”, \*, -) |
| 35 | KR0 | \* | (“yes”, \*, -) |
| 36 | KR1 | \* | (“yes”, \*, -) |
| 37 | KR1 | \* | (“yes”, \*, -) |
| 38 | KR1 | □ | (“yes”, □, -) |

Ví dụ:

Kiểm tra chuỗi bit 010010

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | q | Σ | δ(q, σ) = (p, ρ, {→ | ← | - }) | dữ liệu trên băng |
| 0 | L | ► | (L, ►, →) | ►010010□ |
| 1 | L | 0 | (L0, \*, →) | ►010010□ |
| 2 | L0 | 1 | (L0, 1, →) | ►\*10010□ |
| 3 | L0 | 0 | (L0, 0, →) | ►\*10010□ |
| 4 | L0 | 0 | (L0, 0, →) | ►\*10010□ |
| 5 | L0 | 1 | (L0, 1, →) | ►\*10010□ |
| 6 | L0 | 0 | (L0, 0, →) | ►\*10010□ |
| 7 | L0 | □ | (KR0, □, ←) | ►\*10010□ |
| 8 | KR0 | 0 | (R, \*, ←) | ►\*10010□ |
| 9 | R | 1 | (R1, \*, ←) | ►\*1001\*□ |
| 10 | R1 | 0 | (R1, 0, ←) | ►\*100\*\*□ |
| 11 | R1 | 0 | (R1, 0, ←) | ►\*100\*\*□ |
| 12 | R1 | 1 | (R1, 1, ←) | ►\*100\*\*□ |
| 13 | R1 | \* | (KL1, \*, →) | ►\*100\*\*□ |
| 14 | KL1 | 1 | (L, \*, →) | ►\*100\*\*□ |
| 15 | L | 0 | (L0, \*, →) | ►\*\*00\*\*□ |
| 16 | L0 | 0 | (L0, 0, →) | ►\*\*\*0\*\*□ |
| 17 | L0 | \* | (KR0, \*, ←) | ►\*\*\*0\*\*□ |
| 18 | KR0 | 0 | (R, \*, ←) | ►\*\*\*0\*\*□ |
| 19 | R | \* | (“yes”, \*, -) | ►\*\*\*\*\*\*□ |
| 20 | “yes” |  |  | ►\*\*\*\*\*\*□ |

Ví dụ:

Kiểm tra chuỗi bit 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | q | Σ | δ(q, σ) = (p, ρ, {→ | ← | - }) | dữ liệu trên băng |
| 0 | L | ► | (L, ►, →) | ►10□ |
| 1 | L | 1 | (L1, \*, →) | ►10□ |
| 2 | L1 | 0 | (L1, 0, →) | ►\*0□ |
| 3 | L1 | □ | (KR1, □, ←) | ►\*0□ |
| 4 | KR1 | 0 | (“no”, \*, -) | ►\*0□ |
| 5 | “no” |  |  | ►\*0□ |

Nhận xét:

1, quy tắc 30 xác nhận 1 chuỗi rỗng cũng là đối xứng.

2, Máy Turing M5 cho phép bảo tồn số lượng bit (thay các bit đã xét bằng dấu \* tương ứng), nhưng không cho phép bảo tồn cả chuỗi. Muốn giữ nguyên chuỗi sau khi kiểm tra đối xứng, thay cho dấu \* dùng để đánh dấu bit đã xét ta sẽ dùng kí tự c thay cho bit 0 và kí tự t thay cho bit 1. Sau khi kiểm tra tính đối xứng lại đổi lại các kí tự c → 0, t → 1, như vậy sẽ khôi phục được chuỗi ban đầu.

## Bài 5: (\*)

Ta quy ước biểu diễn số tự nhiên n bằng n+1 chữ số 1. Xây dựng máy Turing thực hiện phép cộng 2 hay nhiều số tự nhiên. Ví dụ: 111+11+1111 = 1111111 (2 + 1 + 3 = 6).

Lời giải:

Ý tưởng: khi gặp dấu “+” thì máy thực hiện thao tác xóa đi dấu “+” đó

và một bit 1 đi.

Ví dụ:

Xét tổng: n1 + n2 + …… + nk.

thì có k-1 dấu “+”, theo ý tưởng thì ta sẽ xóa k-1 dấu “+” và k-1 bit 1. Cuối cùng ta có (n1 + n2 + ….. + nk) + 1 bit 1 là biểu diễn của số (n1 + n2 + ….. + nk).

M6 = (K, ∑, δ, s1) trong đó:

K = {s1, q, s2, s3, “h”}

∑ = {1, □, ►, +, “h”, E}, (E: rỗng)

s1 là trạng thái xuất phát.

Bảng hàm chuyển δ:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | q | σ | δ(q, σ) = (p, ρ, {→ | ← | - }) |
| 1 | s1 | ► | (s1, ►, →) |
| 2 | s1 | 1 | (s1, 1, →) |
| 3 | s1 | + | (q, 1, ←) |
| 4 | q | 1 | (q, 1, ←) |
| 5 | q | ► | (s2, E, →) |
| 6 | s2 | 1 | (s3, E, →) |
| 7 | s3 | 1 | (s1, ►, →) |
| 8 | s1 | □ | (“h”, □, -) |

# Hàm đệ quy nguyên thủy

Chứng minh các hàm sau là đệ quy nguyên thủy

# Addition: a+b

Hàm cộng a + b = f+(a, b)

Ta sử dụng phép đệ quy nguyên thủy:

+) f+(0,b) = b = \_ là hàm đệ quy nguyên thủy. (1)

+) thấy rằng: và S(t) là hai hàm đệ quy nguyên thủy, vậy nên theo nguyên tắc đổi biến hàm: g(x, y, z) = S() cũng là hàm đệ quy nguyên thủy.

Mặt khác:

g(f+(a, b), a, b) = S((a, b, f+(a, b)))

= S(f+(a, b))

= S(a,b) = a + b + 1 = f+(S(a),b)

Vậy: g(f+(a, b), a, b) = f+(S(a), b) (2)

Từ (1) và (­­­­­­­­­­­­­­­­­­­­­­­­­­­2) suy ra f+(a, b) = a + b là hàm đệ quy nguyên thủy.

# Multiplication: a×b

Hàm nhân: f**×**(a,b) = a **×** b

Ta sử dụng phép đệ quy nguyên thủy:

+) f×(0,b) = 0 × b = 0 hàm hằng\_là hàm đệ quy nguyên thủy. (1)

+) Ta đã chứng minh f+(a, b) là hàm đệ quy nguyên thủy vậy nên:

g(a, b, c) = f+() cũng là hàm đệ quy

nguyên thủy.

mà:

g(f×(a, b), a, b) = f+((f×(a, b), a, b), (f×(a, b), a, b))

= f+( f×(a, b), b)

= f+(a **×** b, b) = (a **×** b) + b = (a+1) × b = f×(a+1, b)

Vậy: g(f×(a, b), a, b) = f×(a+1, b) là hàm đệ quy nguyên thủy (2)

Từ (1) và (2) => hàm nhân f**×**(a,b) là hàm đệ quy nguyên thủy.

# Exponentiation: ab

Hàm mũ: f^(a,b) =

Ta sử dụng phép đệ quy nguyên thủy:

+) f^(a,0) = = 1 hàm hằng\_là hàm đệ quy nguyên thủy. (1)

+) Do hàm f×(a, b) và (x1, … , xn) là các hàm đệ quy nguyên thủy nên hàm:

g(a, b, c) = f×((a, b, c), (a, b, c)) = a×b cũng là hàm đệ quy nguyên thủy.

Mà :

g(f^(a, b), a, b) = ×a = = f^(a, b+1) (2)

Từ (1) và (2) ta có hàm f^(a, b) = là hàm đệ quy nguyên thủy.

# Factorial a! : 0! = 1, a'! = a!×a'

Hàm giai thừa: fǃ(a) = a!

Sử dụng nguyên lý đệ quy nguyên thủy:

+) là hàm hằng suy ra là hàm đệ quy nguyên thủy. (1)

+) Hàm sau cũng là hàm đệ quy nguyên thủy:

.

mà:

. (2)

Từ (1) và (2) => là hàm đệ quy nguyên thủy.