



ESSAY LABS of Discrete Mathematics with applications

Cấu trúc rời rạc (Đại học Tôn Đức Thắng)

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC HÀNH
MÔN CẤU TRÚC RỜI RẠC**

ESSAY DISCRETE STRUCTURES

Người hướng dẫn: **GV. NGUYỄN QUỐC BÌNH**

Người thực hiện: **NGUYỄN TIẾN DŨNG – 52000883**

ĐOÀN PHƯƠNG NAM – 52000895

Lớp : 501044

Khoá : 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC HÀNH
MÔN CẤU TRÚC RỜI RẠC**

ESSAY DISCRETE STRUCTURES

Người hướng dẫn: **GV. NGUYỄN QUỐC BÌNH**

Người thực hiện: **NGUYỄN TIẾN DŨNG – 52000883**

ĐOÀN PHƯƠNG NAM – 52000895

Lớp : 501044

Khoá : 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

MỤC LỤC

MỤC LỤC	1
CHƯƠNG 1 – PHÂN CHIA CÔNG VIỆC	2
1.1 Đánh giá mức độ hoàn thành công việc như sau:.....	2
1.2. Ý kiến của các thành viên:	2
CHƯƠNG 2 – LÍ THUYẾT	3
2.1 Lí thuyết Reverse Polish.....	3
2.2 Lí thuyết Basic logic được sử dụng trong bảng chân trị.	4
CHƯƠNG 3 – GIẢI THÍCH	1
3.1 Function Infix2Postfix(Infix):	1
3.2 Function Postfix2Truthtable(Postfix):.....	5
CHƯƠNG 4 – KẾT QUẢ	9
1. TestCase 1: $R (P \& Q)$	9
2. Testcase 2: $\sim P (Q \& R) > R$	10
3. Testcase 3: $P (R \& Q)$	11
4. Testcase 4: $(P > Q) \& (Q > R)$	12
5. Testcase 5: $(P \sim Q) > \sim P = (P (\sim Q)) > \sim P$	13
6. Nhìn tổng quan về các Testcase qua truthtable	14
CHƯƠNG 5: TÀI LIỆU THAM KHẢO	17

CHƯƠNG 1 – PHÂN CHIA CÔNG VIỆC

1.1 Đánh giá mức độ hoàn thành công việc như sau:

STT	Họ tên	MSSV	Nhiệm vụ	Đánh giá hoàn thành	Mức độ	Điểm
1	Nguyễn Tiến Dũng	52000883	-Code python câu 1,2. - Làm lý thuyết chương 2	- Hoàn thành tốt	100%	
2	Đoàn Phương Nam	52000895	-Code python câu 1,2. - Fix bug code và testcase	- Hoàn thành tốt	100%	

1.2. Ý kiến của các thành viên:

- Nguyễn Tiến Dũng: đồng ý với ý kiến của trên.
- Đoàn Phương Nam: đồng ý với ý kiến của trên.

Thành viên thứ 1
(Ký và ghi rõ họ tên)

Nguyễn Tiến Dũng

Thành viên thứ 2
(Ký và ghi rõ họ tên)

Đoàn Phương Nam

CHƯƠNG 2 – LÍ THUYẾT

2.1 Lí thuyết Reverse Polish

Reverse Polish (RPN) là một phương pháp để đại diện cho các biểu thức trong đó các dấu phép tính được đặt sau khi các đối số được triển khai. Vào cuối những năm 1950, nhà triết học và nhà khoa học máy tính người Úc Charles Hamblin đã đề nghị đặt các dấu phép tính phía sau các đối số và từ đó tạo ra kí hiệu RPN.

Ví dụ: biểu thức RPN sau đây sẽ tạo ra tổng 2 và 3, cụ thể là $5: 2\ 3\ +$.

Ký hiệu Reverse Polish, còn được gọi là ký hiệu Postfix, tương phản với Infix của các biểu thức số học tiêu chuẩn.

Đặc điểm của reverse polish là các dấu ngoặc không bắt buộc. Các biểu thức reverse polish chỉ đơn giản được đánh từ trái sang phải và điều này đơn giản hơn rất nhiều so với việc tính toán biểu thức trong các chương trình máy tính. Ví dụ, biểu thức số học $(3 + 4) \times 5$ có thể được thể hiện trong RPN là $3\ 4\ +\ 5\ \times$.

Trong thực tế, reverse polish được sử dụng phổ biến bằng cách sử dụng cấu trúc Stack. Một biểu thức infix có thể được chuyển đổi thành một biểu thức reverse polish bằng cách sử dụng một thuật toán như đệ quy. Trong khoa học máy tính, phương pháp này được sử dụng trong các ngôn ngữ lập trình như PostScript, Forth, STOIC...

2.2 Lí thuyết Basic logic được sử dụng trong bảng chân trị.

Có tổng cộng 5 phép toán logic cơ bản:

1. *And*:

a. Kí hiệu: &

b. Giá trị bảng chân trị:

P	Q	P&Q
True	True	True
True	False	False
False	True	False
False	False	False

Bảng 1: Bảng chân trị của AND

2. *Or*:

a. Kí hiệu: |

b. Giá trị bảng chân trị:

P	Q	P Q
True	True	True
True	False	True
False	True	True
False	False	False

Bảng 2: Bảng chân trị của OR

3. *Not*:

a. Kí hiệu: ~

b. Giá trị bảng chân trị:

P	~P
True	False
False	True

Bảng 3: Bảng chân trị của Not

4. *Implication:*

a. Kí hiệu: \rightarrow

b. Giá trị bảng chân trị:

P	Q	$P \rightarrow Q$
True	True	True
True	False	False
False	True	True
False	False	True

Bảng 4: Bảng chân trị của implication

5. *Bi implication:*

a. Kí hiệu: \leftrightarrow

b. Giá trị bảng chân trị:

P	Q	$P \leftrightarrow Q$
True	True	True
True	False	False
False	True	False
False	False	True

Bảng 5: Bảng chân trị của bi implication

CHƯƠNG 3 – GIẢI THÍCH

3.1 Function Infix2Postfix(Infix):

Bước 1:

- Tạo 1 function với tên “Infix2Postfix”
- Tạo mảng Stack và Postfix

```
def Infix2Postfix(Infix):
    Operations = {
        '=': 1,
        '>': 2,
        '|': 3,
        '&': 4,
        '~': 5
    }
    Spec0 = set(["(", ")"])
    Stack = []
    Postfix = []
```

Bảng 1: Bước 1 của Infix2Postfix

Bước 2:

- Tạo 1 vòng lặp for để đọc biểu thức từ trái sang phải.
- Quét từng phép tính trong Infix.
- Nếu kí tự phép tính là một biểu thức thì ta có trường hợp như sau:
 - Nếu Stack rỗng thì mình sẽ gán giá trị đầu tiên cho Stack.
 - Kiểm tra độ ưu tiên nếu kí tự đang xét lớn hơn kí tự vừa vào Stack :
 - + Nếu lớn hơn sẽ đưa vào Stack
 - + Còn không thì sẽ đưa giá trị ra khỏi Stack và đưa vào Postfix.

```

for c in Infix:
    if c in Operations:
        if len(Stack) > 0:
            topSt = Stack[-1]

            if topSt in Operations.keys():
                if Operations[c] > Operations[topSt]:
                    Stack.append(c)
                else:
                    while topSt in Operations.keys() and Operations[topSt] > Operations[c]:
                        if len(Stack) > 0:
                            p = Stack.pop()
                            Postfix.append(p)

                        if len(Stack) > 0:
                            topSt = Stack[-1]
                        else:
                            break
                    Stack.append(c)
            else:
                Stack.append(c)
        else:
            Stack.append(c)

```

Bảng 2: Bước 2 của Infix2Postfix

Bước 3:

- Nếu giá trị đầu tiên là một dấu ngoặc, ta có trường hợp như sau:
 - Nếu là ngoặc đóng thì mình sẽ kiểm tra Stack là rỗng hay có giá trị
 - Đưa đối số trước topSt vào Postfix cho đến khi Stack đầy được hết tất cả giá trị ra ngoài và tìm thấy dấu ngoặc mở sẽ dừng lại. Nếu không tìm thấy dấu ngoặc mở sẽ báo lỗi.
- Nếu giá trị đầu tiên là phép tính thì ta sẽ đưa vào Postfix.
- Sau khi đọc toàn bộ biểu thức, tất cả còn lại trong Stack sẽ được đưa ra và đẩy các giá trị trong Postfix ra ngoài theo thứ tự.

```

elif c in Spec0:
    if c == ")":
        if len(Stack) > 0:
            topSt = Stack [-1]
            while topSt != "(":
                try:
                    t = Stack.pop()
                    Postfix.append(t)
                    topSt = Stack[-1]
                except IndexError:
                    raise ValueError("'(' Not found popping ")
            Stack.pop()
        else:
            raise ValueError("'')' can't addto top of stack if stack is empty ")
    else:
        Stack.append(c)
else:
    Postfix.append(c)

while len (Stack) > 0:
    topSt=Stack.pop()
    if topSt in Operations.keys():
        Postfix.append(topSt)
for i in Postfix:
    print(i,end='')
return Postfix

```

Bảng 3: Bước 3 của Infix2Postfix

Ví dụ: Biểu thức: $\sim P|(Q\&R)>R$

Từng bước của thuật toán như sau:

Stack = [\sim ,P]

pop = P

pop = P \sim

Stack = []

Stack1 = [Q]

pop = Q

Stack1 = [&,R]

$\text{pop} = R$
 $\text{pop} = \&$
 $\text{pop} = |$
 $\text{Stack} = [>, R]$
 $\text{pop} = R$
 $\text{pop} = >$
 $\text{Stack} []$

Kết quả của trường hợp này là: $P \sim Q R \& | R >$

3.2 Function Postfix2TruthTable(Postfix):

Bước 1:

- Thực hiện 5 hàm nhỏ để hỗ trợ việc tính toán bao gồm: and, or, not, implication, bi implication.

```
def implies(A,B):
    if A:
        return B
    else:
        return True

def bi(A,B):
    if A==B:
        return True
    else:
        return False

def lAnd(A,B):
    return A and B

def lOr(A,B):
    return A or B

def lNot(A):
    return not A
```

Bảng 4: Bước 4 của Postfix2TruthTable

Bước 2:

- Tạo 4 mảng:

- Mảng WorldLS để sắp xếp các biến ban đầu theo bảng chữ cái.
- Mảng TrueOrFalse để tạo giá trị theo thứ tự bảng chữ cái
- Mảng TruthTable để in ra kết quả
- Mảng Stack để tạo bảng chân trị.

- Tạo vòng lặp for để đọc biểu thức từ trái sang phải. Sau đó sắp xếp bảng chữ cái trong danh sách theo thứ tự.

- Tạo 1 bảng chân trị và đưa các giá trị vừa được sắp xếp vào bảng đó.

```
def Postfix2TruthTable(Postfix):
    WordLS = []
    TrueOrFalse=[]
    TruthTable=[]
    Stack= []
    count =0
    for p in set (Postfix):
        if (p>='A') and (p<='Z'):
            WordLS.append(p)
            count+=1

    WordLS.sort()
    table = list(itertools.product([False,True], repeat=count))

    for i in range(len(WordLS)):
        TrueOrFalse.append([t[i] for t in table])
        TruthTable.append(TrueOrFalse[i])
```

Bảng 5: Bước 5 của Postfix2TruthTable

Bước 3:

- Gán từng giá trị trong mảng Stack vào bảng chân trị.
- Sau đó quay lại mảng Stack và lặp lại như thế đến khi hết giá trị trong mảng.
- Trả lại TruthTable để kết thúc.

```

for i in range(len(Postfix)):
    temp = []
    temp1 = []
    temp2 = []

    if (Postfix[i] >= 'A') and (Postfix[i] <= 'Z'):
        for j in range(len(WordLS)):
            if Postfix[i] == WordLS[j]:
                Stack.append(TrueOrFalse[j])
    else:
        if Postfix[i] == '~':
            tmp1 = Stack.pop()
            for k in tmp1:
                temp.append(lNot(k))
            Stack.append(temp)
            Truthtable.append(temp)

        elif Postfix[i] == '&':
            temp1 = Stack.pop()
            temp2 = Stack.pop()
            for k in range(len(temp1)):
                temp.append(lAnd(temp1[k], temp2[k]))
            Stack.append(temp)
            Truthtable.append(temp)

        elif Postfix[i] == '|':
            temp1 = Stack.pop()
            temp2 = Stack.pop()
            for k in range(len(temp1)):
                temp.append(lOr(temp1[k], temp2[k]))
            Stack.append(temp)
            Truthtable.append(temp)

        elif Postfix[i] == '>':
            temp1 = Stack.pop()
            temp2 = Stack.pop()
            for k in range(len(temp1)):
                temp.append(implies(temp2[k], temp1[k]))
            Stack.append(temp)
            Truthtable.append(temp)

        elif Postfix[i] == '=':
            temp1 = (Stack.pop())
            temp2 = (Stack.pop())
            for k in range(len(temp1)):
                temp.append(bi(temp1[k], temp2[k]))
            Stack.append(temp)
            Truthtable.append(temp)

```

```

Truthtable=list(map(tuple,(zip(*Truthtable))))
print()
for i in Truthtable:
    print(i)
return Truthtable

```

Bảng 6: Bước 6 của Postfix2Truthtable

- Ta sẽ so sánh các Postfix để đưa ra được bảng chân trị , từ đó có thể biết được True or False như thế nào.
- Sau khi ta có được bảng chân trị theo dạng **list** và trả về Truth table.

CHƯƠNG 4 – KẾT QUẢ

1. TestCase 1: $R|(P \& Q)$

```
RPQ&|
(False, False, False, False, False)
(False, False, True, False, True)
(False, True, False, False, False)
(False, True, True, False, True)
(True, False, False, False, False)
(True, False, True, False, True)
(True, True, False, True, True)
(True, True, True, True, True)
```

Và ta có bảng chân trị là:

1	False	False	False	False	False
2	False	False	True	False	True
3	False	True	False	False	False
4	False	True	True	False	True
5	True	False	False	False	False
6	True	False	True	False	True
7	True	True	False	True	True
8	True	True	True	True	True
9					

2. Testcase 2: $\sim P|(Q \& R) > R$

```
P~QR&|R>
(False, False, False, True, False, True, False)
(False, False, True, True, False, True, True)
(False, True, False, True, False, True, False)
(False, True, True, True, True, True, True)
(True, False, False, False, False, False, True)
(True, False, True, False, False, False, True)
(True, True, False, False, False, False, True)
(True, True, True, False, True, True, True)
```

Và ta có bảng chân trị là:

1	False	False	False	True	False	True	False
2	False	False	True	True	False	True	True
3	False	True	False	True	False	True	False
4	False	True	True	True	True	True	True
5	True	False	False	False	False	False	True
6	True	False	True	False	False	False	True
7	True	True	False	False	False	False	True
8	True	True	True	False	True	True	True
9							

3. Testcase 3: $P|(R \& Q)$

```

PRQ&|
(False, False, False, False, False)
(False, False, True, False, False)
(False, True, False, False, False)
(False, True, True, True, True)
(True, False, False, False, True)
(True, False, True, False, True)
(True, True, False, False, True)
(True, True, True, True, True)

```

Và ta có bảng chân trị là:

1	False	False	False	False	False
2	False	False	True	False	False
3	False	True	False	False	False
4	False	True	True	True	True
5	True	False	False	False	True
6	True	False	True	False	True
7	True	True	False	False	True
8	True	True	True	True	True
9					

4. Testcase 4: $(P \rightarrow Q) \wedge (Q \rightarrow R)$

```
PQ→QR→&
(False, False, False, True, True, True)
(False, False, True, True, True, True)
(False, True, False, True, False, False)
(False, True, True, True, True, True)
(True, False, False, False, True, False)
(True, False, True, False, True, False)
(True, True, False, True, False, False)
(True, True, True, True, True, True)
```

Và ta có bảng chân trị là:

1	False	False	False	True	True	True
2	False	False	True	True	True	True
3	False	True	False	True	False	False
4	False	True	True	True	True	True
5	True	False	False	False	True	False
6	True	False	True	False	True	False
7	True	True	False	True	False	False
8	True	True	True	True	True	True
9						

5. Testcase 5: $(P|\sim Q) \supset \sim P = (P|(\sim Q)) \supset \sim P$

```
PQ~|P~>PQ~|P~>=
(False, False, True, True, True, True, True, True, True, True, True, True)
(False, True, False, False, True, True, False, False, True, True, True, True)
(True, False, True, True, False, False, True, True, False, False, False, True)
(True, True, False, True, False, False, False, True, False, False, False, True)
```

Và ta có bảng chân trị là:

1	False	False	True	True	True	True	True	True	True	True	True
2	False	True	False	False	True	True	False	False	True	True	True
3	True	False	True	True	False	False	True	True	False	False	True
4	True	True	False	True	False	False	False	True	False	False	True
5											

6. Nhìn tổng quan về các Testcase qua truth table

a. $R|(P \& Q)$

P	Q	R	$P \& Q$	$R (P \& Q)$
False	False	False	False	False
False	False	True	False	True
False	True	False	False	False
False	True	True	False	True
True	False	False	False	False
True	False	True	False	True
True	True	False	True	True
True	True	True	True	True

b. $\sim P|(Q \& R) > R$

P	Q	R	$\sim P$	$Q \& R$	$\sim P (Q \& R) > R$
False	False	False	True	False	False
False	False	True	True	False	True
False	True	False	True	False	False
False	True	True	True	True	True
True	False	False	False	True	True
True	False	True	False	False	True
True	True	False	False	False	True
True	True	True	False	True	True

c. $P|(R \& Q)$

P	Q	R	$R \& Q$	$P (R \& Q)$
False	False	False	False	False
False	False	True	False	False
False	True	False	False	False
False	True	True	True	True
True	False	False	False	True
True	False	True	False	True
True	True	False	False	True
True	True	True	True	True

d. $(P > Q) \& (Q > R)$

P	Q	R	$P > Q$	$Q > R$	$(P > Q) \& (Q > R)$
False	False	False	True	True	True
False	False	True	True	True	True
False	True	False	True	False	False
False	True	True	True	True	True
True	False	False	False	True	False
True	False	True	False	True	False
True	True	False	True	False	False
True	True	True	True	True	True

e. $(P|\sim Q) \supset \sim P = (P|(\sim Q)) \supset \sim P$

P	Q	$\sim Q$	$P \sim Q$	$\sim P$	$(P \sim Q) \supset \sim P$	$\sim Q$	$P (\sim Q)$	$\sim P$	$(P (\sim Q)) \supset \sim P$	$(P \sim Q) \supset \sim P = (P (\sim Q)) \supset \sim P$
F	F	T	T	T	T	T	T	T	T	T
F	T	F	F	T	T	F	F	T	T	T
T	F	T	T	F	F	T	T	F	F	T
T	T	F	T	F	F	F	T	F	F	T

CHƯƠNG 5: TÀI LIỆU THAM KHẢO

- [1] <https://www.merriam-webster.com/dictionary/reverse%20Polish%20notation>
- [2] https://en.wikipedia.org/wiki/Reverse_Polish_notation
- [3] [Infix to Postfix](#)
- [4] [Truth table](#)