

CANTHO UNIVERSITY 

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

BÀI BÁO CÁO


ĐỀ TÀI: KIỂM TRA VĂN PHẠM CHÍNH QUY

Giảng viên giảng dạy:
 Ths: Trần Nguyễn Dương Chi

Nhóm 13:

➢ Phan Nhật Ý	B1913283
➢ Nguyễn Quốc Trâm	B1913276
➢ Nguyễn Trường An	B1913213
➢ Nguyễn Đình Kha	B1913232

www.ctu.edu.vn 1

 **Nội dung**

CANTHO UNIVERSITY

1. Tổng quan về Văn phạm chính quy
2. Phương pháp xây dựng VPCQ
3. Giải thuật kiểm tra tuyến tính trái
4. Giải thuật kiểm tra tuyến tính phải
5. Demo chương trình

www.ctu.edu.vn 2



CANTHO UNIVERSITY

1. Tổng quan

- Nội dung đề tài: Kiểm tra một văn phạm được cho có phải là văn phạm chính quy (VPCQ) hay không?
- Khái niệm VPQC: **Văn phạm chính quy** là văn phạm mà tất cả các luật sinh của nó đều có dạng **tuyến tính trái** (hoặc **tuyến tính phải**)
 - ✓ **Tuyến tính trái**: dạng $A \rightarrow Bw$ hoặc $A \rightarrow w$
 - ✓ **Tuyến tính phải**: dạng $A \rightarrow wB$ hoặc $A \rightarrow w$

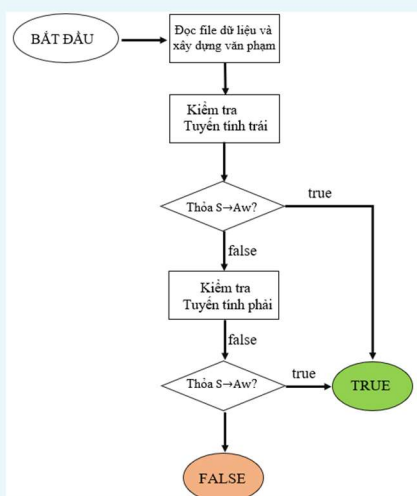
3

www.ctu.edu.vn



CANTHO UNIVERSITY

1. Tổng quan



4

www.ctu.edu.vn



CANTHO UNIVERSITY

2. Xây dựng văn phạm

1. Khởi tạo lớp “Vanpham” với các thuộc tính:

$V = \square$: Tập hợp các biến

$T = \square$: Tập các kí hiệu kết thúc

$S = ''$: Ký tự bắt đầu

$P = \{ \}$: Tập luật sinh

```
class Vanpham:
    V = []
    T = []
    S = ''
    P = {}
```

5

www.ctu.edu.vn



CANTHO UNIVERSITY

2. Xây dựng văn phạm

2. Phương thức tạo văn phạm:

- Nhận luật sinh có dạng:
{"S": "bSb|T", "T": "aT"}
- Sử dụng các phép toán để lấy giá trị cho những thuộc tính của của văn phạm

```
def taoVanPham(self, data):
    # set V
    self.P = data
    self.V = list(self.P.keys())
    # set T
    value = ''
    for i in range(len(self.V)):
        value = value + self.P.get(self.V[i])
    value1 = list(value)
    value2 = list(value.lower())
    for i in range(len(value1)):
        if(value1[i] in value2):
            self.T.append(value1[i])
    try:
        self.T.remove("|")
    except:
        print('', end='')
    s = set(self.T)
    self.T = list(s)
    self.T.sort()
    # Set S
    self.S = self.V[0]
```

6

www.ctu.edu.vn



CANTHO UNIVERSITY

2. Xây dựng văn phạm

3. Các phương thức hỗ trợ

```
def inVanPham(self):
    print("G={ " + str(self.V), " " + str(self.T)+", "+str(self.S)+"",P}")
    print("      P: " + str(self.P))

def layListValue(self, key):
    listValue = []
    listValue = (self.P.get(key)).split('|')
    return listValue
```

www.ctu.edu.vn 7



CANTHO UNIVERSITY

3. Kiểm tra tuyến tính trái

➤B1. Khởi tạo các biến cần thiết.

```
str1 = ''; str2 = ''; str3 = ''
lv1 = list(listValues[j])
lv2 = list((listValues[j]).upper())
lv3 = list((listValues[j]).lower())
```

➤B2. Lấy biến (thuộc tập biến V) trong chuỗi giá trị của luật sinh. Gán vào str2.

➤B3: Đảo ngược hai danh sách lv1 và lv3. Lấy chuỗi ký tự kết thúc là các ký tự in thường cùng thuộc hai danh sách. Gán vào str3 .

➤B4: Ghép str1 = str2 + str3.

➤B5: So sánh str1 với giá trị của luật sinh nếu giống thì thỏa S->Aw hoặc S-> w.

www.ctu.edu.vn 8



CANTHO UNIVERSITY

3. Kiểm tra tuyến tính trái

```
def ktTuyenTinhTrai(self):
    result = True
    print("Kiểm tra tuyến tính trái", end='')
    for i in range(len(self.V)+1):
        listValues = self.layListValue(self.V[i])
        print("\n -Kiểm tra luật sinh", self.V[i], "->", listValues, end='')
        for j in range(len(listValues)):
            str1 = ''; str2 = ''; str3 = ''
            lv1 = list(listValues[j])
            lv2 = list(listValues[j].upper())
            lv3 = list(listValues[j].lower())
            print("\n   +Luật sinh: ", self.S+"->"+listValues[j])
            if (lv1[0] == lv2[0]):
                str2 = lv1[0]
            if (str2 != ''):
                print("   +Bien:", str2)
            else:
                print("   +Bien: none")
```

9

www.ctu.edu.vn



CANTHO UNIVERSITY

3. Kiểm tra tuyến tính trái

```
lv1.reverse()
lv3.reverse()
for k in range(len(lv1)):
    if (lv1[k] == lv3[k]):
        str3 = lv1[k] + str3
print("   +Ky hiệu kết thúc:", str3)
str1 = str2 + str3
print("   +Chuỗi kiểm tra: ", str1)
if (str1 != listValues[j]):
    result = False
    print("   ->Luật sinh không thỏa tuyến tính trái!")
    break
if (result == False):
    break
return result
```

10

www.ctu.edu.vn



CANTHO UNIVERSITY

4. Kiểm tra tuyến tính phải

➤ Tương tự Kiểm tra tuyến tính trái.

```
str1 = ''; str2 = ''; str3 = ''
lv1 = list(listValues[j])
lv2 = list((listValues[j]).upper())
lv3 = list((listValues[j]).lower())
```

➤ Nhưng không cần đảo ngược hai danh sách lv1, lv3.

➤ B5: So sánh str1 với giá trị của luật sinh nếu giống thì thỏa S->wA hoặc S-> w.

11

www.ctu.edu.vn



CANTHO UNIVERSITY

4. Kiểm tra tuyến tính phải

```
def ktTuyenTinhPhai(self):
    result = True
    print("Kiểm tra tuyến tính phải", end='')
    for i in range(len(self.V)):
        listValues = self.layListValue(self.V[i])
        print("\n -Kiểm tra luật sinh", self.V[i], "->", listValues, end='')
        for j in range(len(listValues)):
            str1 = ''; str2 = ''; str3 = ''
            lv1 = list(listValues[j])
            lv2 = list((listValues[j]).upper())
            lv3 = list((listValues[j]).lower())
            print("\n   +Luật sinh: ", self.S+"->"+listValues[j])
            if (lv1[-1] == lv2[-1]):
                str2 = lv1[-1]
            if (str2 != ''):
                print("   +Bien:", str2)
            else:
                print("   +Bien: none")
```

12

www.ctu.edu.vn



CANTHO UNIVERSITY

4. Kiểm tra tuyến tính phải

```
for k in range(len(lv1)):
    if (lv1[k] == lv3[k]):
        str3 = str3 + lv1[k]
print("  +Ky hieu ket thuc:", str3)
str1 = str3+str2
print("  +Chuoi kiem tra: ", str1)
if (str1 != listValues[j]):
    result = False
    print("  ->Luât sinh không thỏa tuyến tính phải!")
    break
if (result == False):
    break
return result
```

13

www.ctu.edu.vn

CANTHO UNIVERSITY

5. Chạy demo chương trình

14

www.ctu.edu.vn

CANTHO UNIVERSITY 

Cám ơn Cô và các bạn đã theo dõi!



www.ctu.edu.vn 15