**UNIVERSITY OF INFORMATION TECHNOLOGY**

**FALCUTY OF COMPUTER ENGINEERING**

**---o0o---**

**A picture containing icon

Description automatically generated**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI: CHƯƠNG TRÌNH PYTHON THỰC HIỆN VIỆC RÚT GỌN BẢNG TRẠNG THÁI**

**LECTURER: TRƯƠNG VĂN CƯƠNG**

**Name: Võ Quang Nhật**

**Student ID: 21522421**

**Class: MTIO2021**

**Course ID: CE118.N22**

**Ho Chi Minh city, Date 30th May 2023**

**MỤC LỤC:**

**CHƯƠNG 1: TIÊU ĐỀ VÀ TÓM TẮT** 3

* 1. **Sơ lược về nội dung** 3
  2. **Mục tiêu và kết quả hướng đến** 3

**CHƯƠNG 2: GIỚI THIỆU**

**2.1. Giới thiệu về vấn đề nghiên cứu** 4

**2.2. Phạm vi hiện thực của đồ án** 4

**CHƯƠNG 3: CƠ SỞ LÝ THUYẾT**

**3.1. Cơ sở lý thuyết thiết kế luận lý số** 5

**3.1.1. Các phương pháp rút gọn bảng trạng thái** 5

**3.2. Ngôn ngữ và thư viện được sử dụng** 5

**3.2.1. Ngôn ngữ lập trình Python** 5

**3.2.2. Thư viện Pandas** 6

**3.2.3. Thư viện Numpy** 7

**CHƯƠNG 4: LƯU ĐỒ GIẢI THUẬT VÀ CÁC CHƯƠNG TRÌNH** 8

**4.1. Lưu đồ giải thuật chương trình chính.** 8

**4.2. Lưu đồ giải thuật của các chương trình con** 9

**4.2.1. Lưu đồ giải thuật gom nhóm TTHT theo Output** 9

**4.2.2. Lưu đồ giải thuật gom nhóm TTKT theo Output** 10

**4.2.3. Lưu đồ giải thuật kiểm tra điều kiện và phân hoạch lại** 11

**4.2.4. Lưu đồ giải thuật rút gọn trạng thái trong nhóm tương đương** 13

**4.3. Source code chương trình:** 14

**4.3.1. Hướng tiếp cận chương trình** 14

**CHƯƠNG 5: HIỆN THỰC VÀ KẾT QUẢ** 15

**5.1. Điều kiện thực nghiệm** 15

**5.2. Kết quả thu được** 15

**CHƯƠNG 6: KẾT LUẬN** 20

**CHƯƠNG 1: TIÊU ĐỀ VÀ TÓM TẮT**

* 1. *Sơ lược về nội dung :*

Nội dung cơ bản của đồ án là: Xác định và tối thiểu hóa trạng thái của một bảng trạng thái. Trạng thái là tập hợp các điều kiện hoặc thông tin có thể xảy ra trong hệ thống. Mục tiêu là tìm cách giảm bớt số lượng trạng thái không cần thiết và không làm mất đi tính chất vốn có của hệ thống.

* 1. *Mục tiêu và kết quả hướng đến của đồ án:*
     1. *Mục tiêu:*

- Tìm hiểu về lý thuyết và phương pháp tối thiểu hóa trạng thái.

- Áp dụng các phương pháp và thuật toán tối thiểu hóa để giảm số lượng trạng thái trong bảng ban đầu.

- Tạo ra một mô hình tối thiểu hóa trạng thái có tính chính xác và ổn định.

* + 1. *Kết quả hướng đến:*

- Tối thiểu hóa số lượng trạng thái trong bảng ban đầu mà không làm mất đi tính chất quan trọng của hệ thống.

- Xây dựng mô hình tối thiểu hóa mới, bao gồm các trạng thái đã được tối thiểu và các qui tắc chuyển đổi tương ứng.

- Hiện thực hóa mô hình tối thiểu để có thể sử dụng và kiểm tra tính hợp lý.

Kết quả của đồ án là bảng trạng thái đã được tối thiếu hóa giúp giảm số lượng trạng thái và đơn giản hóa hệ thống.

**CHƯƠNG 2: GIỚI THIỆU.**

2.1*. Giới thiệu về vấn đề nghiên cứu:*

Vấn đề của đồ án tối thiểu hóa bảng trạng thái được đặt ra như sau:

Mỗi hệ thống có thể có nhiều trạng thái hiện tại khác nhau và số lượng trạng thái có thể trở nên rất lớn và phức tạp. Vấn đề nghiên cứu là làm sao để giảm số lượng trạng thái mà vẫn đảm bảo tính chính xác và đúng đắn của hệ thống. Mục tiêu là tìm ra phương pháp và thuật toán tối thiểu hóa trạng thái hiệu quả và áp dụng vào bảng trạng thái ban đầu để giảm độ phức tạp của hệ thống cũng như chi phí.

2.2. *Phạm vi hiện thực của đồ án:*

Phạm vi hiện thực của đồ án này sẽ tập trung vào các khía cạnh:

- Mô hình bảng trạng thái ban đầu: bảng trạng thái ban đầu sẽ được xây dựng bởi các trạng thái và các qui tắc chuyển đổi tương ứng. Mô hình ban đầu này phải được xây dựng với các tiêu chuẩn phù hợp với việc thực hiện rút gọn trạng thái:

+ Tên trạng thái phải được đặt theo qui định là tiền tố là một chữ cái bất kỳ trong bảng chữ cái và hậu tố phải là một số nguyên

+Số lượng trạng thái xây dựng khi thực hiện chạy chương trình trên các máy tính thông thường nên nhỏ hơn 1000 trạng thái (chương trình có thể chạy đến 10000 trạng thái nhưng quá trình thực hiện khá lâu)

- Quá trình tối thiểu hóa trạng thái: chương trình sẽ tập trung vào việc áp dụng các phương pháp và thuật toán tối thiểu hóa để giảm số lượng trạng thái trong bảng ban đầu. Phạm vi thực hiện sẽ bao gồm việc triển khai các thuật toán tối thiểu hóa trạng thái và xây dựng bảng trạng thái sau khi quá trình tối thiểu hóa hoàn thành

- Kiểm tra tính đúng đắn: Sau khi tối thiểu hóa trạng thái, phạm vi của chương trình cũng bao gồm việc kiểm tra tính đúng đắn của mô hình tối thiểu (ví dụ như kiểm tra tính đúng đắn của các bảng trạng thái tối thiểu được rút gọn bằng tay hay những công cụ khác …)

- Xây dựng giao diện hoặc ứng dụng: Phạm vi cũng có thể mở rộng thành việc xây dựng hoặc phát triển ứng dụng sử dụng mô hình tối thiểu hóa trạng thái bảng trạng thái. Điều này giúp người dùng có thể dễ dàng tương tác với mô hình tối thiểu hóa thông qua phần mềm hoặc web app để có thể sử dụng file đầu vào, thông qua chương trình để xem được kết quả đầu ra.

Phạm vi hiện thực của đồ án sẽ tập trung vào quá trình tối thiểu hóa trạng thái, từ việc xây dựng input đầu vào, áp dụng các phương pháp tối thiểu hóa, kiểm tra tính đúng đắn và có thể xây dựng giao diện hoặc ứng dụng để có thể sử dụng mô hình tối thiểu hóa một cách thuận tiện.

**CHƯƠNG 3: CƠ SỞ LÝ THUYẾT**

*3.1. Cơ sở lý thuyết thiết kế luận lý số:*

*3.1.1. Các phương pháp rút gọn bảng trạng thái:*

*3.1.1.1. Phương pháp phân hoạch (gom nhóm):*

- Nội dung: Phương pháp phân hoạch (gom nhóm) là một phương pháp tiếp cận trong việc tối thiểu hóa bảng trạng thái. Phương pháp này nhằm nhóm các trạng thái có hành vi tương tự lại với nhau. Mỗi nhóm sẽ đại diện cho một trạng thái output tổng quát, giúp giảm các trạng thái tương đương với nhau trong bảng.

- Cách thực hiện: Phương pháp này bắt đầu bằng việc tạo các nhóm ban đầu dựa trên các output khác nhau trong bảng trạng thais bắt đầu. Sau đó các trạng thái trong nhóm được so sánh với nhau và nhóm lại thành nhóm con mới. Quá trình này lặp lại cho đến khi không còn sự phân hoạch trong các nhóm con nữa và mỗi nhóm đại diện cho một trạng thái tổng quát.

*3.1.1.2. Phương pháp lập bảng quan hệ:*

- Nội dung: Phương pháp lập bảng quan hệ là một phương pháp khác để tối thiểu hóa bảng trạng thái. Phương pháp này tập trung vào việc phân tích quan hệ giữa các trạng thái trong bảng và xác định các trạng thái tương đương với nhau để gom về thành một trạng thái duy nhất để giảm số lượng trạng thái.

- Cách thực hiện: Phương pháp này bắt đầu bằng việc xây dựng một bảng quan hệ trong đó mỗi cột và hàng đại diện cho các trạng thái trong bảng ban đầu. Qua đó, quan hệ giữa các trạng thái được xác định dựa trên TTKT và output tương ứng của chúng. Sau đó các trạng thái có quan hệ tương đương nhau được kết hợp lại thành một trạng thái duy nhất, giảm số lượng trạng thái trong bảng.

*3.2. Ngôn ngữ và thư viện được sử dụng:*

*3.2.1. Ngôn ngữ lập trình python:*

*3.2.1.1. Sơ lược về ngôn ngữ lập trình Python:*

- Python là một ngôn ngữ lập trình thông dịch, mạnh mẽ và dễ học. Nó được phát triển vào cuối những năm 1980 và đã trở thành một trong những ngôn ngữ phổ biến nhất trên thế giới. Dưới đây là một số đặc điểm cơ bản của Python:

+ Dễ học và đọc: Python có cú pháp rõ ràng và dễ hiểu, giúp người mới học lập trình dễ dàng tiếp cận. Ngôn ngữ này sử dụng cấu trúc gọn nhẹ và không yêu cầu các dấu ngoặc nhọn hoặc dấu chấm phẩy, làm cho mã nguồn dễ đọc và dễ bảo trì.

+ Đa năng: Python là một ngôn ngữ đa năng có thể sử dụng trong nhiều lĩnh vực khác nhau. Nó hỗ trợ các phong cách lập trình khác nhau như lập trình hướng đối tượng, lập trình cấu trúc và lập trình hàm

+ Thư viện phong phú: Python có một hệ sinh thái thư viện mạnh mẽ và phong phú. Có hàng ngàn thư viện và framework khác nhau được phát triển bởi cộng đồng Python, cung cấp các công cụ và chức năng đa dạng để giải quyết các vấn đề từ đơn giản đến phức tạp.

*3.2.1.2. Các ứng dụng cơ bản của ngôn ngữ lập trình Python:*

- Các ứng dụng cơ bản của ngôn ngữ lập trình Python:

+ Ứng dụng web: Python được sử dụng phổ biến trong phát triển ứng dụng web. Có các framework như Django và Flask giúp xây dựng các ứng dụng web mạnh mẽ, bảo mật và dễ dàng triển khai.

+Khoa học dữ liệu và trí tuệ nhân tạo: Python là một trong những ngôn ngữ phổ biến trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo. Có các thư viện như NumPy, Pandas, Matplotlib và TensorFlow cho phép xử lý dữ liệu, phân tích, học máy và thực hiện các nhiệm vụ trí tuệ nhân tạo.

+Hệ thống và mạng: Python có thư viện chuẩn hỗ trợ việc phát triển các ứng dụng hệ thống và mạng. Với thư viện như Socket, Threading và Multiprocessing, Python cho phép xây dựng các ứng dụng mạng, máy chủ, và các công cụ quản lý hệ thống.

+ Kịch bản hóa: Python là một ngôn ngữ kịch bản hóa mạnh mẽ, giúp tự động hóa các tác vụ lặp lại hoặc thực hiện các tác vụ đơn giản. Điều này giúp tiết kiệm thời gian và nâng cao hiệu suất làm việc.

*3.2.2. Thư viện Pandas:*

Pandas là một thư viện mã nguồn mở phổ biến trong ngôn ngữ lập trình Python, được sử dụng rộng rãi trong việc xử lý và phân tích dữ liệu. Nó cung cấp các cấu trúc dữ liệu và công cụ mạnh mẽ để làm việc với dữ liệu số liệu dạng bảng (tabular data), giúp dễ dàng thực hiện các phép tính, truy vấn, tiền xử lý và phân tích dữ liệu phức tạp.

Dưới đây là một số đặc điểm và tính năng chính của thư viện Pandas:

+ DataFrame: Đây là cấu trúc dữ liệu chính trong Pandas, tương đương với bảng dữ liệu trong hệ quản trị cơ sở dữ liệu hoặc spreadsheet. DataFrame cho phép lưu trữ và xử lý dữ liệu theo các cột và hàng. Nó cung cấp các phương thức để tạo, truy cập, sửa đổi và thao tác với dữ liệu trong DataFrame một cách dễ dàng.

+ Series: Đây là cấu trúc dữ liệu một chiều trong Pandas, tương đương với một cột trong DataFrame. Series giúp lưu trữ và xử lý các dữ liệu một chiều như chuỗi thời gian, chuỗi số, chuỗi văn bản, vv. Nó cung cấp các phương thức để truy cập, lọc và tính toán trên dữ liệu trong Series.

+ Xử lý dữ liệu: Pandas cung cấp nhiều phương thức và chức năng để xử lý dữ liệu, bao gồm: lọc dữ liệu theo các điều kiện, sắp xếp, nhóm dữ liệu, gộp dữ liệu từ các nguồn khác nhau, xoá bỏ dữ liệu trùng lặp, và thực hiện các phép biến đổi dữ liệu.

+ Đọc và ghi dữ liệu: Pandas hỗ trợ đọc và ghi dữ liệu từ nhiều định dạng tệp tin như CSV, Excel, SQL, HDF5, vv. Nó giúp dễ dàng nhập và xuất dữ liệu từ và đến các nguồn dữ liệu khác nhau.

+ Phân tích và thống kê: Pandas cung cấp các công cụ mạnh mẽ để thực hiện phân tích và thống kê dữ liệu. Nó hỗ trợ tính toán các thống kê cơ bản, như min, max, mean, median, và tính toán các chỉ số thống kê khác. Ngoài ra, nó cũng cung cấp khả năng xử lý và thống kê dữ liệu theo nhóm, và hỗ trợ việc vẽ biểu đồ để trực quan hóa dữ liệu.

Thư viện Pandas là một công cụ mạnh mẽ và linh hoạt trong việc xử lý dữ liệu trong Python. Với những tính năng và khả năng của nó, Pandas đã trở thành một công cụ không thể thiếu đối với các nhà phân tích dữ liệu, khoa học dữ liệu, và các lập trình viên làm việc với dữ liệu.

*3.2.3. Thư viện Numpy:*

NumPy là một thư viện mã nguồn mở cho Python, là viết tắt của "Numerical Python". Nó cung cấp một cấu trúc dữ liệu mạnh mẽ và các công cụ để làm việc với mảng đa chiều và tính toán khoa học số trong Python. NumPy được sử dụng rộng rãi trong các lĩnh vực như khoa học dữ liệu, tính toán khoa học, trí tuệ nhân tạo và phân tích dữ liệu.

Dưới đây là một số đặc điểm và tính năng chính của thư viện NumPy:

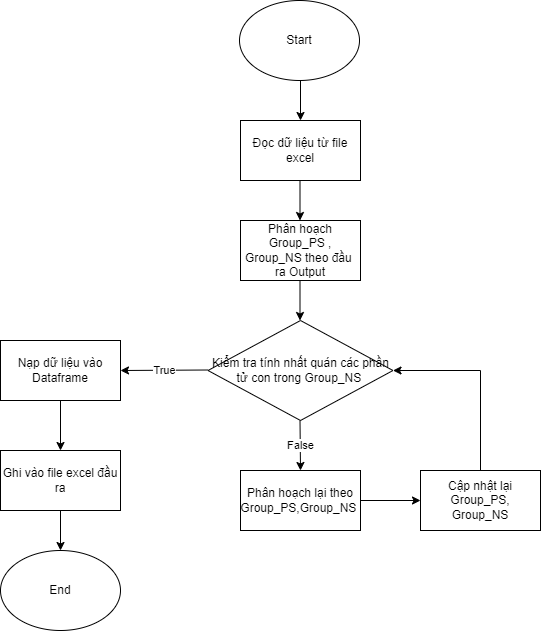
+ Mảng đa chiều: NumPy cung cấp đối tượng ndarray, là một mảng đa chiều (ndarray) trong Python. Mảng ndarray cho phép lưu trữ dữ liệu nhiều chiều, chẳng hạn như mảng 1D, mảng 2D hoặc thậm chí mảng nhiều chiều hơn. Mảng ndarray cung cấp các phương thức mạnh mẽ để truy cập, truy vấn và xử lý dữ liệu trong mảng một cách hiệu quả.

+ Tính toán vectorized: NumPy cho phép thực hiện các phép tính trên toàn bộ mảng một cách hiệu quả và nhanh chóng thông qua tính toán vectorized. Thay vì sử dụng vòng lặp để thực hiện các phép tính trên từng phần tử của mảng, NumPy cho phép áp dụng các phép tính trên toàn bộ mảng một cách đồng thời, làm tăng hiệu suất tính toán.

+ Hỗ trợ các phép tính toán số học và logic: NumPy cung cấp các hàm và phép toán số học cơ bản như cộng, trừ, nhân, chia, mũ, căn bậc hai và các phép tính logic như AND, OR, NOT. Điều này rất hữu ích trong việc thực hiện các phép tính và biến đổi dữ liệu trong quá trình phân tích và tính toán số học.

**CHƯƠNG 4: LƯU ĐỒ GIẢI THUẬT VÀ CÁC CHƯƠNG TRÌNH**

*4.1. Lưu đồ giải thuật chương trình chính*:



- Giải thích:

+Bắt đầu chương trình, chương trình sẽ đọc dữ liệu từ file excel chứa các bảng trạng thái chưa được rút gọn bao gồm TTHT, TTKT khi X=0 và khi X=1 và ngõ ra Output khi X=0 và X=1.

+ Sau khi đọc dữ liệu từ file excel và nạp vào các mảng có tên tương ứng, chương trình sẽ thực hiện gom nhóm TTHT vào Group\_PS bằng cách gom các trạng thái có giá trị ngõ ra Output giống nhau vào một mảng con trong mảng Group\_PS. Tiếp theo chương trình sẽ dựa vào Group\_PS đã gom để gom nhóm Group\_NS.

+ Sau khi gom 2 nhóm Group\_PS và Group\_NS, chương trình sẽ thực hiện một vòng lặp với điều kiện là kiểm tra xem trong Group\_NS có mảng con nào chứa cặp TTKT khác với các cặp còn lại không:

\* Nếu kết quả so sánh trả về = False: Thực hiện phân hoạch lại 2 nhóm Group\_PS và Group\_NS, sau đó cập nhật lại 2 nhóm Group\_PS và Group\_NS và tiếp tục so sánh với điều kiện của vòng lặp.

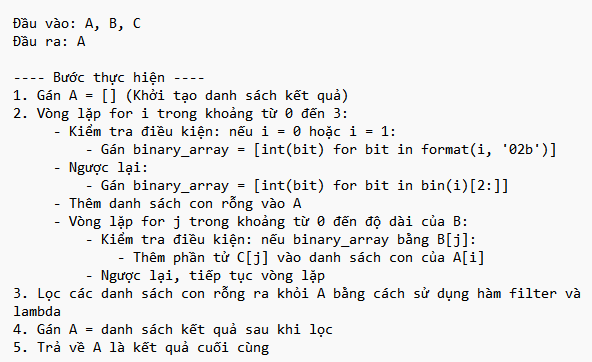
\*Nếu kết quả so sánh = True : Nạp dữ liệu vào file excel output và kết thúc chương trình.

*4.2. Lưu đồ giải thuật các chương trình con:*

*4.2.1. Lưu đồ giải thuật gom nhóm TTHT theo Output:*

- Đê thực hiện gom nhóm TTHT cũng như TTKT dựa theo bảng trạng thái ban đầu. Ta cần tạo một hàm ‘create\_group\_list’.

*4.2.1.1. Giải thuật của hàm ‘create\_group\_list’:*



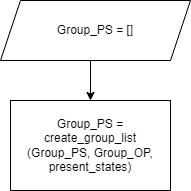
- Trong đó:

+ Mảng A là mảng cần được gom nhóm.

+ Mảng B là mảng được dùng để kiểm tra điều kiện.

+ Mảng C là mảng chứa các phần tử cần gom nhóm.

*4.2.1.2 Sơ đồ khối thực hiện gom nhóm:*



- Trong đó:

+ Group\_PS là group cần phân nhóm

+ Group\_OP là mảng 2 chiều chứa các mảng con, mỗi mảng con có định dạng ‘[output0, output1]’

+ present\_states là mảng chứa các TTHT trong bảng trạng thái ban đầu.

*4.2.1.3. Giải thích:*

Hàm `create\_group\_list` nhận đầu vào là danh sách `A`, `B`, và `C`. Nó tạo ra một danh sách mới `A` dựa trên giá trị của biến `i` trong khoảng từ 0 đến 3 (vì giá trị ngõ ra output là giá trị 2bit đều nằm trong khoảng giá trị ‘[00,01,10,11]’).

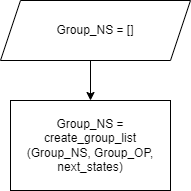
Khi gọi hàm create\_group\_list(Group\_PS, Group\_OP, present\_states) để tiến hành gom nhóm cho Group\_PS. Hàm create\_group\_list sẽ lần lượt duyệt từng giá trị nhị phân của ‘i’ trong mỗi vòng lặp với tất cả giá trị trong Group\_OP. Nếu giá trị của Group\_OP[i] == `bin(i)[2:]` thì hàm sẽ nạp phần tử present\_states[i] vào mảng con Group\_PS[i] của Group\_PS.

Cuối cùng sau khi hoàn thành tất cả công việc hàm sẽ trả về Group\_PS đã được gom nhóm các TTHT có ngõ ra output giống nhau.

*4.2.2. Lưu đồ giải thuật gom nhóm TTKT theo nhóm TTHT:*

Thực hiện gom nhóm TTKT ta cũng thực hiện tương tự như TTHT, chương trình sẽ sử dụng hàm ‘create\_group\_list’ ở trên để thực hiện việc gom nhóm TTKT. Vì khi gom nhóm TTKT chương trình cần nạp nguyên cặp ‘[next\_state0, next\_state1]’ vào mảng con của Group\_NS nên chương trình chỉ duyệt theo 1 chiều dài của mảng next\_states chính vì thế nên ta có thể sử dụng hàm ‘create\_group\_list’ trong trường hợp này.

*4.2.2.1. Sơ đồ khối thực hiện gom nhóm Group\_NS:*



+ Trong đó:

Group\_NS là group cần phân nhóm

Group\_OP là mảng 2 chiều chứa các mảng con, mỗi mảng con có định dạng ‘[output0, output1]’

next\_states là mảng 2 chiều chứa các mảng con, mỗi mảng con có định dạng ‘[next\_state0, next\_state1]’

*4.2.2.2. Giải thích:*

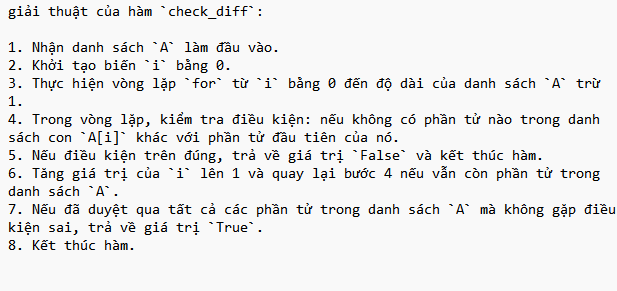
Khi gọi hàm create\_group\_list(Group\_NS, Group\_OP, next\_states) để tiến hành gom nhóm cho Group\_NS. Hàm create\_group\_list sẽ lần lượt duyệt từng giá trị nhị phân của ‘i’ trong mỗi vòng lặp với tất cả giá trị trong Group\_OP. Nếu giá trị của Group\_OP[i] == `bin(i)[2:]` thì hàm sẽ nạp mảng con next\_states[i] vào mảng con Group\_NS[i] của Group\_NS.

Cuối cùng sau khi hoàn thành tất cả công việc hàm sẽ trả về Group\_NS đã được gom nhóm các cặp TTKT có ngõ ra output giống nhau.

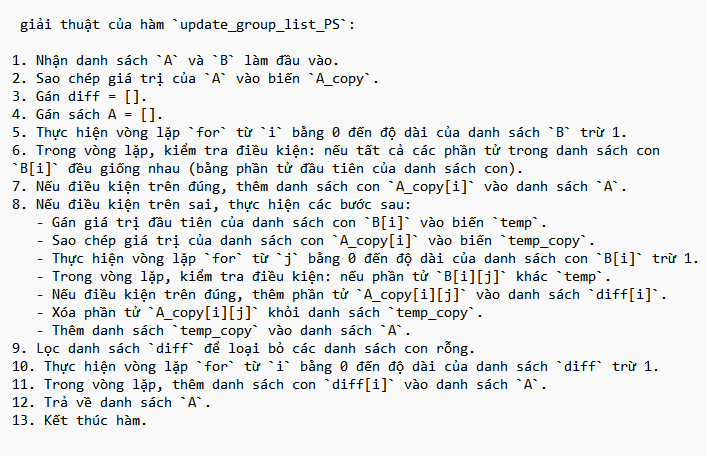
*4.2.3. Lưu đồ kiểm tra điều kiện và phân hoạch lại*

*4.2.3.1. Giải thuật các hàm liên quan trong lưu đồ:*

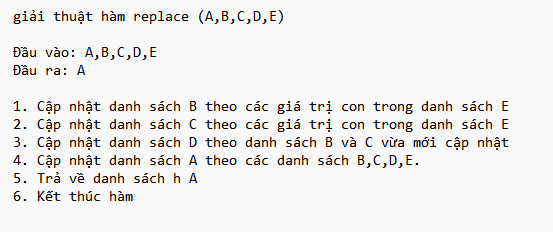
+ hàm ‘check\_diff(A)’ :



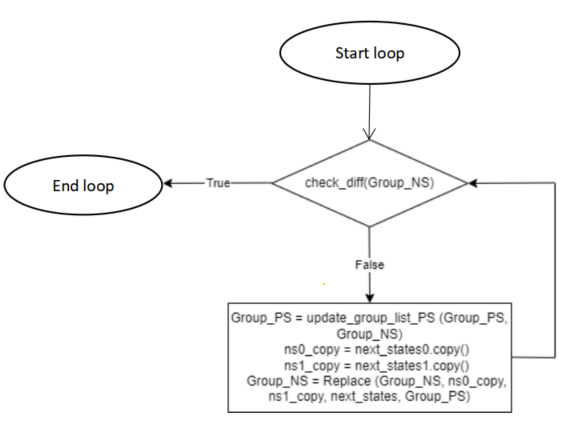
+ Hàm ‘update\_group\_list\_PS (A, B)’:



+Hàm ‘Repalce (A,B,C,D,E)’:



*4.2.3.2. Sơ đồ giải thuật vòng lặp kiểm tra điều kiện và phân hoạch lại:*



*4.2.3.3. Giải thích:*

- Sau khi cập nhật các nhóm Group\_NS, Group\_PS dựa vào bảng trạng thái ban đầu. Chương trình sẽ thực hiện một vòng lặp để kiểm tra xem còn mảng con nào còn chứa các phần tử khác nhau trong mảng con hay không.

- Nếu kết quả điều kiện kiểm tra là False:

+ Đầu tiên chương trình sẽ thực hiện cập nhật lại nhóm Group\_PS bằng hàm ‘update\_group\_list (Group\_PS, Group\_NS)’. Khi hàm ‘update\_group\_list (Group\_PS, Group\_NS)’ được gọi, hàm sẽ tự động tạo một mảng trống sau đó nó sẽ thực hiện duyệt xem trong một mảng con của Group\_NS, những phần tử nào khác với phần tử đầu tiên đó. Vì Group\_NS và Group\_PS lúc này có cùng kích thước nên những phần tử trong Group\_PS tương ứng với các phần tử giống nhau trong Group\_NS sẽ được thêm vào mảng trống và các phần tử có giá trị TTKT khác với các phần tử còn lại cũng được thêm vào một mảng diff []. Sau khi quá trình duyệt kết thúc các phần tử trong mảng diff cũng sẽ được thêm vào Group\_PS như một nhóm mới.

+ Tiếp theo chương trình sẽ tiến hành tạo 2 mảng độc lập mang giá trị next\_state0 và next\_state1 của TTKT trong bảng trạng thái ban đầu là ns0\_coppy và ns1\_coppy.

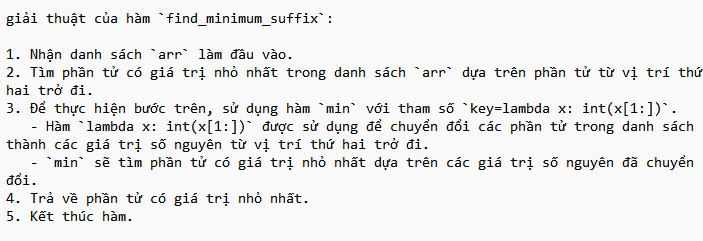
+ Sau cùng chương trình sẽ cập nhật lại Group\_NS theo các thông số đã được cập nhật trước của vòng lặp.

- Nếu kết quả là True: kết thúc vòng lặp.

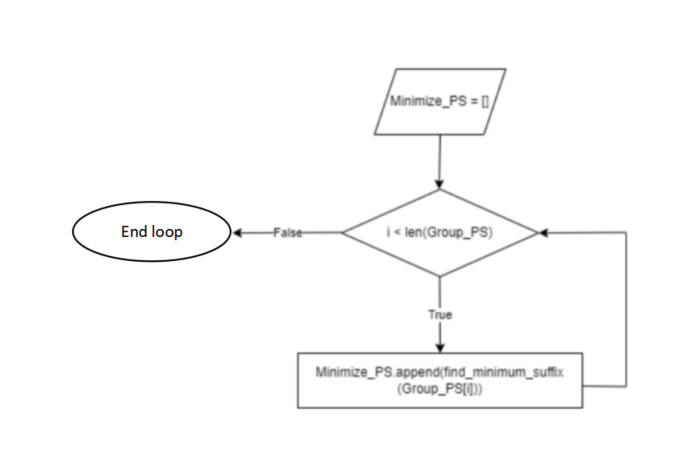
*4.2.4. Lưu đồ giải thuật rút gọn trạng thái trong nhóm tương đương:*

*4.2.4.1. Hàm liên quan trong lưu đồ: (dùng để chọn ra phần tử có hâu tố nhỏ nhất trong nhóm tương đương):*

- Hàm ‘find\_minimum\_suffix (arr)’:



*4.2.4.2. Lưu đồ giải thuật rút gọn trạng thái trong nhóm tương đương:*



*4.2.4.3. Giải thích:*

Sau khi kết thúc vòng while và đảm bảo tính đúng đắn của nhóm Group\_NS. Chương trình sẽ thực hiện vòng lặp for nhằm duyệt từng nhóm tương đương của Group\_PS. Mỗi nhóm sẽ chọn ra phần tử có hậu tố nhỏ nhất và phần tử đó sẽ được thêm vào danh sách Minimize\_PS.

*4.3. Source code chương trình*:

*4.3.1. Hướng tiếp cận chương trình:*

*https://github.com/QuangNhatvo/DoAnCuoiKy*

**CHƯƠNG 5: HIỆN THỰC VÀ KẾT QUẢ.**

*5.1. Điều kiện thực nghiệm:*

- Điều kiện thực nghiệm phải thỏa mãn ít nhất 2 yêu cầu sau:

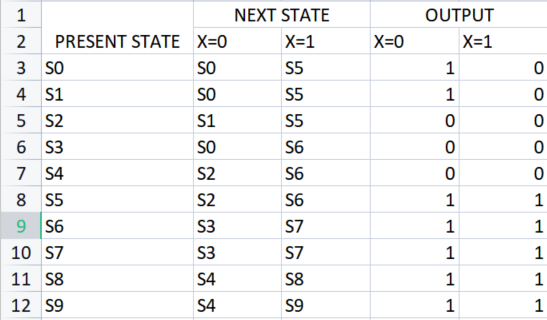
+ Tên trạng thái phải được đặt theo qui định là tiền tố là một chữ cái bất kỳ trong bảng chữ cái và hậu tố phải là một số nguyên

+Số lượng trạng thái xây dựng khi thực hiện chạy chương trình trên các máy tính thông thường nên nhỏ hơn 1000 trạng thái (chương trình có thể chạy đến 10000 trạng thái nhưng quá trình thực hiện khá lâu)

*5.2. Kết quả thu được:*

*5.2.1. Kết quả thu được khi thực nghiệm bảng trạng thái đầu vào có số trạng thái nhỏ hơn 100.*

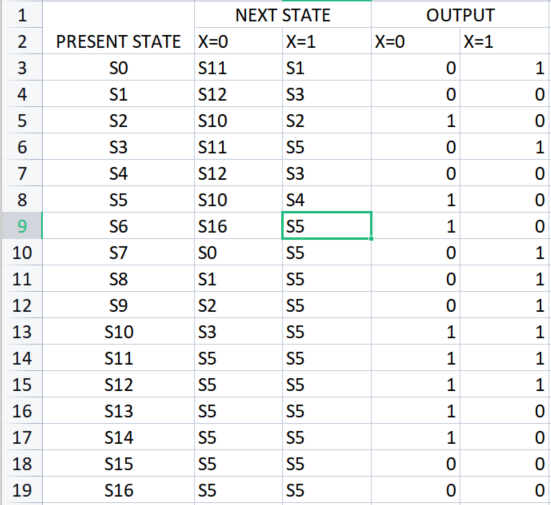
1. Testcase 1:



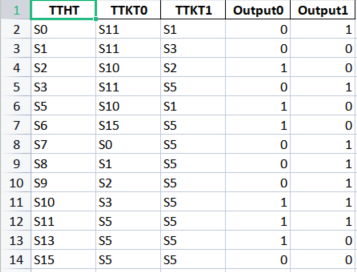
Result:



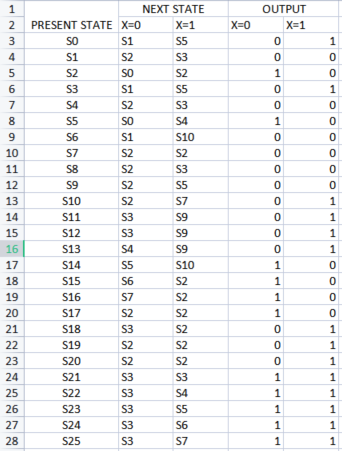
1. Testcase 2:



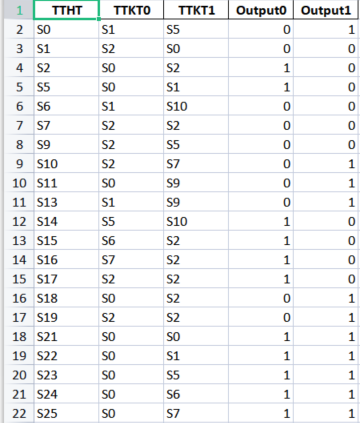
Result:



1. Testcase 3:



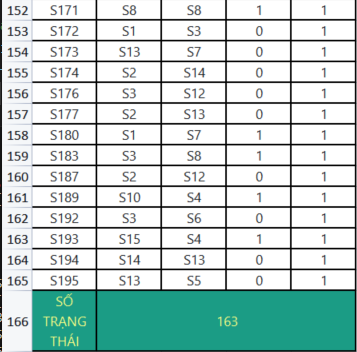
Result:



*5.2.3. Kết quả thu được khi thực nghiệm bảng trạng thái đầu vào có số trạng thái lớn hơn 100*

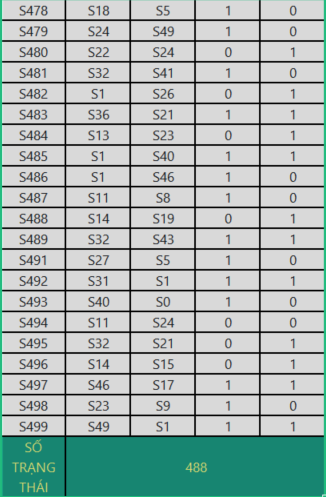
1. Testcase 4: file:\Testcase\autoTest1.xlsx

Result: file:\Result\Result\_autotest\Result1.xlsx



1. Testcase 5: file:\Testcase\autoTest6.xlsx

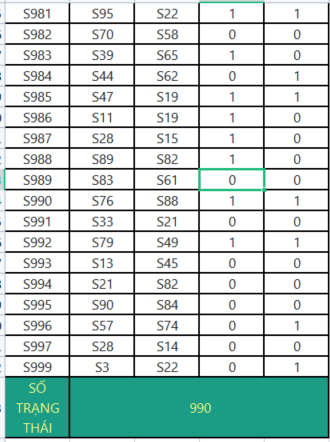
Result: file:\Result\Result\_autotest\Result6.xlsx



*5.2.3. Kết quả thu được khi thực nghiệm bảng trạng thái đầu vào có số trạng thái là 1000:*

1. Testcase 6: file:\Testcase\autoTest11.xlsx

Result: file:\Result\Result\_autotest\Result11.xlsx



CHƯƠNG 6: KẾT LUẬN.

Sau khi kết thúc đồ án rút gọn bảng trạng thái, có thể rút ra các kết luận sau:

1. Hiệu quả của phương pháp rút gọn: Việc áp dụng phương pháp rút gọn bảng trạng thái đã giúp giảm số lượng trạng thái trong bảng, từ đó làm cho hệ thống trở nên đơn giản hơn và dễ quản lý hơn. Việc giảm số lượng trạng thái cũng có thể giảm độ phức tạp của việc phân tích và kiểm thử hệ thống.

2. Tối ưu hóa hiệu suất: Rút gọn bảng trạng thái cũng có thể tối ưu hóa hiệu suất của hệ thống. Bằng cách loại bỏ các trạng thái không cần thiết, ta có thể giảm thời gian và tài nguyên tính toán cần thiết cho việc xử lý và quản lý trạng thái.

3. Độ tin cậy: Tùy thuộc vào phương pháp rút gọn được áp dụng, độ tin cậy của hệ thống có thể được cải thiện. Việc loại bỏ các trạng thái trùng lặp, không cần thiết hoặc không quan trọng có thể giúp giảm khả năng xảy ra lỗi và tăng tính ổn định của hệ thống.

4. Áp dụng thực tế: Kỹ thuật rút gọn bảng trạng thái là một phương pháp quan trọng và được áp dụng trong nhiều lĩnh vực, như hệ thống điều khiển, mô hình hóa và mô phỏng, kiểm thử phần mềm và công nghệ thông tin. Đồ án đã cung cấp một cách tiếp cận để thực hiện việc rút gọn bảng trạng thái trong môi trường lập trình Python.

5. Học tập và nâng cao kỹ năng: Đồ án rút gọn bảng trạng thái không chỉ đưa ra một giải pháp thực tế cho việc tối thiểu hóa trạng thái, mà còn cung cấp cơ hội học tập và nâng cao kỹ năng lập trình và phân tích hệ thống. Việc thực hiện đồ án đã giúp củng cố kiến thức về lý thuyết và thực hành trong lĩnh vực này.

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN