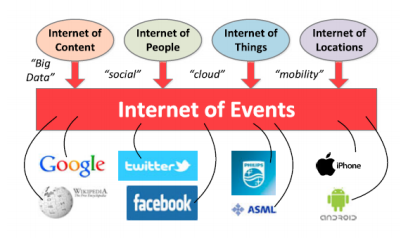
**Khai phá quy trình**

**5.1. Giới thiệu về Khai phá quy trình (Process Mining)**

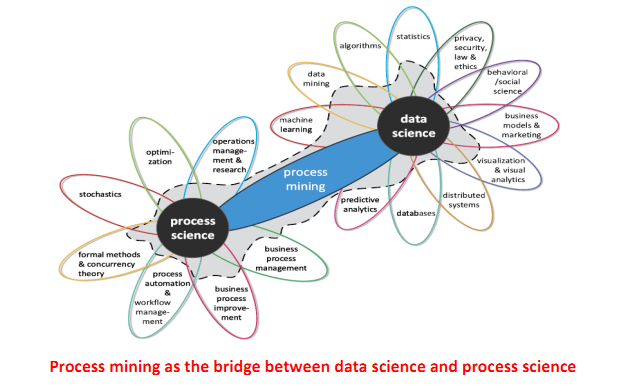
- Khai phá quy trình là một lĩnh vực nghiên cứu chú trọng vào phân tích các quy trình dựa trên các dữ liệu sự kiện (event data)



- Khai phá quy trình là một cầu nối giữa Khoa học Dữ liệu (Data Science) và Khoa học Quy trình (Process Science)

+ Khoa học dữ liệu: học máy, khai phá dữ liệu, thuật toán, thống kê, bảo mật – an ninh – luật và đạo đức, khoa học hành vi/xã hội, mô hình kinh doanh và tiếp thị, trực quan hóa và phân tích trực quan, hệ thống phân tán, phân tích dự báo

+ Khoa học quy trình: quản lý quy trình nghiệp vụ (QTNV), cải tiến QTNV, tự động quy trình và quản lý dòng công việc, phương pháp hình thức và lý thuyết tương tranh, ngẫu nhiên, tối ưu hóa, quản lý điều hành và vận trù học



**Mạng Petri**

**5.2. Mạng Petri**

- Mạng Petri là một công cụ minh họa (một đồ thị phân đôi gồm các trạng thái và chuyển tiếp) cho việc diễn tả và phân tích các quy trình tương tranh trong hệ thống nhiều thành phần (hệ thống phân tán)

**5.2.1. Hệ thống chuyển tiếp trạng thái**

**5.2.1.1. Định nghĩa**

*Định nghĩa 5.1.* (Hệ thống chuyển tiếp hay Hệ thống chuyển tiếp trạng thái)

Một hệ thống chuyển tiếp trạng thái là một bộ ba trong đó

- là một tập các trạng thái

- là một tập các hoạt động (thông thường là hành động)

- là tập các chuyển tiếp

- Chú ý:

+ là tập các trạng thái khởi tạo

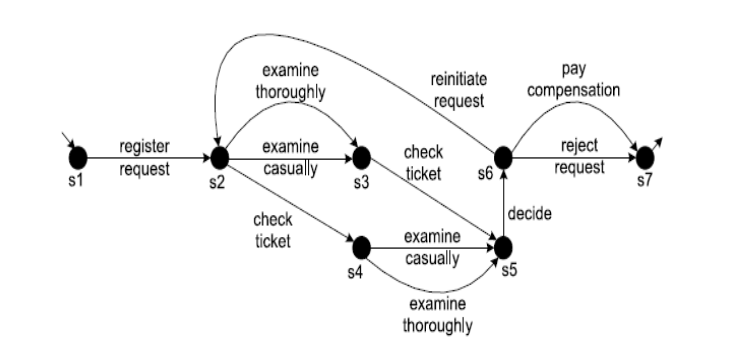
+ là tập các trạng thái kết thúc

- Trong đa số các trường hợp, tập hợp trạng thái là hữu hạn. Khi đó, hệ thống chuyển tiếp trạng thái còn gọi là Máy hữu hạn trạng thái hay Automata hữu hạn (FA)

**5.2.1.2.** **Hành vi của hệ thống chuyển tiếp trạng thái**

- Bắt đầu tại một trạng thái khởi tạo và lần lượt di chuyển qua các đỉnh dựa trên hành động

- Một đường đi kết thúc thành công khi nó dừng lại tại một trong các trạng thái kết thúc. Ngược lại, một đường đi deadlock khi nó dừng tại một trạng thái không phải kết thúc mà không có đường đi khỏi đó



Hình 5.2. Hệ thống chuyển tiếp trạng thái

Ví dụ 5.1: Quan sát một hệ thống chuyển tiếp trạng thái trong hình 5.2. Hệ thống này bao gồm:

- Tập trạng thái . Trong đó,

- Tập các hoạt động register request, examine casually, examine thoroughly, check ticket, reinitiate request, decide, reject request, pay compensation}

- Tập các sự chuyển tiếp

**5.2.1.3. Vấn đề khi sử dụng hệ thống chuyển tiếp**

- Hệ thống chuyển tiếp tuy đơn giản nhưng lại gặp vấn đề trong việc diễn tả tương tranh một cách ngắn gọn, hay còn gọi là ‘sự bùng nổ trạng thái’. Nhưng mạng Petri có thể sử dụng để giải quyết điều này một cách chặt chẽ và có hiệu quả

- Ví dụ: Cho n hoạt động song song. Tất cả n hoạt động này cần phải thực thi, bất kỳ trình tự nào cũng được chấp thuận

Tổng cộng có tất cả luồng thực thi. Khi đó, hệ thống chuyển tiếp sẽ cần có trạng thái và chuyển tiếp

Khi , số trạng thái là 1024 và số sự chuyển tiếp là 5120. Đây là một con số tương đối lớn

Trong khi đó mạng petri chỉ cần 10 chuyển tiếp và 10 vị trí để mô hình hóa hệ thống trên

**5.2.2.** **Đa tập hợp:**

- Cho miền và ánh xạ xác định một đa tập hợp trên như sau: Với mỗi là số lần xuất hiện trong đa tập hợp

Hiển nhiên và chúng ta có thể dùng định dạng sau cho

Vì vậy, được nhận dạng bởi danh sách . Ở đây, tần số , có nghĩa là không xuất hiện trong .

chứa các phần tử khác nhau trong đa tập hợp

**5.2.3. Mạng Petri**:

Định nghĩa 5.2. Một mạng Petri là một bộ ba trong đó:

- là tập hữu hạn các vị trí

- là tập hữu hạn các chuyển dịch sao cho

- là tập các cung có hướng, gọi là luồng quan hệ

1. Một token là một nút chuyển tiếp đặc biệt, được biểu diễn bởi một chấm đen

Token đại diện cho các phần tử trong thế giới thực. Vị trí có thể chứa token nhưng chuyển tiếp thì không thể chứa token

2. Một chuyển tiếp là tích cực nếu mọi vị trí vào của nó đều chứa token

Một chuyển tiếp tích cực có thể “firing” bằng cách tiêu thụ một token ở các vị trí vào của nó

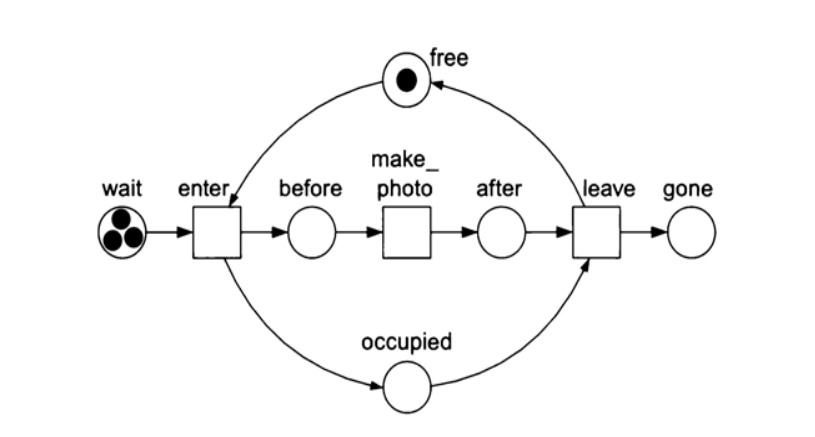
3. Một bộ đánh dấu là một phân bố của các token tại các vị trí. Một bộ đánh dấu của mạng N là một hàm , gán cho mỗi vị trí một số token tại vị trí này

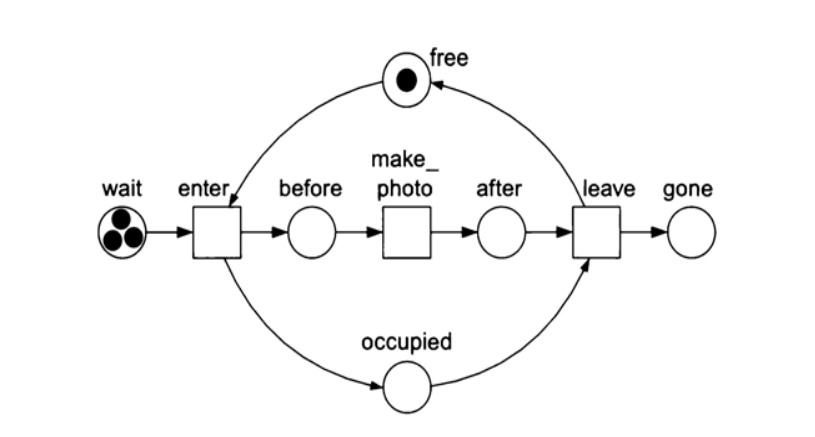
Kí hiệu, miền của ánh xạ m, được xem như một đa tập hợp

4. Một mạng Petri đánh dấu là một cặp , trong đó là mạng Petri và là một đa tập hợp trên , được kí hiệu bởi bộ đánh dấu của mạng

**Tập hợp tất cả các đa tập hợp trên là hoặc**

**Tập hợp tất cả mạng Petri đánh dấu là**



Ví dụ 5.3.

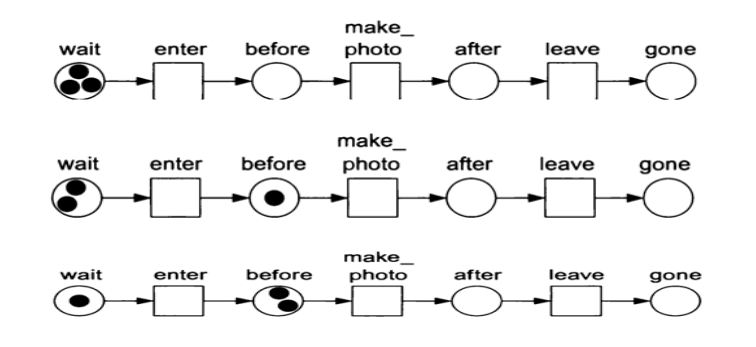
- Các thành phần của mạng Petri ở trên:

+ 3 chuyển tiếp

+ 6 vị trí

+ Các luồng quan hệ

Ví dụ 5.4.



- Mạng petri trên gồm có:

+ 3 chuyển tiếp

+ 4 vị trí

+ 3 bộ đánh dấu của mạng trên là:

Định nghĩa 5.3. (Đầu vào là vị trí, đầu ra là chuyển tiếp)

Cho là một mạng Petri. Các phần tử của gọi là node

1. Một node là node vào của một node nếu và chỉ nếu có một cung đi từ đến

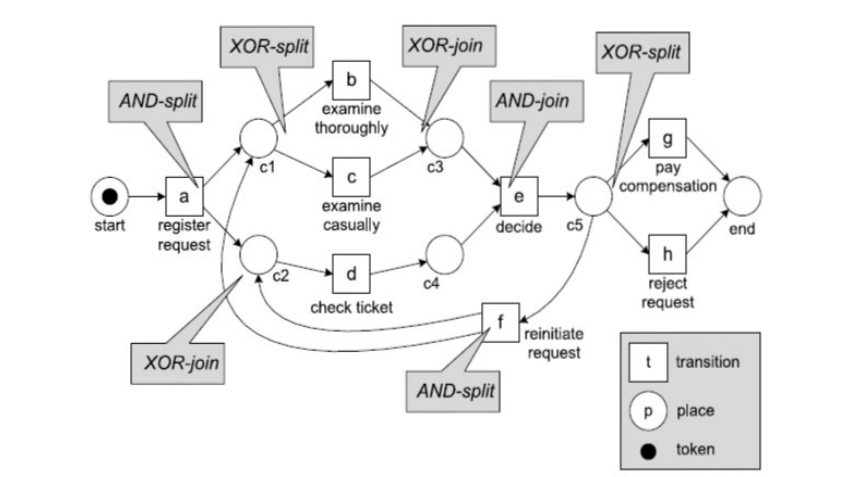
. Ngược lại, node là node ra của node nếu và chỉ nếu có một cung đi từ đến

2. Với mọi , kí hiệu là tập vào của và

là tập ra của

3. Với mọi tập , định nghĩa là tập một chuỗi các phần tử của

Ví dụ: Với và :



- Ở hình trên ta có:

+

+

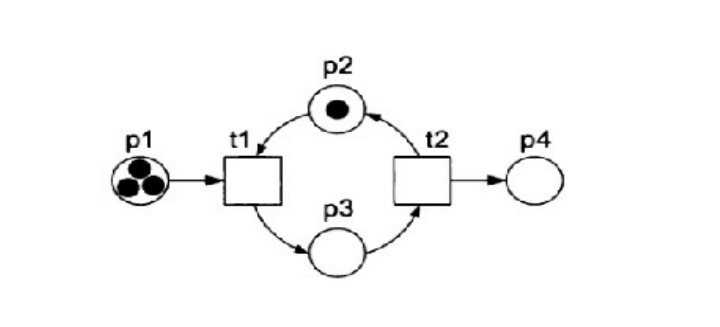
**5.2.4. Kích hoạt chuyển tiếp tích cực và thay đổi bộ đánh dấu:**

*- Một chuyển tiếp gọi là tích cực nếu ở mỗi vị trí vào của nó đều có ít nhất một token*

*- Các chuyển tiếp có thể tiêu thụ token ở mỗi vị trí vào để tạo ra một token ở mỗi vị trí ra*

*- Các chuyển tiếp có thể tái kích hoạt việc tiêu thụ và sản xuất token cho đến khi hết token ở ít nhất một trong các vị trí vào (chuyển tiếp chuyển thành không tích cực)*

*- Các chuyển tiếp có thể đồng thời tiêu thụ token*



- Mạng Petri trên bao gồm:

+ Các vị trí

+ Các chuyển tiếp

+ Các luồng quan hệ

- Tập vào và tập ra cho mỗi chuyển tiếp là:

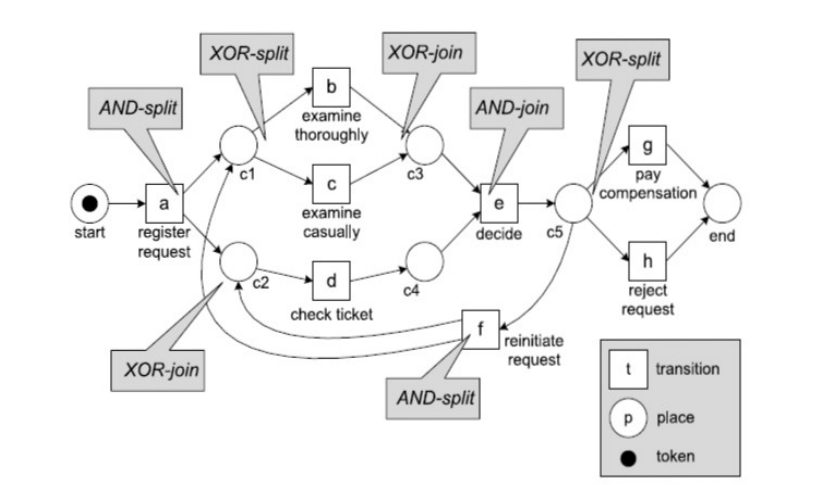
+

+

- Bộ đánh dấu ban đầu của mạng Petri trên là:

- Trong mạng Petri trên, tích cực, không tích cực

**5.2.5. Nguyên lý thứ tự thao tác:**



Có 5 nguyên lý cơ bản sau:

- Sequence: sắp xếp các thao tác theo một danh sách tuyến tính

- Or-split: chọn một nhánh để thực thi

- And-split: tất cả các nhánh sẽ được thực thi

- Or-join: một trong các nhánh vào phải sẵn sàng để tiếp tục

- And-join tất cả các nhánh vào phải sẵn sàng để tiếp tục

**Tổng kết**

**1. Vị trí và chuyển tiếp của mạng Petri**

**2. Tính chủ động và bị động của vị trí và chuyển tiếp**

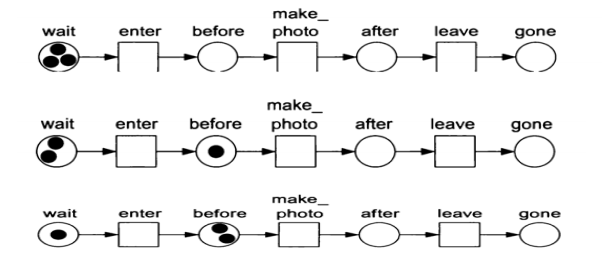
**3. Cung - Quan hệ nhị phân**

**4. Gán nhãn cho các cung và chuyển tiếp**

**5. Mô hình hóa bài toán thành mạng Petri**

**Vận dụng 1.**

Cho một quy trình của máy X-ray



1. Xác định hai quan hệ và , và tập luồng quan hệ

- Tập vị trí

- Tập chuyển tiếp

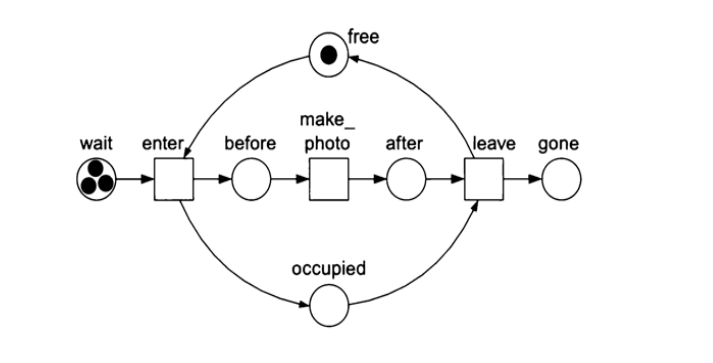
- Tập cung vào

- Tập cung ra

- Tập luồng quan hệ

2. Một bệnh nhân có thể vào phòng X-Ray chỉ sau khi bệnh nhân trước đó rời khỏi phòng. Chúng ta cần đảm bảo rằng hai vị trí **before** và **after** sẽ không cùng chứa nhiều hơn một token

Có hai trạng thái khả thi: phòng X-Ray có thể **free** hoặc **occupied**. Chúng ta sẽ mô hình điều này bằng cách thêm vào hai vị trí như sau:



Với mạng Petri này, vị trí **before** có thể chứa nhiều hơn 1 token được không ? Vì sao ? Xác định lại tập vị trí

- Vị trí **before** không thể chứa nhiều hơn token bởi vì:

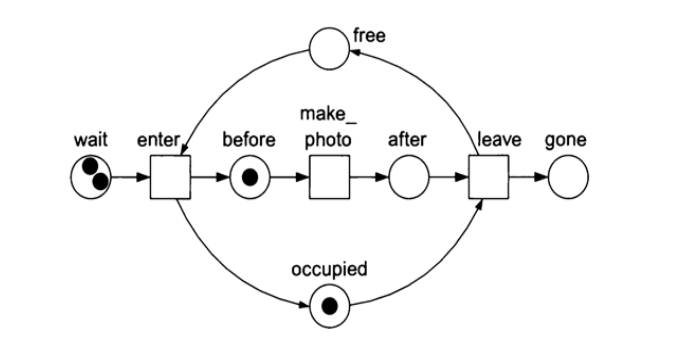
+ Ở trạng thái đầu, vị trí **enter** tích cực. Do đó **enter** sẽ tiêu thụ một token ở **wait** và **free** để sản xuất 1 token cho **before**. Bộ đánh dấu lúc này là

+ Khi đó, vị trí **enter** sẽ không tích cực. Nên không thể tiếp tục tiêu thụ token. Trong lúc đó, token ở **before** có thể được tiêu thụ bởi **make\_photo**. Sau đó, lại được tiêu thụ bởi **leave**. Bộ đánh dấu lúc này là

+ Cứ tiếp tục như vậy, số token tại **before** sẽ không bao giờ lớn hơn 1

- Tập vị trí

3. Khi vị trí **free** không có token, chuyển tiếp **enter** có thể kích hoạt lại lần nữa không ? Vì sao ? Thiết lập hai quan hệ và ?



- Vị trí **enter** có thể được kích hoạt lại. Bởi vì:

+ Đầu tiên, chuyển tiếp **make\_photo** sẽ kích hoạt. Bộ đánh dấu:

+ Sau đó, chuyển tiếp **leave** sẽ kích hoạt bởi vì các vị trí vào của nó (**after** và **occupied**) đã chứa token. Bộ đánh dấu:

+ Lúc này, chuyển tiếp **enter** đã có thể kích hoạt trở lại vì các vị trí vào của nó (**wait** và **free**) đã chứa token

5.1. Process Mining

5.2. Background

5.2.2. Formal Definition

5.2.3. On enabled transition and Marking changes

5.2.4. Important usage of Petri Net via explaining figure 5.5

5.3. Behavior of Petri Net

5.3.1. Firing, Reachability to Labeled Petri Net

5.3.2. Respresenting Petri Net as Special Transition System

5.4. Structure and Basic Problem

5.4.1. Causality, concurrency and sychronization

5.4.2. Effect of Concurrency

5.5. Review Problem

5.6. Assignment on Modelling

5.6.1. Essential notion for Modelling with Petri Net

5.6.2. Dynamic of Petri Nets via Special properties

5.6.3. Modeling by Petri Networks – Problem

5.6.4. Practical assignment

**5.3. Hành vi của mạng Petri**

**5.3.1. Từ “firing”, khả năng truy cập đến mạng Petri dán nhãn:**

- Một token là một chấm đen trong mạng Petri

Định nghĩa 5.5. Luật “firing”

- Cho là mạng Petri đánh dấu với và

1. Một chuyển tiếp gọi là tích cực nếu có ít nhất một token ở mỗi vị trí vào của nó

2. Chuyển tiếp tích cực tại bộ đánh dấu , kí hiệu , nếu và chỉ nếu

3. Luật “firing” là quan hệ nhỏ nhất thỏa mãn:

với mọi và với mọi

*Quan sát 1.*

1. Vị trí có thể bao gồm token, chuyển tiếp thì không. Nhưng chuyển tiếp có thể thay đổi bộ đánh dấu bằng cách tiêu thụ token. Chính vì vậy mà vị trí thụ động còn chuyển tiếp thì chủ động

*Để “fire” một chuyển tiếp thì nó phải tích cực*

2. **Sự tích cực**: Một bộ đánh dấu , có dạng

vì vậy có thể nói tích cực tại M nếu và chỉ nếu

3. Một chuyển tích tích cực có thể “fire”, khi đó bộ đánh dấu biến thành . Nếu một chuyển tiếp tích cực “fire”, nó tiêu thụ một token từ mỗi vị trí vào và sản xuất một token tại mỗi vị trí ra.

4. nghĩa là chuyển tiếp tích cực tại bộ đánh dấu

5. nghĩa là chuyển tiếp tích cực t “fire” biến thành

Định nghĩa 5.6. (Chuỗi Firing)

Cho là một mạng Petri đánh dấu với

1. Chuỗi gọi là chuỗi “firing” của **nếu và chỉ nếu,** với một số số nguyên , tồn tại các bộ đánh dấu và các chuyển tiếp sao cho

* , và

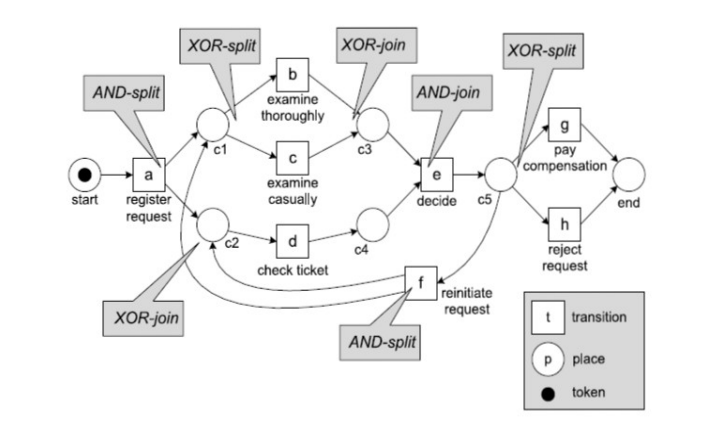
2. Một bộ đánh dấu gọi là có khả năng truy cập từ bộ đánh dấu nếu và chỉ nếu tồn tại một chuỗi các chuyển tiếp tích cực mà khi chúng “firing” thì dẫn từ đến

Tập các bộ đánh dấu có khả năng truy cập của ký hiệu là

3. (Hệ thống mạng Petri) Một hệ thống mạng Petri bao gồm mạng Petri và một bộ đánh dấu phân biệt – Bộ đánh dấu ban đầu

Ví dụ 5.6. Trong hình bên dưới, bộ đánh dấu

Ta có mạng Petri đánh dấu



- Chuỗi rỗng cũng được xem là một chuỗi “firing” của

- Chuỗi cũng tích cực trong . Firing sẽ thu được bộ đánh dấu .

Do đó, có thể viết hoặc

- Chuỗi là một chuỗi “firing” khả thi và

- Chuỗi là một chuỗi “firing” khả thi và

- Tập các bộ đánh dấu có khả năng truy cập của mạng Petri trên là

Định nghĩa 5.7. Mạng Petri dán nhãn

Thông thường, các chuyển tiếp được định danh bằng một chữ cái, nhưng cũng có thể có các nhãn dài hơn để diễn tả hành động liên quan. Một **mạng Petri dán nhãn** với trong đó là mạng Petri đã được định nghĩa

* là một tập các hành động gán nhãn, và ánh xạ là một hàm dán nhãn
* Sử dụng nhãn cho một số các hành động đặc biệt, gọi là vô hình. Một chuyển tiếp với gọi là *không thể quan sát, vô hình*

**Quan sát 2.**

1. Ví dụ X-ray ở Practical Problem 1 diễn tả rằng có thể đi qua một vài bộ đánh dấu trong mạng Petri bằng một chuỗi các firing

Các chuyển tiếp sẽ tiếp tục firing cho đến khi mạng Petri không còn bất kì chuyển tiếp tích cực nào. Khi đó, ta sẽ thu được bộ đánh dấu dừng (bộ đánh dấu kết thúc)

2. Có thể chuyển đổi bất kì mạng Petri nào thành mạng Petri gán nhãn. Điều ngược lại không phải lúc nào cũng khả thi bởi vì có thể có một số chuyển tiếp có cùng nhãn

**Question 5.1.** Trong việc đánh dấu mạng Petri, khi chúng ta mô hình hệ thống bởi một hệ thống mạng Petri , khi đó sẽ xảy ra một số vấn đề

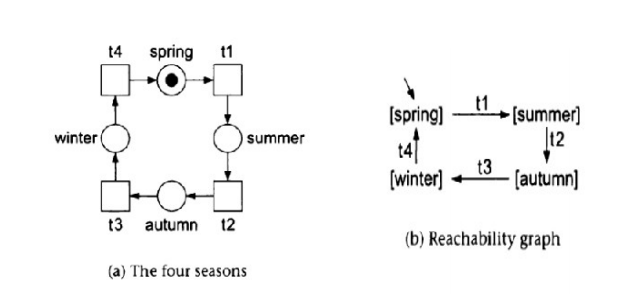
1. Có bao nhiêu bộ đánh dấu có khả năng truy cập

2. Bộ đánh dấu nào có thể truy cập được

3. Liệu rằng sẽ có một bộ đánh dấu kết thúc hay không ?

Chúng ta đã biết bộ đánh dấu bắt đầu của hệ thống đã cho. Các câu hỏi trên sẽ được trả lời bằng cách tìm tập các bộ đánh dấu có thể truy cập từ . Biểu diễn tập này thành một đồ thị - Đồ thị truy cập của mạng Petri. Mỗi node là một bộ đánh dấu có thể truy cập và mỗi cạnh là một chuyển tiếp chuyển từ bộ đánh dấu này sang bộ đánh dấu khác

**Ví dụ 5.7.** Xem xét mạng Petri mô hình cho bốn mùa như sau



Mỗi bộ đánh dấu có thể truy cập được biểu diễn dưới dạng đa tập hợp. Đa tập hợp được biểu diễn là bộ đánh dấu bắt đầu trong hình a

* Cạnh vào mà không có đỉnh nguồn cho thấy bộ đánh dấu đó là bộ đánh dấu khởi đầu. Dán nhãn mỗi cạnh của đồ thị truy cập bằng với chuyển tiếp fire tới bộ đánh dấu kế tiếp
* Hình b là đồ thị đầy đủ. Có thể thấy được có 4 bộ đánh dấu có thể truy cập
* Nếu một bộ đánh dấu có thể truy cập từ thì khi đó trong đồ thị truy cập sẽ có một đường đi từ đến . Đường đi này biểu diễn một chuỗi các firing dẫn đến từ ban đầu

Một chuỗi các chuyển tiếp gọi là *run*. Một *run* là hữu hạn nếu đường đi và chuỗi chuyển tiếp là hữu hạn. Mặt khác *run* là vô hạn

Đồ thị ở trên không có *run* vô hạn

**5.3.2. Biểu diễn mạng Petri thành hệ thống chuyển tiếp đặc biệt:**

Định nghĩa 5.8. (Đồ thị truy cập)

Cho với là một mạng Petri dán nhãn

định nghĩa một hệ thống chuyển tiếp với

và

hoặc với nhãn

là **đồ thị truy cập** của

**Tổng kết**

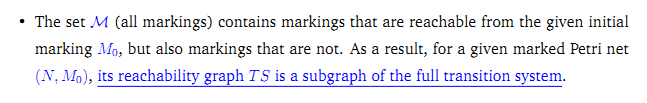
**1. Sự chuyển đổi khái niệm trong mạng Petri và hệ thống chuyển tiếp**

**(label 🡪 action, (initial) marking 🡪 (initial) state)**

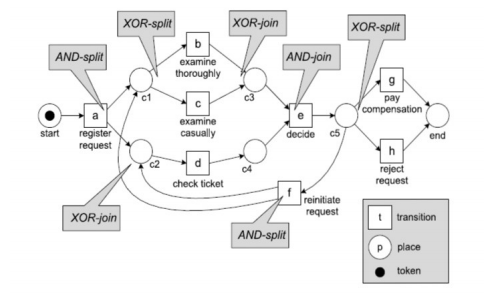
**2. Tập TR**

**3. Quá trình mô hình hóa tập TR**

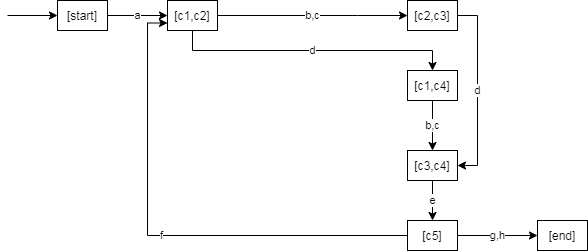
**4. Đồ thị truy cập là đồ thị con của hệ thống chuyển tiếp đầy đủ**

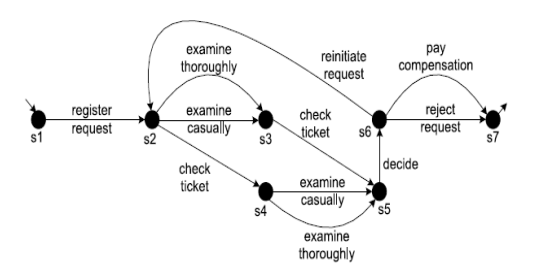


**Practice 5.2.** *Xây dựng hệ thống chuyển tiếp TS được tạo ra từ mạng petri đánh dấu dán nhãn sau*



- Đồ thị truy cập của mạng Petri trên là:





**Chú ý:** là tập chứa bộ đánh dấu khởi đầu của mạng Petri. Mạng Petri không định nghĩa cụ thể tập các bộ đánh dấu kết thúc

Tuy nhiên, trong trường hợp này,

**5.4. Mạng Petri – Cấu trúc và các vấn đề cơ bản**

- Vị trí và chuyển tiếp đường kết nối bởi cung. Sự kết nối của các node quy định hành vi của mạng. Một cách trang trọng, cách mà chuyển tiếp kết nối quy định thứ tự mà chúng kích hoạt (fire). Đầu tiên, nhắc lại các luật và các ý tưởng liên quan trong một mạng Petri

1. Khi một chuyển tiếp fire, tổng số token bằng số token khởi đầu trừ cho số token tiêu thụ cộng với số token sản xuất

2. Tổng số token trong mạng thay đổi nếu số vị trí vào của không giống với số vị trí ra của . Vì vậy, sự kích hoạt của chuyển tiếp có thể làm tăng hoặc giảm số token hiện có

3. Khi có một vài chuyển tiếp kích hoạt tại cùng một thời điểm, không thể xác định được số nào trong chúng được kích hoạt. Tình huống này là lựa chọn không đơn định ***(nondeterministic choice)***

Mặc dù trong trường hợp này không thể biết được chuyển tiếp nào kích hoạt, nhưng vẫn biết được một trong số chúng đã kích hoạt

**5.4.1. Tính nhân quả, tính đồng thời và tính đồng bộ**

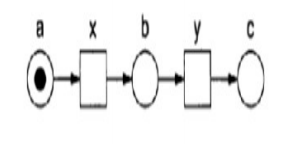
- Trong thực tế, các chuyển tiếp biểu diễn các sự kiện. Giả sử rằng có 3 sự kiện , chúng ta xem xét 3 trường hợp sau:

+ Trường hợp 1: Sự kiện diễn ra trước sự kiện

+ Trường hợp 2: Sự kiện và sự kiện diễn ra đồng thời (trong cùng một thời gian ở bất kì thứ tự nào)

+ Trường hợp 3: Sự kiện diễn ra sau cả hai sự kiện và

1. Tính nhân quả (Trường hợp 1)

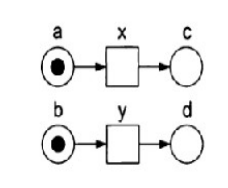


- Nhân quả là mối quan hệ giữa hai sự kiện trong hệ thống cần phải có thứ tự. Trong mạng Petri, có thể biểu diễn mối quan hệ này bằng hai chuyển tiếp kết nối thông qua một vị trí trung gian

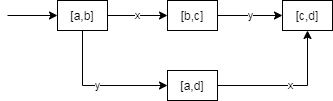
1. Tính đồng thời (Tính song song)

- Đây là một tính năng quan trọng của hệ thống. Trong một hệ thống đồng thời, một vài sự kiện có thể diễn ra cùng một lúc

- Trong hình dưới đây, chuyển tiếp và có thể kích hoạt độc lập với chuyển tiếp còn lại bởi vì không có mối quan hệ nhân quả giữa và . Với một số cấu trúc, ta có thể mô hình hóa sự đồng thời. Thông thường, sau đó sẽ cần có sự đồng bộ



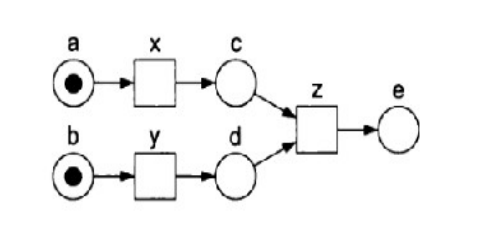
**Question:** Có thể tìm một đồ thị truy cập của mạng ở trên được không



1. Tính đồng bộ

- Có thể mô hình tính đồng bộ trong một mạng Petri bằng một chuyển tiếp với ít nhất hai vị trí vào.

- Trong hình dưới, chỉ được kích hoạt sau khi và kích hoạt



**5.4.2. Tác dụng của sự đồng thời (sự song song)**

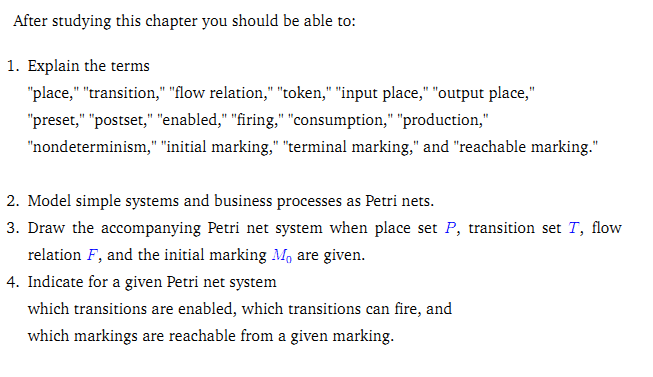
**Question 5.2.**  Có thể lượng hóa sự đồng thời với một tiến trình cho trước trong mạng Petri hay không ?

Câu trả lời là có, công cụ là hệ thống chuyển tiếp.

**- Nếu mô hình có nhiều sự đồng thời hoặc nhiều token trong cùng một vị trí thì hệ thống chuyển tiếp (TS) sẽ lớn hơn rất nhiều so với mạng Petri**

**- Một mạng Petri đánh dấu có thể có vô hạn các trạng thái có thể truy cập**

**5.5. Tổng kết và nhắc lại một số vấn đề**



**Problem 5.1.** Giải thích các từ sau và cho ví dụ minh họa

- Chuyển tiếp tích cực

- “Firing” của một chuyển tiếp

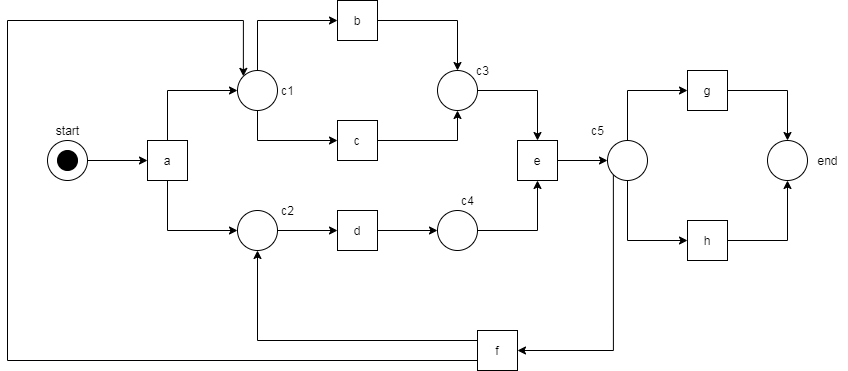
- Bộ đánh dấu có thể truy cập

- Bộ đánh dấu kết thúc

- Lựa chọn không đơn định

- - - - - - - - - - - - - - - - - -

- Lấy ví dụ sau

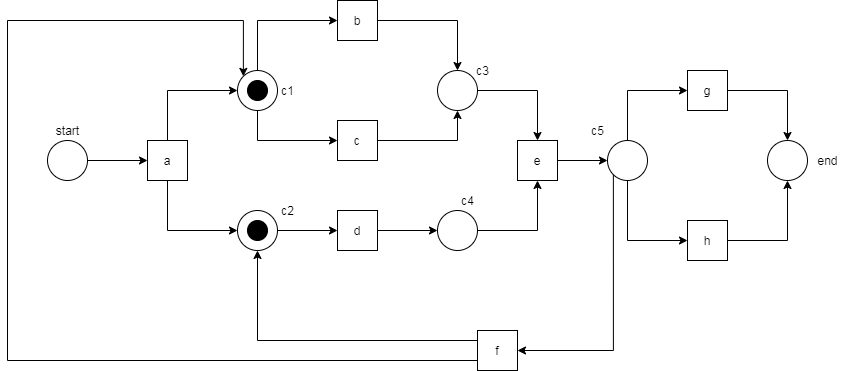


- Chuyển tiếp tích cực là chuyển tiếp mà mọi vị trí vào của nó đều có ít nhất 1 token

Ví dụ: a là chuyển tiếp tích cực

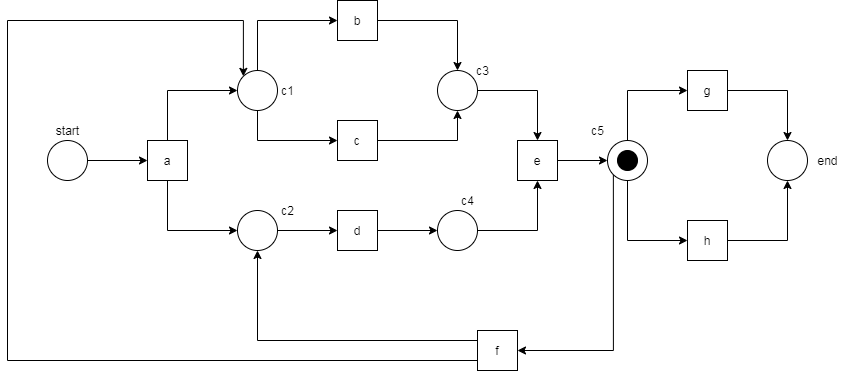
- Firing của một chuyển tiếp là quá trình chuyển tiếp đó tiêu thụ một token ở mỗi vị trí vào và tạo ra một token tại mỗi vị trí ra

Ví dụ: a tiêu thụ token ở vị trí start và tạo ra một token ở vị trí và

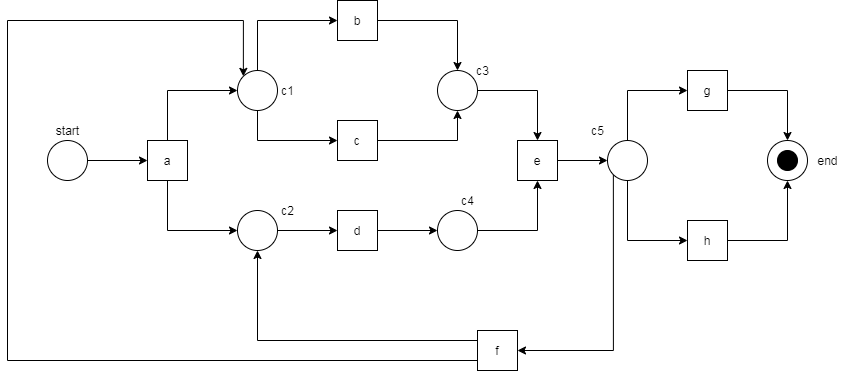


- Một bộ đánh dấu gọi là có khả năng truy cập từ bộ đánh dấu nếu và chỉ nếu tồn tại một chuỗi các chuyển tiếp tích cực mà khi chúng “firing” thì dẫn từ đến

Ví dụ: Bộ đánh dấu có thể truy cập từ bộ đánh dấu ban đầu qua chuỗi kích hoạt

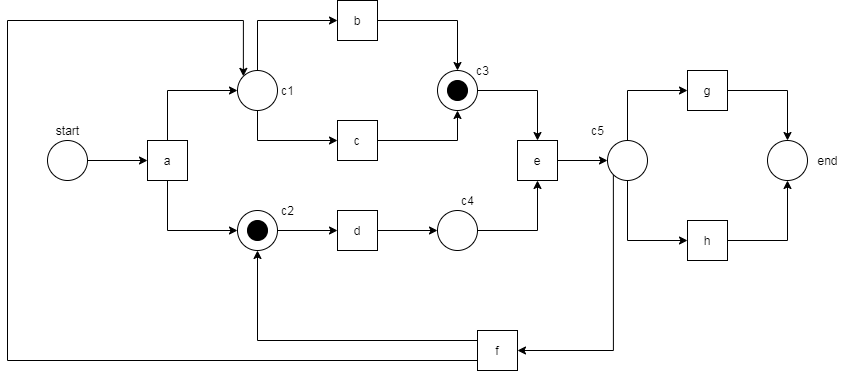


- Bộ đánh dấu kết thúc là bộ đánh dấu mà tại đó mạng Petri không còn bất kì chuyển tiếp tích cực nào

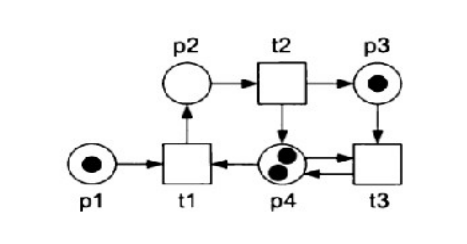
Ví dụ: Bộ đánh dấu là bộ đánh dấu kết thúc, bởi vì tại đó, không còn chuyển tiếp tích cực nào

- Lựa chọn không đơn định xảy ra khi có một vài chuyển tiếp kích hoạt tại cùng một thời điểm, không thể xác định được số nào trong chúng được kích hoạt

Ví dụ: Từ bộ đánh dấu qua bộ đánh dấu , ta không thể xác định được chuyển tiếp hay đã kích hoạt mà chỉ biết mà có ít nhất một chuyển tiếp đã kích hoạt



**Problem 5.2.** Xem xét mạng Petri ở hình dưới



1. Mô hình mạng Petri thành

-

-

-

-

2. Chỉ ra tập vào và tập ra của mỗi chuyển tiếp

-

-

-

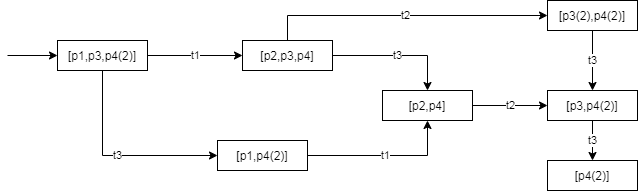
3. Chuyển tiếp nào tích cực tại

- Chuyển tiếp tích cực tại

4. Chỉ ra tất cả các bộ đánh dấu có thể truy cập. Các bộ đánh dấu kết thúc là gì ?

- Có tổng cộng 7 bộ đánh dấu có thể truy cập:

- Bộ đánh dấu kết thúc là vì ở bộ đánh dấu này, không có chuyển tiếp nào ở trạng thái tích cực



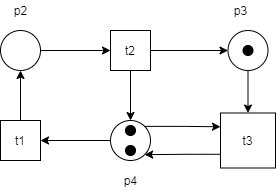
5. Có một bộ đánh dấu có thể truy cập mà có lựa chọn không đơn định hay không ?

- Không có bộ đánh dấu có thể truy cập có lựa chọn không đơn định. Bởi vì, giữa hai bộ đánh dấu truy cập kề nhau, chỉ có duy nhất một cách kích hoạt chuyển tích để chuyển trạng thái của mạng Petri

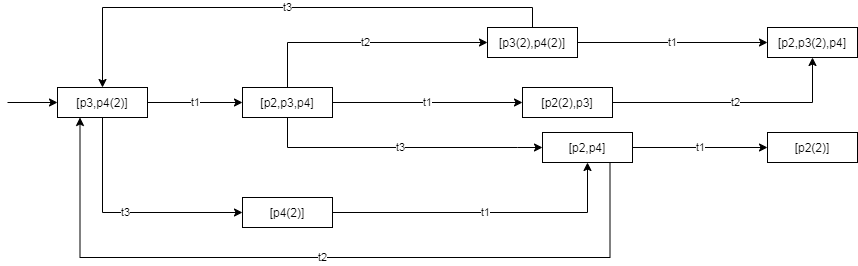
- Ví dụ: Từ bộ đánh dấu qua bộ đánh dấu , chỉ có duy nhất một cách để chuyển trạng thái mạng Petri là kích hoạt chuyển tiếp

6. Số lượng bộ đánh dấu có thể truy cập tăng hay giảm nếu loại bỏ

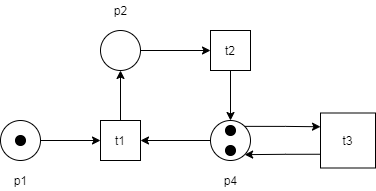
(1) Vị trí và các cung kề



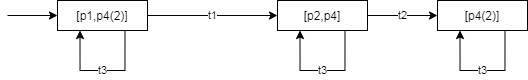
- Số lượng bộ đánh dấu có thể truy cập sẽ tăng lên (Có ít nhất là 8 như hình bên dưới)



(2) Vị trí và các cung kề



- Số bộ đánh dấu có thể truy cập giảm (chỉ có 3 bộ đánh dấu có thể truy cập)

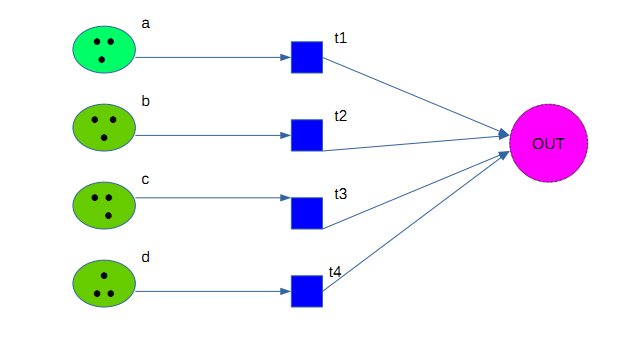


**Problem 5.3.** (Từ một mạng Petri đánh dấu nhỏ đến một hệ thống chuyển tiếp lớn)

Xem xét mạng Petri với , với 4 vị trí bắt đầu được khởi tạo 3 token, và nút kết thúc được kí hiệu là vị trí OUT.

Giả sử có 4 chuyển tiếp , và bộ đánh dấu bắt đầu là , xem ở hình dưới

Kí hiệu cho hệ thống chuyển tiếp được tạo ra từ mạng Petri



1. Xác định của

-

-

-

2. Nếu không cho phép sự đồng thời trong tiến trình này (mạng Petri đánh dấu) thì có bao nhiêu trạng thái của hệ thống chuyển tiếp được tạo ra ? Trong TS, có bao nhiêu chuyển tiếp ?

- Không cho phép đồng thời (interleaving semantic) tức là mỗi cạnh trong hệ thống chuyển tiếp TS chỉ cho phép kích hoạt duy nhất một transition

**\* Tìm số trạng thái của hệ thống chuyển tiếp**

- Bộ đánh dấu ban đầu là

- Ta chứng minh bất kỳ bộ đánh dấu đều có thể truy cập được với

+ Từ bộ đánh dấu khởi đầu , ta sẽ kích hoạt chuyển tiếp lần 🡪 Bộ đánh dấu là

+ Tiếp theo, ta sẽ kích hoạt chuyển tiếp lần 🡪 Bộ đánh dấu là

+ Tiếp theo, ta sẽ kích hoạt chuyển tiếp lần 🡪 Bộ đánh dấu là

+ Cuối cùng, ta sẽ kích hoạt chuyển tiếp lần 🡪 Bộ đánh dấu là

🡪 Ta có điều phải chứng minh

- Do đó, số trạng thái của hệ thống chuyển tiếp chính là cách sắp xếp 4 số 3,2,1,0 vào 4 vị trí (một số có thể xuất hiện nhiều lần)

**\* Tìm số chuyển tiếp của hệ thống chuyển tiếp**

- Hai bộ đánh dấu

và

có một cạnh nối giữa chúng nếu

và

- Từ bộ đánh dấu có chuyển tiếp qua bộ đánh dấu kề bằng cách kích hoạt một trong bốn chuyển tiếp

- Ta xét các trường hợp sau:

+ Mỗi bộ đánh dấu sẽ nối đến 4 bộ đánh dấu kề nếu 🡪 Có bộ như vậy 🡪 Có chuyển tiếp

+ Mỗi bộ đánh dấu sẽ nối đến 3 bộ đánh dấu kề nếu có duy nhất một vị trí có số token bằng 0

🡪 Có bộ đánh dấu như vậy 🡪 Có chuyển tiếp

+ Mỗi bộ đánh dấu sẽ nối đến 2 bộ đánh dấu kề nếu có đúng hai vị trí có số token bằng 0

🡪 Có bộ đánh dấu như vậy 🡪 Có chuyển tiếp

+ Mỗi bộ đánh dấu sẽ nối đến 1 bộ đánh dấu kề nếu có đúng ba vị trí có số token bằng 0

🡪 Có bộ đánh dấu như vậy 🡪 Có chuyển tiếp

🡪 Có tổng cộng chuyển tiếp

3. Nếu cho phép sự đồng thời diễn ra trong mạng này, có bao nhiêu trạng thái và bao nhiêu chuyển tiếp của hệ thống chuyển tiếp được tạo ra ?

- Khi cho phép sự đồng thời (true concurrency), mỗi chuyển tiếp trong hệ thống chuyển tiếp sẽ cho phép xuất hiện ít nhất một transition

- Tương tự ở trên, ta suy ra được số trạng thái của hệ thống chuyển tiếp là

**\* Tìm số chuyển tiếp của hệ thống chuyển tiếp:**

- Ta xét các trường hợp sau

+ , tức là 4 vị trí đều có số token khác 0. Khi đó, ta có thể chọn ít nhất 1 trong 4 chuyển tiếp để kích hoạt:

🡪 Có cách chọn, mặt khác, số bộ đánh dấu thỏa mãn điều kiện là 🡪 Có chuyển tiếp

+ , tức là tồn tại duy nhất một chuyển tiếp có số token bằng 0. Khi đó, ta có thể chọn ít nhất là 1, tối đa là 3 chuyển tiếp để kích hoạt:

🡪 Có cách chọn, mặt khác số bộ đánh dấu thỏa mãn điều kiện là 🡪 Có chuyển tiếp

+ , tức tức là tồn tại đúng hai chuyển tiếp có số token bằng 0. Khi đó, ta có thể chọn ít nhất là 1, tối đa là 2 chuyển tiếp để kích hoạt:

🡪 Có cách chọn, mặt khác số bộ đánh dấu thỏa mãn điều kiện là 🡪 Có chuyển tiếp

+ , tức tức là tồn tại đúng ba chuyển tiếp có số token bằng 0. Khi đó, ta có thể chọn tối đa là 1 chuyển tiếp để kích hoạt:

🡪 Có cách chọn, mặt khác số bộ đánh dấu thỏa mãn điều kiện là 🡪 Có chuyển tiếp

🡪 Có tổng cộng

Vậy nếu cho phép đồng thời (true concurrency) thì hệ thống có 256 trạng thái và 2145 chuyển tiếp

**5.6. Mô hình hóa mạng Petri**

**5.6.1. Một số khái niệm cơ bản trong việc mô hình hóa mạng Petri**

- Một token có thể mô hình cho nhiều thứ khác nhau:

+ Một đối tượng vật lý. Ví dụ: sản phẩm, chi tiết, thuốc, con người,…

+ Một đối tượng thông tin. Ví dụ: tin nhắn, tin hiệu, báo cáo,…

+ Một tập hợp các đối tượng

+ Một chỉ thị cho trạng thái. Ví dụ: trạng thái đèn giao thông

+ …

- Một vị trí có thể chứa token. Vai trò của vị trí trong cấu trúc mạng Petri có mối liên hệ mật thiết với token mà nó chứa. Một vị trí có thể mô hình cho:

+ Một buffer

+ Một trung gian giao tiếp

+ Một vị trí địa lý

+ Một trạng thái khả thi hoặc là một điều kiện của trạng thái

- Vị trí là phần tử thụ động của mạng Petri. Một token trong vị trí biểu diễn một phần trạng thái của mạng Petri, nhưng một vị trí không thể tự thay đổi trạng thái. Chuyển tiếp là thành phần chủ động của mạng Petri. Khi một chuyển tiếp kích hoạt, trạng thái của mạng Petri thay đổi

- Một chuyển tiếp có thể mô hình cho:

+ Một sự kiện

+ Một sự chuyển đổi các đối tượng

+ Một sự vận chuyển các đối tượng

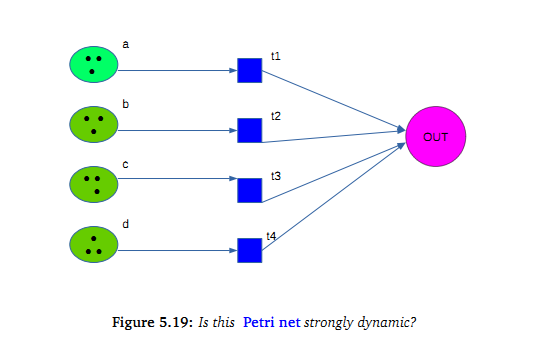
🡪 Biểu diễn sự kiện bằng chuyển tiếp, biểu diễn trạng thái bằng vị trí và token

**5.6.2. Động lực của mạng Petri thông qua một số tính chất đặc biệt:**

- Một mạng Petri đánh dấu là k-bounded nếu không có vị trí nào chứa nhiều hơn k token. Một cách khác,

- Một mạng Petri gọi là **safe** nếu và chỉ nếu nó là 1-bounded

- Một mạng Petru gọi là **bị chặn** nếu và chỉ nếu tồn tại sao cho mạng Petri là k-bounded



- Một mạng Petri gọi là **deadlock free** nếu tất cả các bộ đánh dấu có thể truy cập có ít nhất một chuyển tiếp tích cực. Hay nói một cách khác

- Một chuyển tiếp trong mạng Petri đánh dấu là **live** nếu trong tất cả các bộ mạng dấu có thể truy cập, t đều tích cực. Một cách trang trọng

- Một mạng Petri đánh dấu là **live** nếu tất cả các chuyển tiếp là live

**5.6.3.** **Vấn đề khi mô hình hóa mạng Petri**

- Trong tiến trình, thông thường sẽ có nhiều nhân tố, chúng có các hoạt động nghiệp vụ khác nhau. Do đó, mạng Petri của chúng sẽ có các vị trí / token và chuyển tiếp phân biệt

- Tuy nhiên, động lực của hệ thống toàn thể (bởi tất cả nhân tố) được xây dựng bởi sự tương tác lẫn nhau, một số vị trí có thể chia sẻ token và đôi lúc là chuyển tiếp

- Vì vậy, mạng Petri của cả hệ thống không đơn giản là hợp nhất phân biệt của các mạng thành phần. Mà phải được chồng chất (superimposition) từ các mạng nhỏ hơn cấu thành nên nó. Ví dụ:

+ Hệ thống y tế, bệnh viện: Một ít nhất 2 loại nhân tố - nhân viên y tế và bệnh nhân tương tác lẫn nhau, chia sẻ chung nguồn tài nguyên (vị trí) và di chuyển cùng với nhau (chuyển tiếp)

+ …

**Question 5.3.** Làm thế nào để xây dựng mạng Petri cho các hệ thống lớn mà không làm mất các kiến thức/thông tin cơ bản của các mạng thành phần, bên cạnh đó phải cho thấy được động lực đúng đắn của hệ thống đó ?

Định nghĩa 5.9. (Toán tử chồng chất)

Xét hệ thống có hai loại nhân tố, kí hiệu là mạng Petri của chúng.

Giả định rằng và , trong đó vị trí có thể phân biệt, nhưng với cùng bộ đánh dấu khởi tạo

a. Toán tử chồng chất, , trong đó , được định nghĩa như sau:

- Nếu thì với (biểu diễn của là giống nhau), giữ lại một bản trong mạng gộp)

- Nếu thì , giữ lại cả hai biểu diễn

Cụ thể, chúng ta có thể định nghĩa hai chuyển tiếp/sự kiện của hai mạng thành một node của mạng Petri hợp nhất nếu mọi sự kiện diễn ra trên cùng token vật lý

b. Toán tử chồng chất (hợp nhất) mạng Petri được xác định bởi:

**5.6.4. Bài tập lớn: Chuyên gia tư vấn y tế**

- Bối cảnh: Dưới thảm họa dịch bệnh SARS gây ra một sự một sự thiếu hụt lớn về giường bệnh ICU trong thành phố H, bệnh nhân cần được chuyên gia tư vấn về việc ngoại trú của bệnh viện X. Chúng ta sẽ sử dụng mạng Petri để mô hình hóa tiến trình này

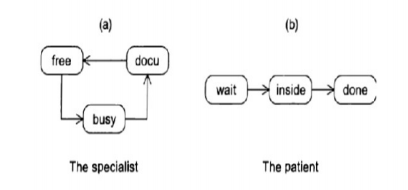
- Giả định và dữ liệu:

+ Chuyên gia: Mỗi bệnh nhân có một buổi gặp mặt với các chuyên gia hiện tại. Chuyên gia sẽ nhận bệnh nhân. Vào mỗi thời điểm, chuyên gia chỉ ở một trong các trạng thái dưới đây

* Chuyên gia đang rảnh và chờ bệnh nhân tiếp theo (trạng thái ***free***)
* Chuyên gia đang bận tư vấn cho bệnh nhân (trạng thái ***busy***)
* Chuyên gia đang phân tích kết quả của lượt tư vấn (trạng thái ***docu***)

+ Mỗi bệnh nhân đến gặp chuyên gia đều ở một trong ba trạng thái:

* Bệnh nhân đang chờ (trạng thái ***wait***, nhận giá trị n nếu có n bệnh nhân đang chờ)
* Bệnh nhân đang được tư vấn bởi chuyên gia (trạng thái ***inside***)
* Bệnh nhân đã được tư vấn bởi chuyên gia (trạng thái ***done***)



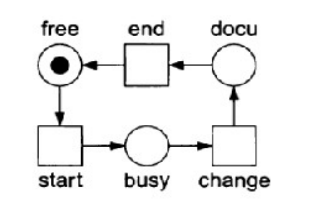
+ Chuyên gia sẽ đi qua ba trạng thái lặp đi lặp lại. Bệnh nhân đi qua ba trạng thái một lần duy nhất (trên một lượt)

+ Dữ liệu sự kiện: Có ba sự kiện rất quan trọng đối với quá trình nghiệp vụ

* Đầu tiên, chuyên gia bắt đầu với lượt tư vấn với bệnh nhân (sự kiện ***“start”***)
* Thứ hai, chuyên gia kết thúc lượt tư vấn với bệnh nhân và tiến hành phân tích kết quả của lượt tư vấn (sự kiện ***“change”***)
* Thứ ba, chuyên gia kết thúc quá trình phân tích (sự kiện ***“end”***)

**Mục tiêu: Mô hình quá trình nghiệp vụ (theo từng bước) bởi mạng Petri**

**1. Cho mạng Petri mô hình hóa trạng thái của chuyên gia**



**Trong bộ đánh dấu được hiển thị, chuyên gia đang ở trạng thái *free***

**1a. Chỉ ra vị trí và chuyển tiếp của mạng petri trên**

-

-

**1b. Biểu diễn mạng petri trên bằng hệ thống chuyển tiếp với giả định rằng**

**i. Mỗi vị trí không thể chứa nhiều hơn một token trong bất kì bộ đánh dấu nào**

**ii. Mỗi vị trí có thể chứa bất kì token trong bất kì bộ đánh dấu nào**

- Ta sẽ biểu diễn bộ đánh dấu là bộ ba với là số token trong *free*, là số token trong *busy*, là số token trong *docu*

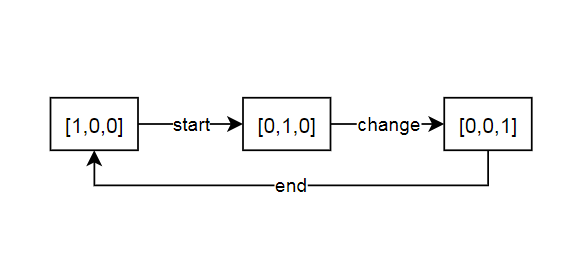
Không cho phép đồng thời (interleaving semantic)

*i. Xét các trường hợp của bộ đánh dấu khởi tạo :*

- . Khi đó số trạng thái của hệ thống TS là 1 và có 0 chuyển tiếp

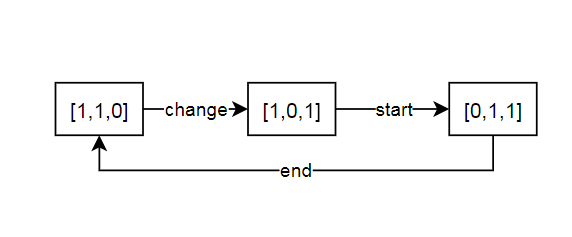
-

Khi đó, số trạng thái của hệ thống TS là 3 và có 3 chuyển tiếp



-

Khi đó số trạng thái của hệ thống TS là 3 và có 3 chuyển tiếp



- . Khi đó số trạng thái của hệ thống TS là 1 và có 0 chuyển tiếp

*ii. Ta xét bộ đánh dấu ban đầu*

*Tìm số trạng thái của TS*

- Vì mỗi vị trí cho chứa bất kì số token trong bất kì bộ đánh dấu nào. Nên ta có

- Mỗi lần kích hoạt một trong ba chuyển tiếp start, change hay end đều tạo ra một token và tiêu thụ một token. Do đó số token trong mạng sẽ không đổi hay

- Xét các bộ đánh dấu với và . Ta sẽ chứng minh chung đều có thể truy cập. Thật vậy:

+ Do nên không mất tính tổng quát, ta có thể giả sử

+ Kích hoạt chuyển tiếp start lần. Bộ đánh dấu lúc đó sẽ là

+ Xét hai trường hợp:

TH1:

* Kích hoạt chuyển tiếp change lần. Điều này khả thi, bởi vì lúc đó, vị trí busy có . Bộ đánh dấu lúc đó sẽ là
* Mặt khác , do đó
* Vì vậy, bộ đánh dấu là

TH2:

* Kích hoạt chuyển tiếp end lần. Bộ đánh dấu sẽ là
* Kích hoạt chuyển tiếp start lần. Bộ đánh dấu sẽ là
* Mặt khác , do đó
* Vì vậy, bộ đánh dấu là

- Do đó, tất cả các bộ đánh dấu với đều có thể đạt được (n là tổng số token ban đầu)

- Vậy số trạng thái của hệ thống chuyển tiếp sẽ là số nghiệm của phương trình

với là tổng số token ban đầu và

- Theo bài toán chia kẹo Euler, phương trình này sẽ có nghiệm

- Do đó, số trạng thái sẽ là

*Tìm số chuyển tiếp của TS*

- Xét bộ đánh dấu

+ Nếu thì sẽ có ba chuyển tiếp nối từ bộ đánh dấu đến các bộ đánh dấu khác tương ứng với ba transition start, change và end.

Với , dễ dàng thấy sẽ không tồn tại bộ nào thỏa điều kiện

Với , số bộ đánh dấu thỏa sẽ là nghiệm của phương trình

với là tổng số token ban đầu và

Theo bài toán chia kẹo Euler, số trạng thái này là

+ Nếu tồn tại đúng một vị trí có số token bằng 0 thì sẽ có hai chuyển tiếp nối từ bộ đánh dấu đến các bộ đánh dấu khác

Với , dễ dàng thấy sẽ không tồn tại bộ nào thỏa điều kiện

Với , không mất tính tổng quát, giả sử . Khi đó, số bộ đánh dấu thỏa sẽ là nghiệm của phương trình

với là tổng số token ban đầu và

Theo bài toán chia kẹo Euler, số trạng thái này là

🡪 Trong TS, có trạng thái thỏa điều kiện

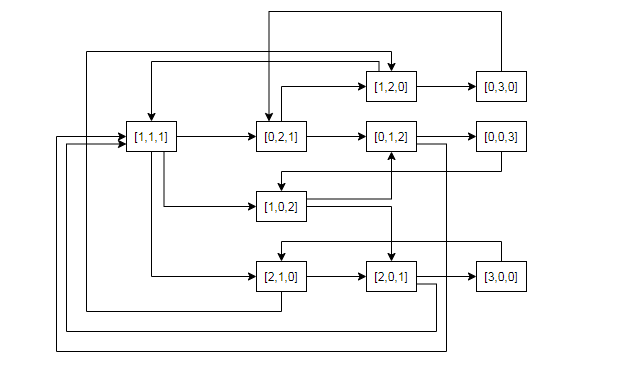
+ Nếu tồn tại đúng hai vị trí có số token bằng 0 thì sẽ có một chuyển tiếp nối từ bộ đánh dấu đến các bộ đánh dấu khác

Không mất tính tổng quát, giả sử . Khi đó, số bộ đánh dấu thỏa sẽ là nghiệm của phương trình

Có một trạng thái như vậy

🡪 Trong TS, có 3 trạng thái thỏa điều kiện như vậy là

- Tóm lại, hệ thống chuyển tiếp TS từ mạng Petri sẽ có trạng thái. Số chuyển tiếp

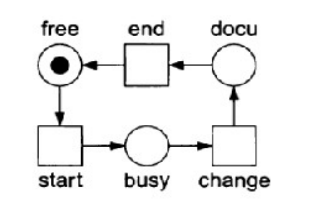


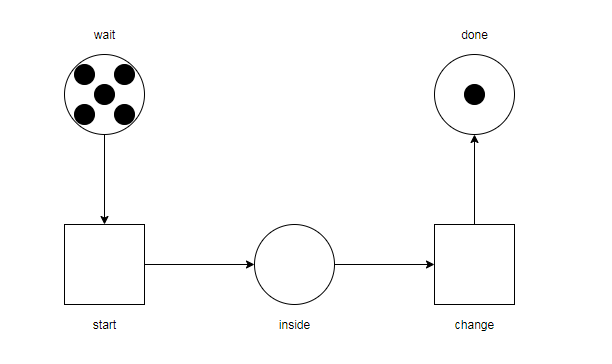
Ví dụ với n = 3

**2. Định nghĩa là mạng Petri mô hình trạng thái của bệnh nhân**

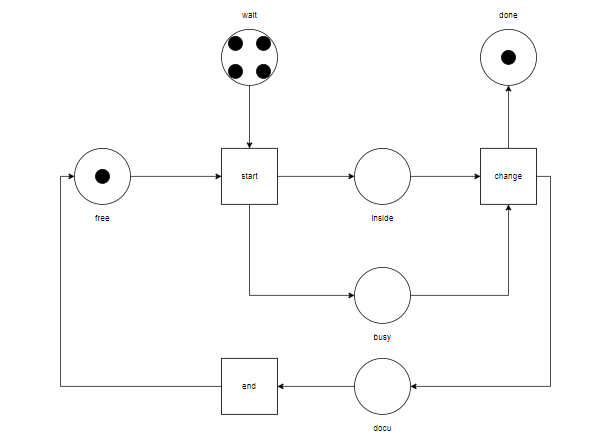
**a. Diễn tả các ý nghĩa khả thi của token trong trạng thái *inside* của mạng**

**b. Xây dựng mạng Petri , giả sử rằng có 5 bệnh nhân ở trạng thái wait, không có bệnh nhân ở trạng thái inside và một bệnh nhân ở trạng thái done**

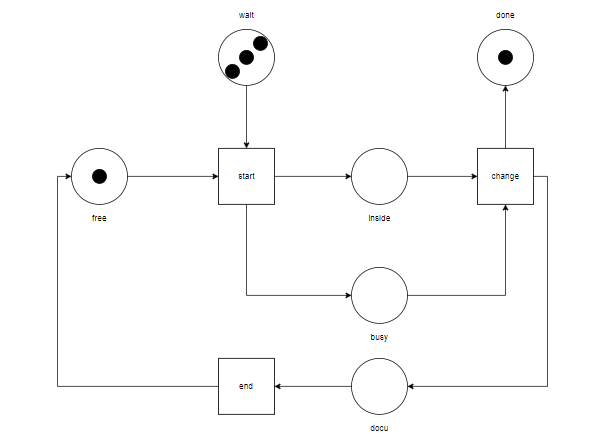


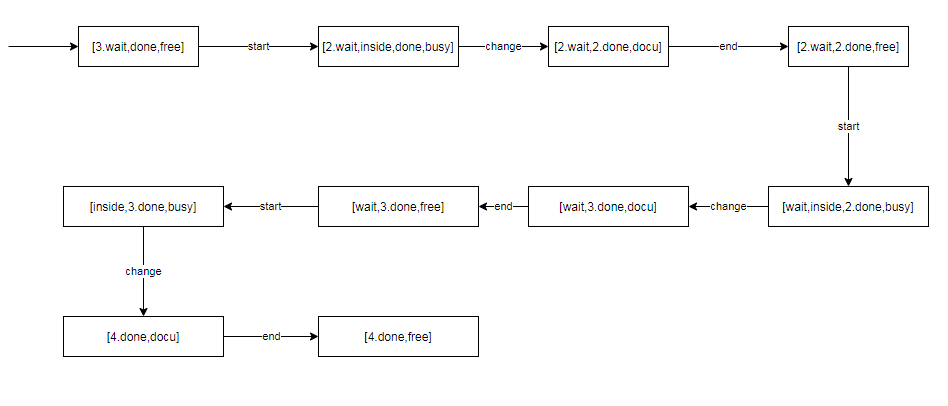


**3. Xác định mạng Petri chồng chất mô hình , cho phép một chuyên gia tư vấn cho bệnh nhân, giả định rằng có bốn bệnh nhân đang chờ để gặp chuyên gia, một bệnh nhân ở trạng thái *done*, và chuyên gia đang ở trạng thái free (Diễn tả các nghiệp vụ xung quanh bệnh nhân)**



**4. Xem xét bộ đánh dấu khởi tạo trong mạng Petri . Bộ đánh dấu nào có thể đạt được từ bằng cách kích hoạt một chuyển tiếp trong một lần. Vì sao ?**





**5. Mạng Petri N có deadlock free hay không. Giải thích**

- Mạng Petri N không phải là mạng Petri deadlock free. Bởi vì số lượng bệnh nhân là có hạn nên sẽ tồn tại bộ đánh dấu . Khi đó, chuyên gia ở trạng thái *free* và không có bệnh nhân nào ở trạng thái *wait* 🡪 Không tồn tại chuyển tiếp tích cực

**6. Đề xuất một mạng Petri đơn giản với hai chuyên gia sẵn sàng để tư vấn cho bệnh nhân. Diễn tả cấu trúc một cách mạch lạc**

- Dùng mạng Petri tô màu

**7. Viết package hiện thực câu 1, 2, 3, 4. Chương trình có thể chứa tối đa 10 bệnh nhân trong vị trí *wait***

1

- 2 người: chuẩn bị kiến thức (định nghĩa, hành vi, tính chất, hợp nhất) (Phú + Khoa)

- 2 người: câu 1 🡪 câu 6 (Minh + Đăng)

- 1 người: câu 7 (Cát)