*Assignment 3: Bayesian Network*

# 1. Giới thiệu

Mục đích của bài tập lớn này nhằm kiểm tra mức độ hiểu và vận dụng của các sinh viên với chủ đề suy diễn dựa vào xác suất; cụ thể là xây dựng và suy diễn với mạng Bayes.

Phần code cho sẵn ban đầu chứa các file sau đây, nằm bên trong file bayesnets.zip.

| | |
|---|---|
| **Files bạn sẽ chỉnh sửa:** | |
| bayesianNetwork.py | Tất cả các hàm cần chỉnh sửa sẽ nằm trong file này. Sinh viên sẽ nộp lại duy nhất file này. |
| **Files hỗ trợ:** | |
| `main.py` | Đây là file main để thực thi bài tập lớn này |
| `models/` | Thư mục chứa các các tập tin mô tả mô hình |
| `testcases/` | Thư mục chứa các tập tin mô tả câu hỏi (truy vấn) |

**Lệnh thực thi**

```
python main.py --model=<tên file model> --testcase=<tên file test>
```

**Ví dụ 1:**

```
python main.py --model=model01.txt --testcase=testcase01.txt
```

# 2. Nội dung công việc

Trong bài tập lớn này, các bạn được yêu cầu xây dựng lớp BayesianNetwork, các hàm và các cấu trúc dữ liệu để cho phép đặc tả Bayes Nets cũng như thực hiện công việc suy diễn trên Bayes Nets với các hình thức chính xác (exact) và xấp xỉ (approximate). Trong đó:

- Hiện thực suy diễn chính xác là bắt buộc.
- Hiện thực suy diễn xấp xỉ là phần không bắt buộc và sẽ được tính điểm cộng.

Cụ thể sinh viên cần hoàn thành các hàm sau trong lớp `BayesianNetwork`:

1. Hoàn tất hàm `init()` để khởi tạo đối tượng thuộc lớp `BayesianNetwork`. Hàm này sẽ nhận đầu vào là một file chứa đặc tả về mạng Bayes Nets. Chi tiết về định dạng của file này được mô tả ở **Phần 3.1**.
2. Hoàn tất hàm `exact_inference()` để thực hiện việc suy luận chính xác trên Bayes Nets. Hàm này sẽ nhận đầu vào là một file chứa đặc tả về câu truy vấn (câu hỏi). Chi tiết về định dạng của file này được mô tả ở **Phần 3.2**.
3. Hoàn tất hàm `approx_inference()` để thực hiện việc suy luận xấp xĩ trên Bayes Nets. Hàm này sẽ nhận đầu vào là một file chứa đặc tả về câu truy vấn (câu hỏi). Chi tiết về định dạng của file này được mô tả ở P**hần 3.2**.

Lưu ý: Sinh viên có thể viết thêm các hàm hỗ trợ khác trong class BayesianNetwork hoặc các class hỗ trợ khác nhưng phải nằm trong file bayesianNetwork.py.

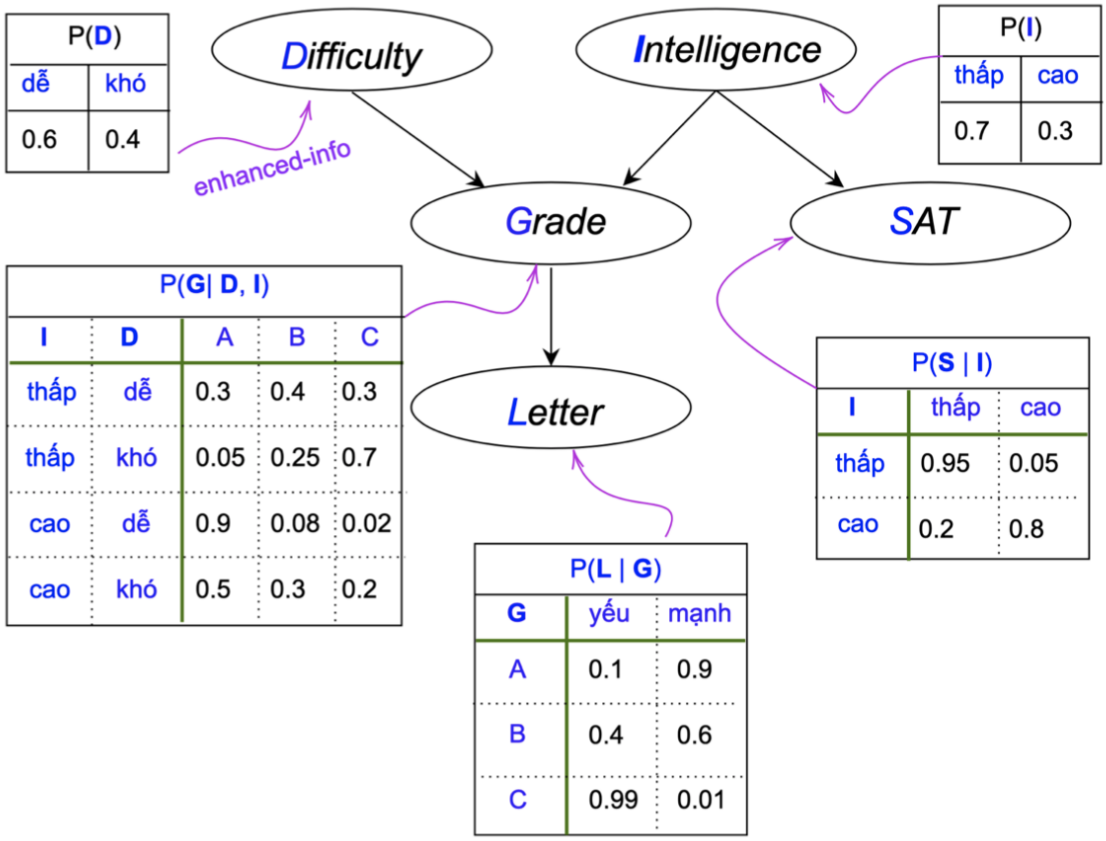
# 3. Định dạng tập tin

## 3.1. Định dạng tập tin đặc tả mô hình

Tập tin mô hình là một tập tin văn bản, có cấu trúc như ví dụ sau. Ở đó, các biến, các tập giá trị, v.v được lấy từ Bayes Nets trong **Hình 1**. Phần mô tả cho định dạng này được trình bày sau đây.

**Ví dụ 2:**

```
5
I;;Thap,Cao;2;0.7,0.3
D;;De,Kho;2;0.6,0.4
G;I,D;A,B,C;2,2,3;0.3,0.4,0.3,0.05,0.25,0.7,0.9,0.08,0.02,0.5,0.3,0.2
L;G;Yeu,Manh;3,2;0.1,0.9,0.4,0.6,0.99,0.01
S;I;Thap,Cao;2,2;0.95;0.05,0.2,0.8
```




**Hình 1**: Một mạng Bayes - dùng làm minh họa trong tài liệu này.

- Dòng đầu tiên của tập tin mô hình là một số nguyên N ($2 \le N \le 20$). Đây là số biến ngẫu nhiên của Bayes Net hay số node trên đồ thị DAG (minh họa trong **Hình 1**).
- Kể từ dòng thứ 2, mỗi dòng đặc tả thông tin cần thiết để khởi tạo một node trong đồ thị. Thứ tự các node được liệt kê từ trên xuống dưới đã thoả mãn thứ tự topo của đồ thị DAG. Thông tin này bao gồm 4 phần được ngăn cách bởi dấu chấm phẩy **';'** như sau: **NODE**; **PARENTS**; **DOMAIN**; **SHAPE**; **PROBABILITIES**
    - **NODE**: là tên của một biến ngẫu nhiên hay node trên đồ thị. Đây là một chuỗi chỉ ký tự chữ, số, và không chứa ký tự đặc biệt.
    - **PARENTS**: là danh sách chứa các node cha của node hiện tại và được ngăn cách với nhau bởi dấu phẩy **','**. Danh sách này có thể rỗng trong trường hợp node hiện tại không có cha.

- **DOMAIN**: là danh sách chứa tập giá trị của biến hiện tại, các giá trị trong tập này được phân tách bởi dấu phẩy **‘,’**.
- **SHAPE**: là danh sách chứa các số nguyên dương: $D_1$, $D_2$, …, $D_n$. Trong đó, $D_1$ đến $D_{n-1}$ tương ứng là kích thước tập giá trị của các node cha theo đúng thứ tự được liệt kê trong PARENTS. $D_n$ là kích thước của node hiện tại.
- **PROBABILITIES**: là dãy gồm $D_1 \times D_2 \times .. \times D_m$ số thực. Đây là bảng xác suất có điều kiện được lưu dưới dạng tuyến tính hóa cho array nhiều chiều ($D_1$,$D_2$,..,$D_m$) với cách lưu trữ của ngôn ngữ C/C++ (row-major order).

Lưu ý: Code cho sẵn đã xử lý phần đọc định dạng file này. Sinh viên không cần xử lý những trường hợp tập tin mô tả mô hình không đúng định dạng nêu trên.

## 3.2. Định dạng tập tin mô tả câu hỏi

Mỗi tập tin mô tả câu hỏi chứa một câu hỏi duy nhất và được chia làm 2 phần, ngăn cách với nhau bởi dấu chấm phẩy **‘;’**:

- Phần một là các biến truy vấn (query), nếu có nhiều biến truy vấn thì ngăn cách nhau bởi dấu phẩy **‘,’**. Các biến truy vấn này sẽ được gán giá trị cụ thể trong miền giá trị của nó thông qua dấu **‘=’**. Lưu ý phần biến truy vấn không được phép rỗng.
- Phần hai là các biến bằng chứng (evidence), nếu có nhiều biến bằng chứng thì ngăn cách nhau bởi dấu phẩy **‘,’**. Các biến bằng chứng này sẽ được gán giá trị cụ thể trong miền giá trị của nó thông qua dấu **‘=’**. Lưu ý phần biến bằng chứng được phép rỗng (như **Ví dụ 4**).

**Ví dụ 3:**

```
G=A;I=Cao,D=Kho
```

Tương đương với yêu cầu tính `P(G=A | I=Cao, D=Kho)`

**Ví dụ 4:**

```
G=B,S=Cao;
```

Tương đương với yêu cầu tính `P(G=B, S=Cao)`

Lưu ý: Code cho sẵn đã xử lý phần đọc định dạng file này. Sinh viên không cần xử lý những trường hợp tập tin mô tả mô hình không đúng định dạng nêu trên.

# 4. Quy định và cách chấm điểm

## 4.1. Quy định:

- Thời gian thực thi tối đa cho mỗi truy vấn là: 20 giây
- Sai số so với đáp án là ± 10^-5
- Môi trường thực thi: python 3.7
- Thư viện hỗ trợ: numpy, các thư viện chuẩn của python. Ngoài ra, **KHÔNG được sử dụng** các thư viện ngoài khác.

## 4.2. Cách tính điểm:

Tổng điểm bài tập lớn này là 110 điểm: bao gồm 100 điểm phần bắt buộc vào 10 điểm phần điểm cộng. Điểm này được tính như sau:

- Có 25 câu truy vấn chính xác (mỗi câu 4 điểm)
- Có 05 câu truy vấn xấp xĩ (mỗi câu 2 điểm)