# TDT4171 Artificial Intelligence Methods

Uncertainty, Bayesian networks

January 21, 2021

- **Delivery deadline: February 03, 2021 by 23:59.**
- Required reading for this assignment: Chapter 13, 14 (the parts in the curriculum).
- Deliver your solution on Blackboard. Please upload your report as a **PDF file**. For the programming part, please upload the source code alongside the PDF. Please **do not** put the PDF or source code in an archive.
- Students can NOT work in groups. Each student can only submit solution individually.
- This homework counts for 10% of the final grade.
- This assignment totals 10 points. The number of points for each problem is labeled. All sub-problems count equally towards that problem's total.
- Cribbing(”koking”) from other students is not accepted, and if detected, will lead to immediate failure of the course. The consequences will apply to both the source and the one cribbing.

# Problem 1

**2 points**

The probability that a person has 0, 1, 2, 3, 4, or 5 or more siblings is 0.15, 0.49, 0.27, 0.06, 0.02, 0.01, respectively.

**a)** What is the probability that a child has at most 2 siblings?

**b)** What is the probability that a child has more than 2 siblings given that he has at least 1 sibling?

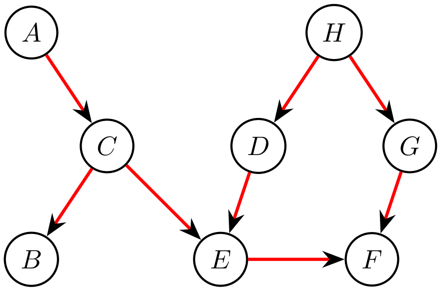
**c)** Three friends who are not siblings are gathered. What is the probability that they combined have three siblings?

**d)** Emma and Jacob are not siblings, but combined they have a total of 3 siblings. What is the probability that Emma has no siblings?

# Problem 2

**2 points**

Given the Bayesian network structure below, decided whether the statements are true or false. Justify each answer with an explaination.



**a)** If every variable in the network has a Boolean state, then the Bayesian network can be represented with 18 numbers.

**b)** $G \perp\!\!\!\perp A$

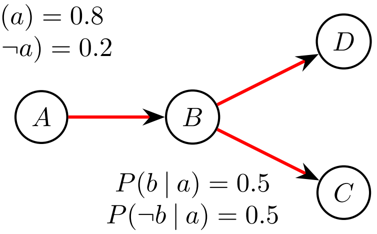
**c)** $E \perp\!\!\!\perp H \mid \{D, G\}$

**d)** $E \perp\!\!\!\perp H \mid \{C, D, F\}$

# Problem 3

**2 points**

The Bayesian network below contains only binary states. The conditional probability for each state is listed. From the Bayesian network, calculate the following probabilities:



$P(a) = 0.8$
$P(\neg a) = 0.2$

$P(d \mid b) = 0.6$
$P(\neg d \mid b) = 0.4$
$P(d \mid \neg b) = 0.8$
$P(\neg d \mid \neg b) = 0.2$

$P(c \mid b) = 0.1$
$P(\neg c \mid b) = 0.9$
$P(c \mid \neg b) = 0.3$
$P(\neg c \mid \neg b) = 0.7$

$P(b \mid a) = 0.5$
$P(\neg b \mid a) = 0.5$
$P(b \mid \neg a) = 0.2$
$P(\neg b \mid \neg a) = 0.8$

**a)** $P(b)$

**b)** $P(d)$

**c)** $P(c \mid \neg d)$

**d)** $P(a \mid \neg c, d)$

# Problem 4

**4 points**

For this exercise you are going to implement inference by enumeration for Bayesian neural networks. The algorithm is detailed in Figure 14.9 of *Artificial Intelligence: A Modern Approach* [1]. It is completely up to you what programming language to use, but make sure your code is readable. This includes sensible variable and function names, and comments where appropriate.

We have provided a Python file implementing parts of this exercise. You are free to use this as is, and implement only the remaining parts. It is not a requirement to use the provided code; you can refactor it or disregard it if you wish. In the event you are not using the provided code, please provide a clear description of the environment you are using to run the code, including version of programming language, packages, and other details needed to run your code.

**Make sure to read the whole exercise before you start coding.**

**a)** First you need to implement a Bayesian network that supports discrete conditional probability distributions for each state. This includes

- A discrete conditional probability table (CPT). This class should have the following functionality: PROBABILITY(EVENT, EVIDENCE), a function that takes in a variable state and the state of the variable's parents, and returns the conditional probability of that event given the evidence. This function is not meant to calculate anything. It is only meant to look up the probability in the CPT. (This part is implemented in the provided code. See class Variable.)
- A directed acyclic graph (DAG) to represent the network topology. A Bayesian network is a DAG of variables. It is up to you how you want to implement the DAG, but you will need to be able to add variables and edges to the network. You will also need to be able to provide a lexicographic ordering of the nodes (see below). (This is implemented in the provided code. See class BayesianNetwork.)
- A lexicographical topological ordering of nodes. Lexicographical ordering is required for FIRST(*vars*) in ENUMERATE-ALL to work. A lexicographical ordering ensures that each selected node has all its parent states fixed. You can use any method you choose to select nodes in correct order, but a tip is to use *Kahn's algorithm* to put the nodes in a sorted list before iterating through them. You can find the algorithm described here: `https://en.wikipedia.org/wiki/Topological_sorting`

**If you choose to use the provided code, the lexicographic ordering of the variables is the only part of this exercise (4a) you will need to implement.**

**b)** Implement the inference by enumeration algorithm found in Figure 14.9 of *Artificial Intelligence: A Modern Approach* [1]. We have also provided an outline of the algorithm below.

NB: When using a mutable type (list, dict, etc.) in python a pointer, and not a copy of the object is passed as arguments in a function. To ensure correct behaviour in recursive algorithms you will often have to copy the object with the *.copy()* method.

**Algorithm 1** Inference by Enumeration

```
function ENUMERATION-ASK($X, \boldsymbol{e}, bn$)
    $\boldsymbol{Q}(X) \leftarrow$ a distribution over $X$, initially empty
    for each value $x_i$ of $X$ do
        $\boldsymbol{Q}(x_i) \leftarrow$ ENUMERATE-ALL($bn$.VARS, $\boldsymbol{e}_{x_i}$)        ▷ $\boldsymbol{e}_{x_i}$ is $\boldsymbol{e}$ extended with $X = x_i$
    return NORMALIZE($\boldsymbol{Q}(X)$)

function ENUMERATE-ALL($vars, \boldsymbol{e}$)
    if EMPTY?($vars$) then return 1.0
    $Y \leftarrow$ FIRST($vars$)
    if $Y$ has value $y$ in $\boldsymbol{e}$ then
        return $P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\boldsymbol{e}$)
    else
        return $\sum_y P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$, $\boldsymbol{e}_y$))
                                                                    ▷ $\boldsymbol{e}_y$ is $\boldsymbol{e}$ extended with $Y = y$
```

**c)** The Monty Hall problem is a probability puzzle based on the American TV-show *Let's Make a Deal*, and named after its original host Monty Hall. The puzzle goes as follows:

> Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, ”Do you want to pick door No. 2?” Is it to your advantage to switch your choice?

We assume that the game show hosts always acts according to the following rules:

1. The host must always open a door that was not picked by the contestant.
2. The host must always open a door to reveal a goat and never the car.
3. The host must always offer the chance to switch between the originally chosen door and the remaining closed door.

Model the Monty Hall problem as a Bayesian Network using the following states *ChosenByGuest*, *OpenedByHost*, and *Prize*. Use your implementation of inference by enumeration, and the evidence described in the problem statement to answer the question; is it to your advantage to switch your choice? Answering this question entails calculating

$$P(Prize \mid ChosenByGuest = 1, OpenedByHost = 3)$$

(As an example, we have implemented problem 3c) in the provided code.)

**Turn in your code, a drawing of the Bayesian network with conditional probability tables, and the calculated posterior probabilities.**

# References

[1] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 3 edition, 2010.