

# **PRESENTATION PROJET ALGO AV.**

## **Intelligence Artificielle**

### **Conception d'un**

### **Système Expert**

Lundi 11/03/24 – Vendredi 15/03/24

ISEN 3

**ISEN**  
MÉDITERRANÉE  
L'ÉCOLE  
DES INGÉNIEURS  
DU NUMÉRIQUE

# PROJET D'APPLICATION DU COURS D'ALGO AVANCEE

Dans l'alignement du cours d'algorithmique avancée, utilisation des aspects suivants du C :

- listes chaînées,
- récursivité,

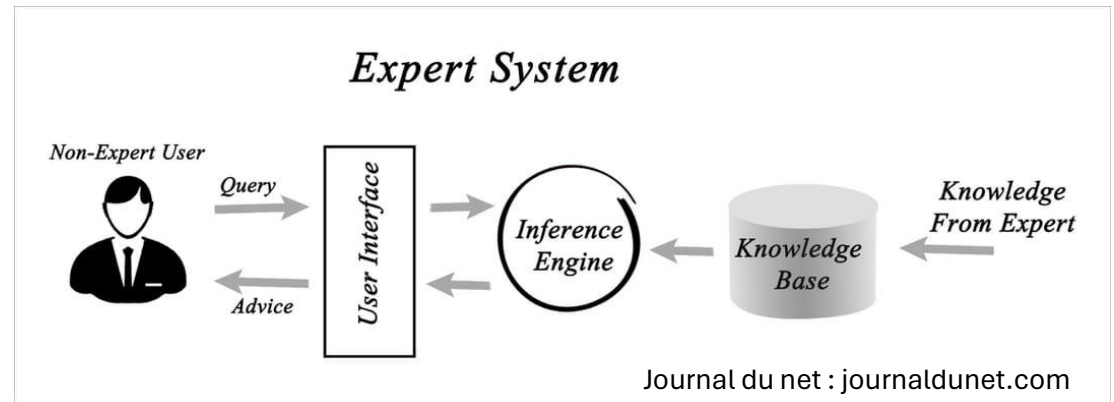
Mais aussi *une* démarche projet

Et enfin *un* domaine d'application spécifique

# SUJET PROPOSE

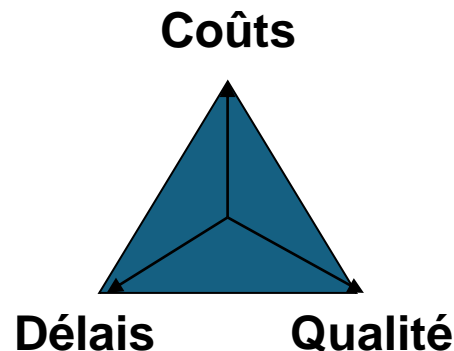
Sujet Système Expert :

- Spécifications du système expert (moteur d'inférences),
- Développement du système expert,
- spécification d'un domaine d'expertise à l'aide de règles (base de connaissances)



# DEMARCHE PROJET (1)

- A intégrer impérativement pour tout élève ingénieur.
- L'objectif ***réussite projet*** se trouve à l'intérieur du triangle vertueux :



- Gérer les délais, le temps : son temps, (ou son manque de temps)
- Gérer les coûts : essentiellement les dépassements de temps
- Gérer la qualité : le respect du cahier des charges, les normes de codage

## DEMARCHE PROJET (2)

- Préciser le périmètre fonctionnel
- Prévoir l'architecture et les composants logiciels
- Découper en tâches et sous-tâches,
- Evaluer les charges de travail
- Identifier les risques
- Prévoir et effectuer des points d'avancement

Dans le cas de ce projet : risque majeur de non maîtrise du planning

# INTELLIGENCE ARTIFICIELLE

## LE TEST DE TURING (1)

- Le Mathématicien anglais Alan Turing 1912-1954 propose de tenter de répondre à la question :  
« Les machines peuvent-elles penser ? ».

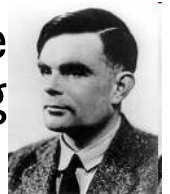


- Il propose donc un test pour répondre à cette question en 1950 dans son article *Computing Machinery and Intelligence* (considéré comme l'article fondateur de l'IA).

# INTELLIGENCE ARTIFICIELLE

## LE TEST DE TURING (2)

- Pendant une heure, un juge dialogue ***sur n'importe quel thème*** avec un programme et un humain via des lignes de communication informatiques.
- Si il est impossible de déterminer l'interlocuteur humain e programme, alors ce dernier est considéré comme intellig (ou imitant l'intelligence).



- (1) Le prix Loebner (100.000 USD) récompense le programme qui réussit le test de Turing (pas encore gagné à ce jour!)
- (2) A noter que ChatGPT ne veut pas passer le test de turing (c'est une IA de service qui ne cherche pas à se faire passer pour un humain).

# SYSTEME EXPERT (SE)

Les systèmes experts (S.E) viennent des recherches en Intelligence Artificielle (I.A).

Ils reproduisent certains raisonnements humains comme :

- les syllogismes,
- la déduction formelle.



# RAISONNEMENT FORMEL (1)

*Le syllogisme (logique aristotéliscienne) :*

*"Tous les Hommes sont Mortels",*

**or** *"les Grecs sont des Hommes",*

**donc** *"les Grecs sont Mortels"*

C'est un modèle de raisonnement de type :

$\{H \text{ est } \mathbf{M}\} \text{ or } \{\mathbf{G} \text{ est } H\} \text{ donc } \{G \text{ est } M\}.$

# RAISONNEMENT FORMEL (2)

- Les systèmes experts utilisent un raisonnement logique basé sur une règle d'inférence comme les syllogismes. C'est le ***modus ponens*** ou chaînage avant (fonctionnant de la sorte :

{Si A est vrai} et {Si A implique B},  
alors on peut déduire que {B est vrai}.

*Nota : les systèmes experts font aussi du chaînage arrière*

# ARCHITECTURE D'UN SYSTEME EXPERT

Un système expert est composé :

- d'un moteur d'inférences (capable de déduire des connaissances).

**Il est générique c'est celui que vous allez devoir coder avec deux modes :**

- Le chaînage avant
- Le chaînage arrière

- d'une base de règles (expertise d'un domaine).  
**Vous devez produire un exemple adapté à une situation.**

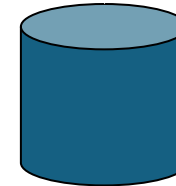
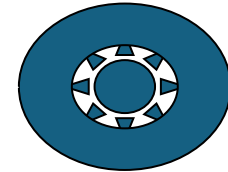
- d'une base de faits (les données)

**A noter que les règles sont formées d'une ou plusieurs hypothèses devant toutes être vérifiées et d'une seule conclusion (qui devient vraie si toutes les hypothèses sont vérifiées).**

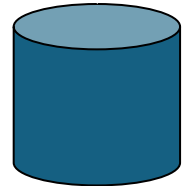
**INTERFACE UTILISATEUR**



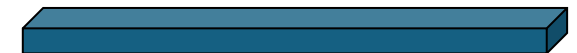
**MOTEUR  
D'INFERENCE**



**FAITS**



**REGLES**



**INTERFACE EXPERT**

# FONDEMENTS DES SYSTEMES EXPERTS

Le type de raisonnement couramment reproduit par les systèmes experts est le raisonnement logique basé sur une règle d'inférence fonctionnant de la sorte : "Si A est vrai" et "Si A implique B", alors on peut déduire "B est vrai".

Ce sont des systèmes formels basés sur la logique : exemple de règles d'inférences

- **Modus ponens (chaînage avant)**

Si A et  $(A \rightarrow B)$  Alors on **déduit** B (noté  $A, (A \rightarrow B) \vdash B$ )

- **Modus tollens (chaînage arrière)**

Si non B et  $(A \rightarrow B)$  Alors on **déduit** non A

- **Enchaînement**

Si  $A \rightarrow B$  et  $B \rightarrow C$  alors on **déduit**  $A \rightarrow C$

# FONDEMENTS DES SYSTEMES EXPERTS (SUITE)

*On rappelle que la proposition logique "A implique B" ne prouve pas que B est vrai. Pour cela, il faut que la prémisse de l'implication soit vraie.*

- Si la règle d'inférence ne produit que des phases vraies : la théorie est dite **cohérente**,
- Si la règle d'inférence produit toutes les phases vraies : la théorie est dite **complète**,
- Enfin une théorie complète est **décidable** s'il existe un algorithme qui permet de déterminer si un énoncé quelconque est (logiquement) vrai ou non.

# COMPREHENSION DES MECANISMES D'INFERENCE

1. L'exemple proposé permet de comprendre le mode opératoire du **chainage avant** (à vous de le faire)
2. Il permet aussi de comprendre le mode opératoire du **chaînage arrière**.
  - Il montre que la recherche d'une solution oblige le moteur d'inférences à faire des **tentatives**.
  - En effet l'arbre et/ou et parcouru jusqu'à ce qu'une solution soit trouvée.

# UN EXEMPLE SIMPLE

- **Règles**

b d e -> f;

g d -> a;

f c -> a;

b -> x;

e -> d;

a x -> h;

c -> d;

x c -> a;

x b -> d;

- **Faits**

b;

c;

A noter que la syntaxe des règles et des faits devra suivre la syntaxe présentée :

- **Hypothèses** séparées par des espaces,
- **Conclusion** unique (derrière la flèche ->).
- Fin de la règle avec un point-virgule (comme en C!)

# COMPREHENSION DES MECANISMES D'INFERENCE : CHAÎNAGE AVANT

Faire l'exercice par groupe :

Toutes les règles dont toutes les hypothèses sont des faits conduisent à des faits déduits

Déduire tous les faits possibles à partir des faits initiaux.

Prouver que  $h$  est vrai.

Expliquer le raisonnement, c'est-à-dire les règles utilisées.

Proposer un algorithme itératif ou récursif.

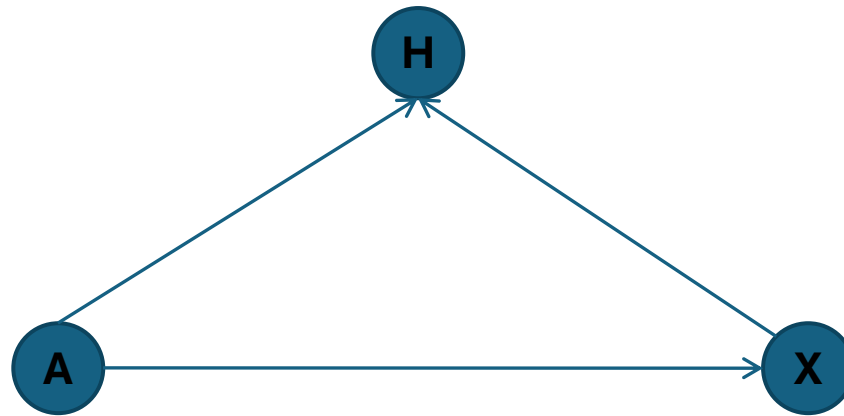


# COMPREHENSION DES MECANISMES D'INFERENCE : CHAÎNAGE ARRIERE

Exemple sur la base de connaissance :

- Démontrer que H est vrai

Suivi de l'algorithme de chaînage arrière

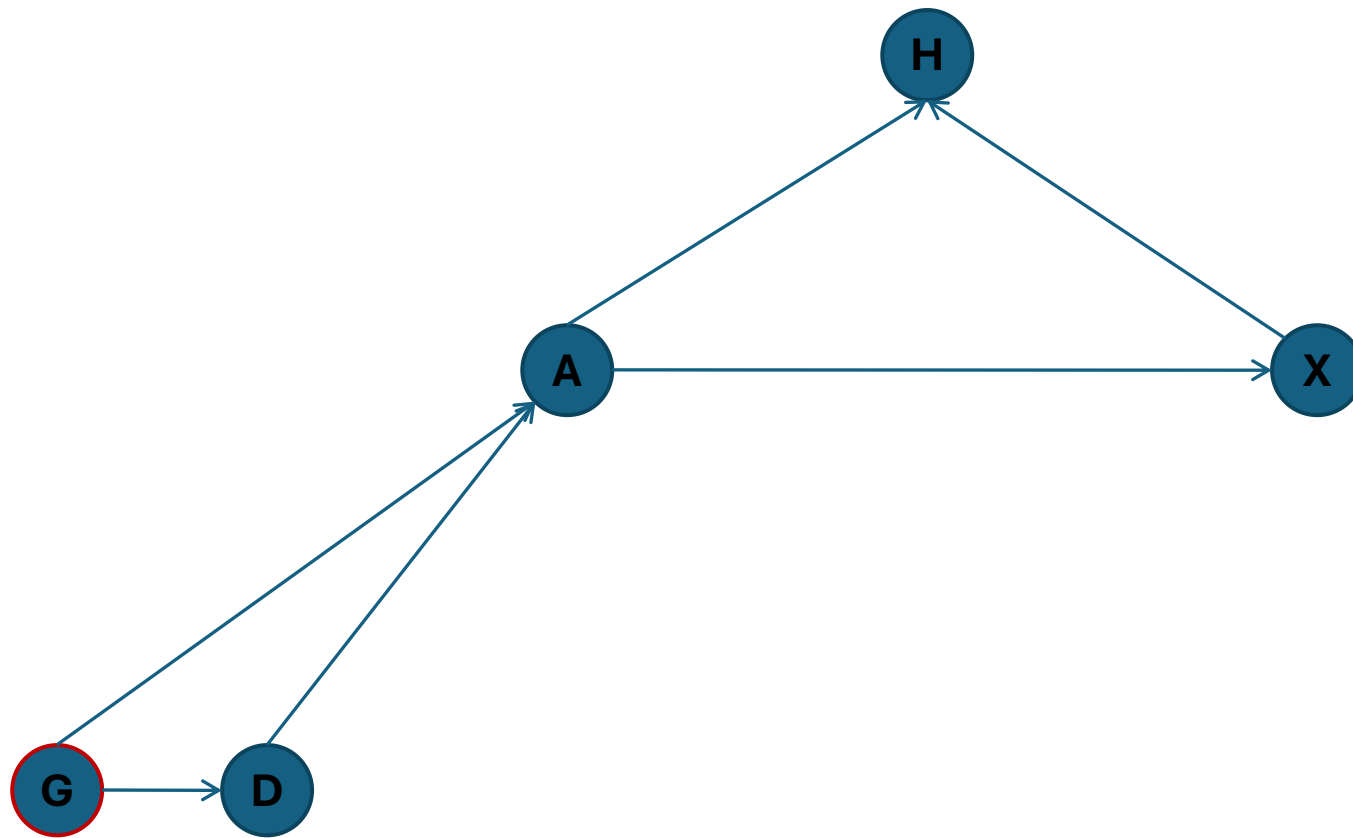


### Règles

`b d e -> f;`  
`g d -> a;`  
`f c -> a;`  
`b -> x;`  
`e -> d;`  
`a x -> h;`  
`c -> d;`  
`x c -> a;`  
`x b -> d;`

### Faits

`b;`  
`c;`

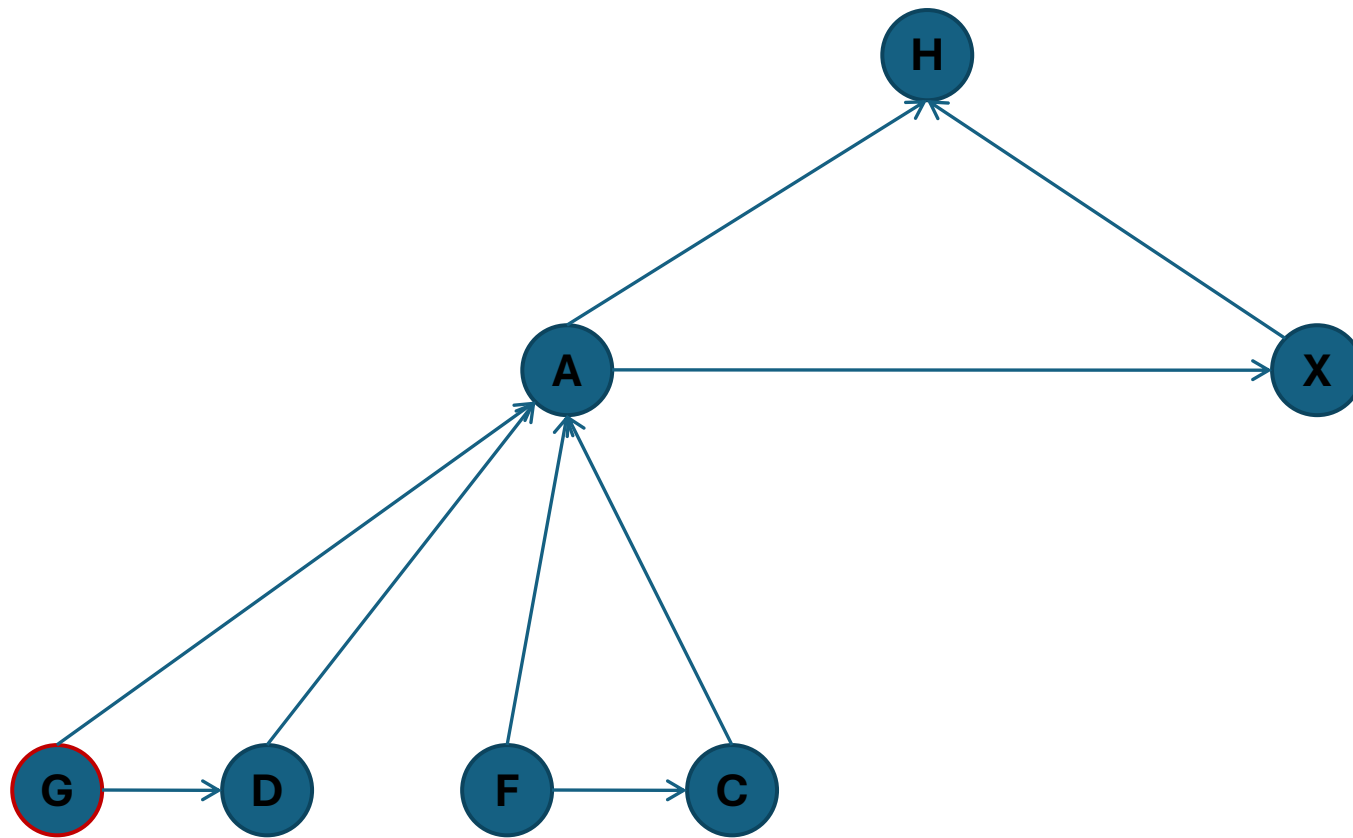


### Règles

**b d e -> f;**  
**g d -> a;**  
**f c -> a;**  
**b -> x;**  
**e -> d;**  
**a x -> h;**  
**c -> d;**  
**x c -> a;**  
**x b -> d;**

### Faits

**b;**  
**c;**

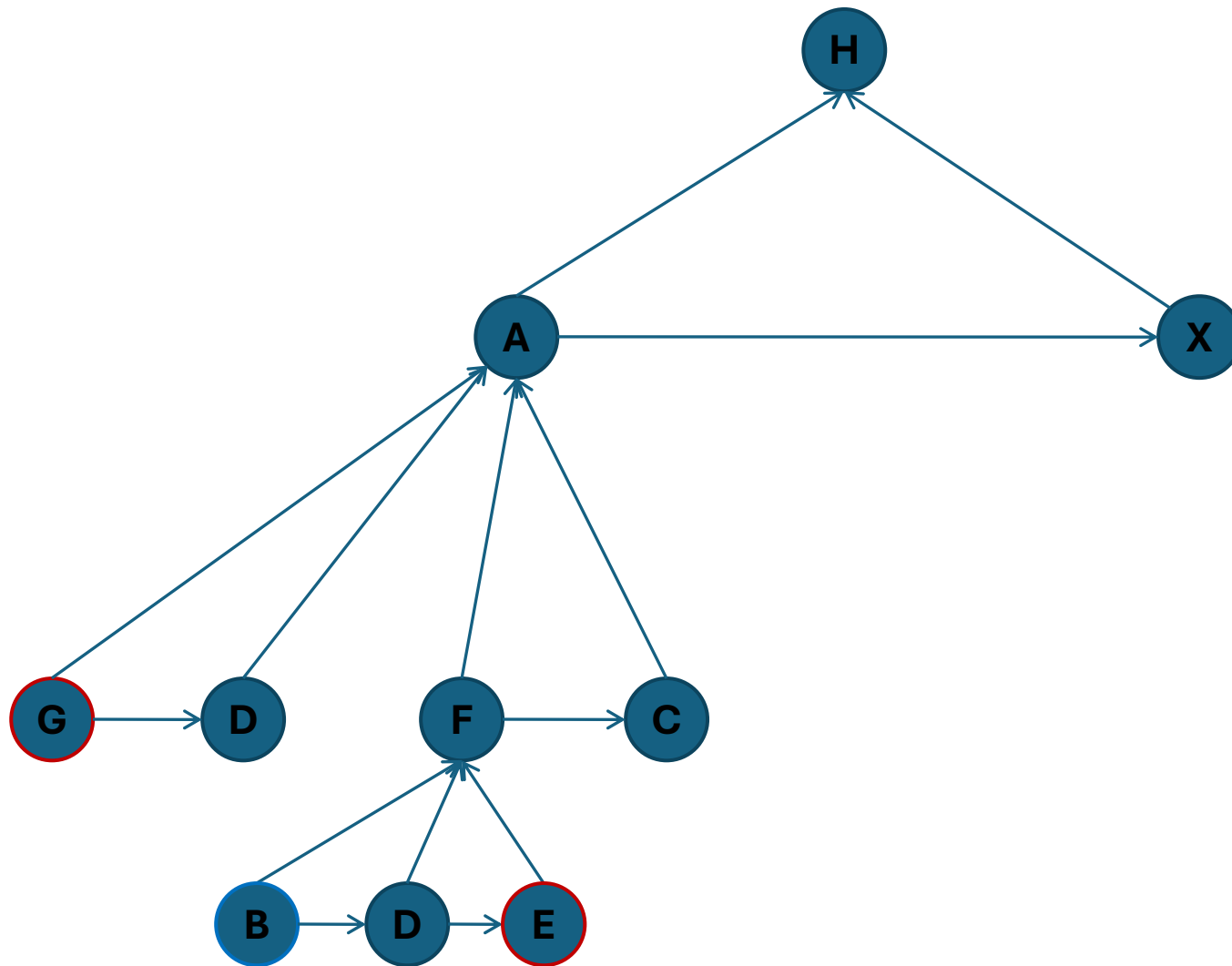


### Règles

$b \ d \ e \rightarrow f;$   
 $g \ d \rightarrow a;$   
 $f \ c \rightarrow a;$   
 $b \rightarrow x;$   
 $e \rightarrow d;$   
 $a \ x \rightarrow h;$   
 $c \rightarrow d;$   
 $x \ c \rightarrow a;$   
 $x \ b \rightarrow d;$

### Faits

$b;$   
 $c;$

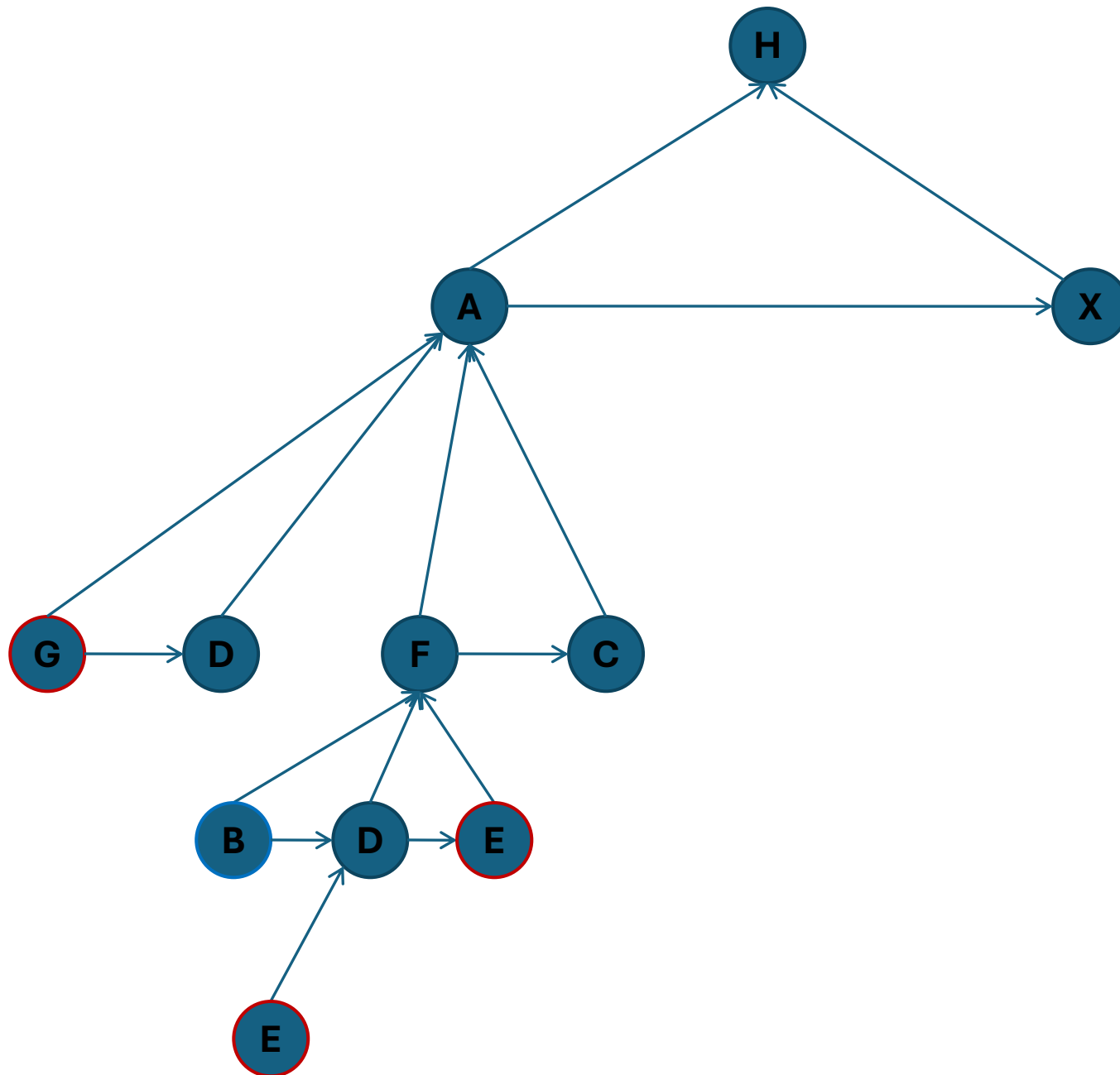


### Règles

$b \ d \ e \rightarrow f$ ;  
 $g \ d \rightarrow a$ ;  
 $f \ c \rightarrow a$ ;  
 $b \rightarrow x$ ;  
 $e \rightarrow d$ ;  
 $a \ x \rightarrow h$ ;  
 $c \rightarrow d$ ;  
 $x \ c \rightarrow a$ ;  
 $x \ b \rightarrow d$ ;

### Faits

$b$ ;  
 $c$ ;

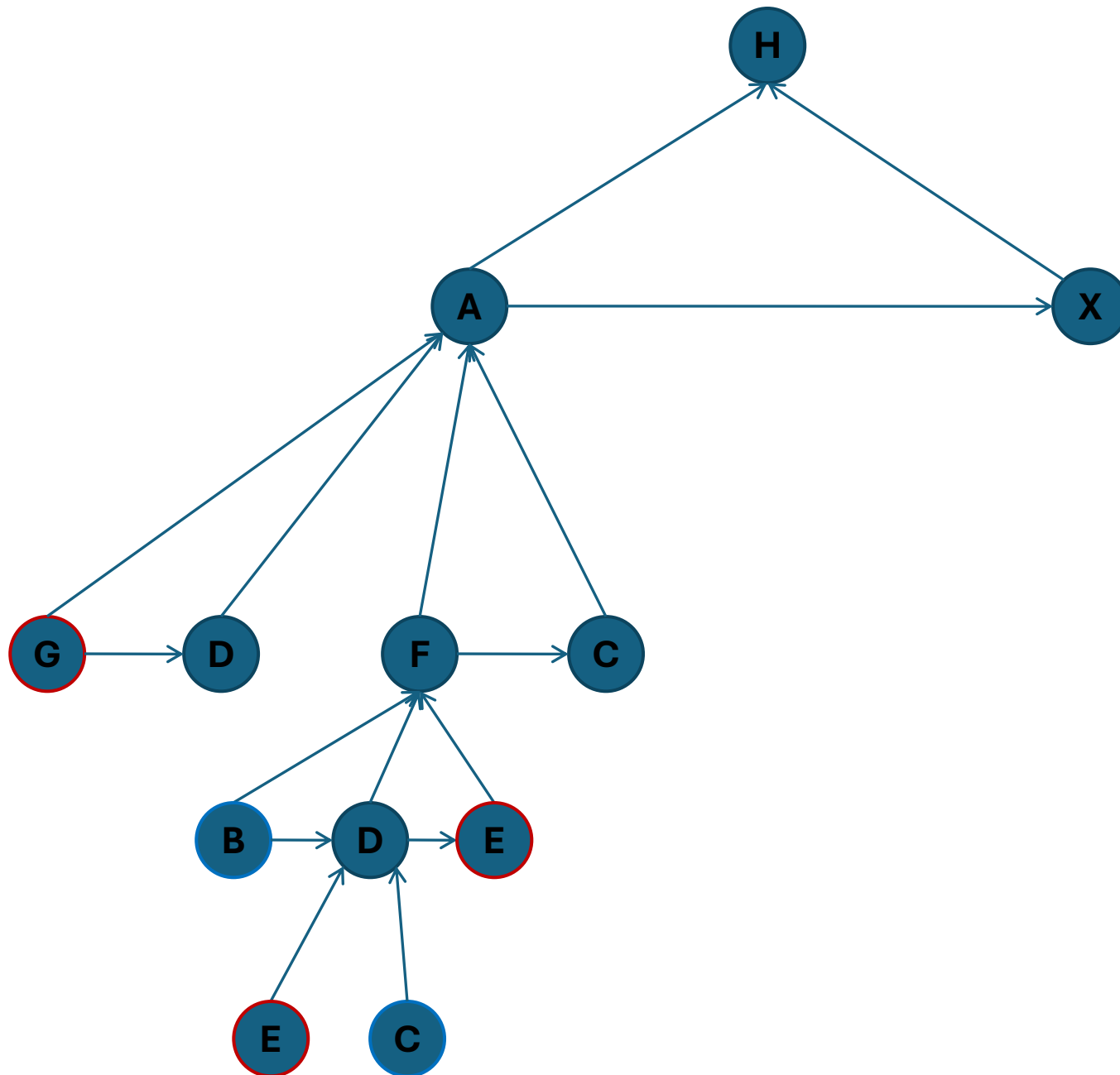


### Règles

$b \ d \ e \rightarrow f;$   
 $g \ d \rightarrow a;$   
 $f \ c \rightarrow a;$   
 $b \rightarrow x;$   
 $e \rightarrow d;$   
 $a \ x \rightarrow h;$   
 $c \rightarrow d;$   
 $x \ c \rightarrow a;$   
 $x \ b \rightarrow d;$

### Faits

$b;$   
 $c;$

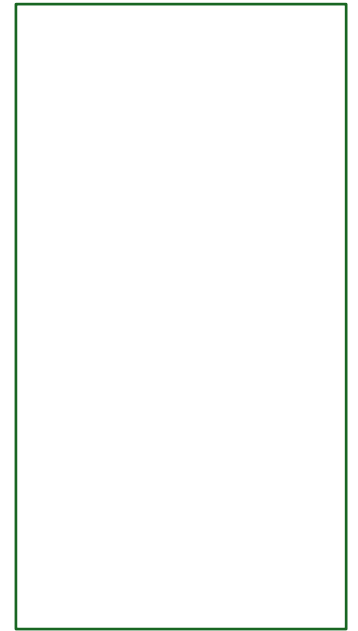
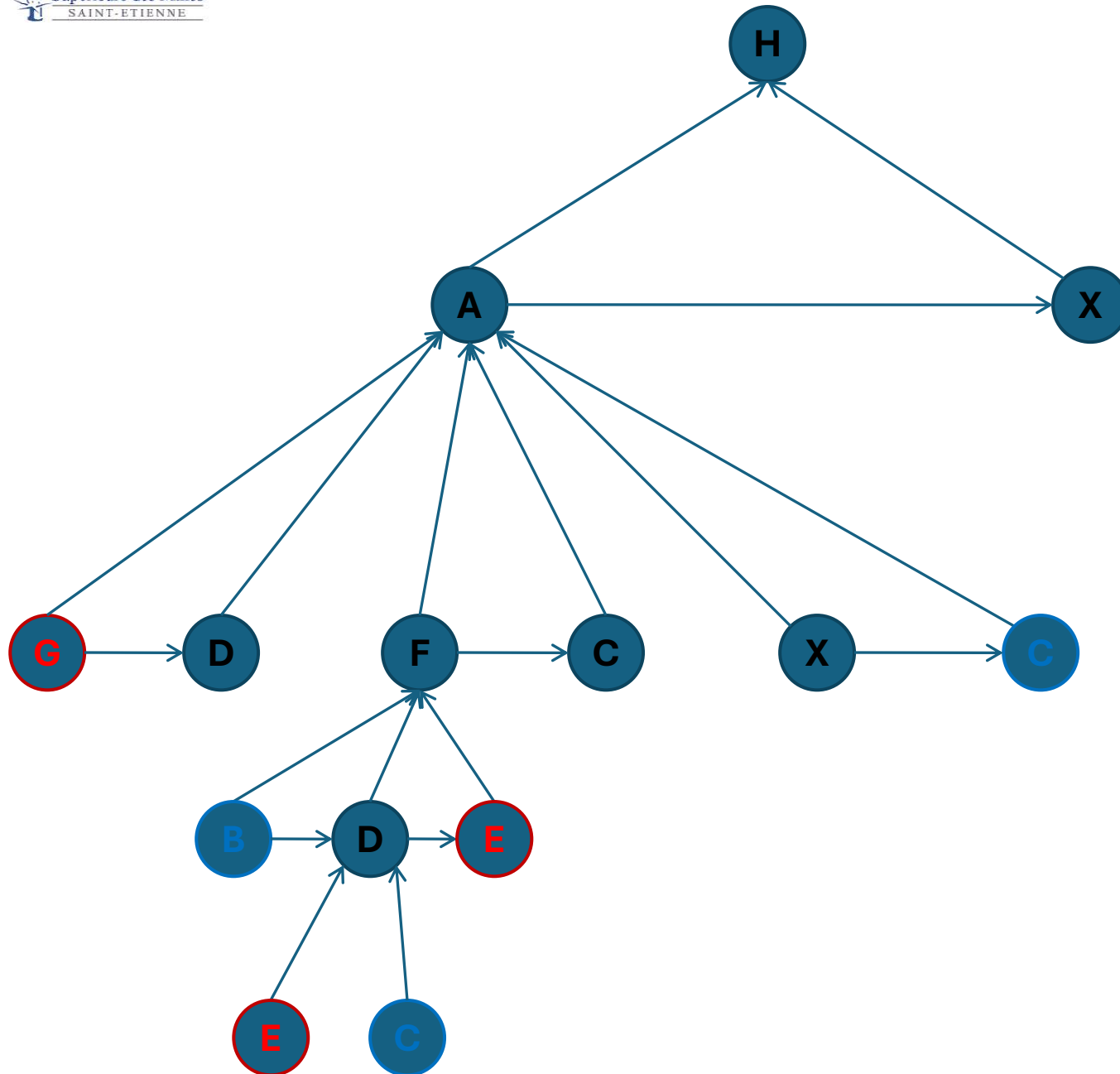


### Règles

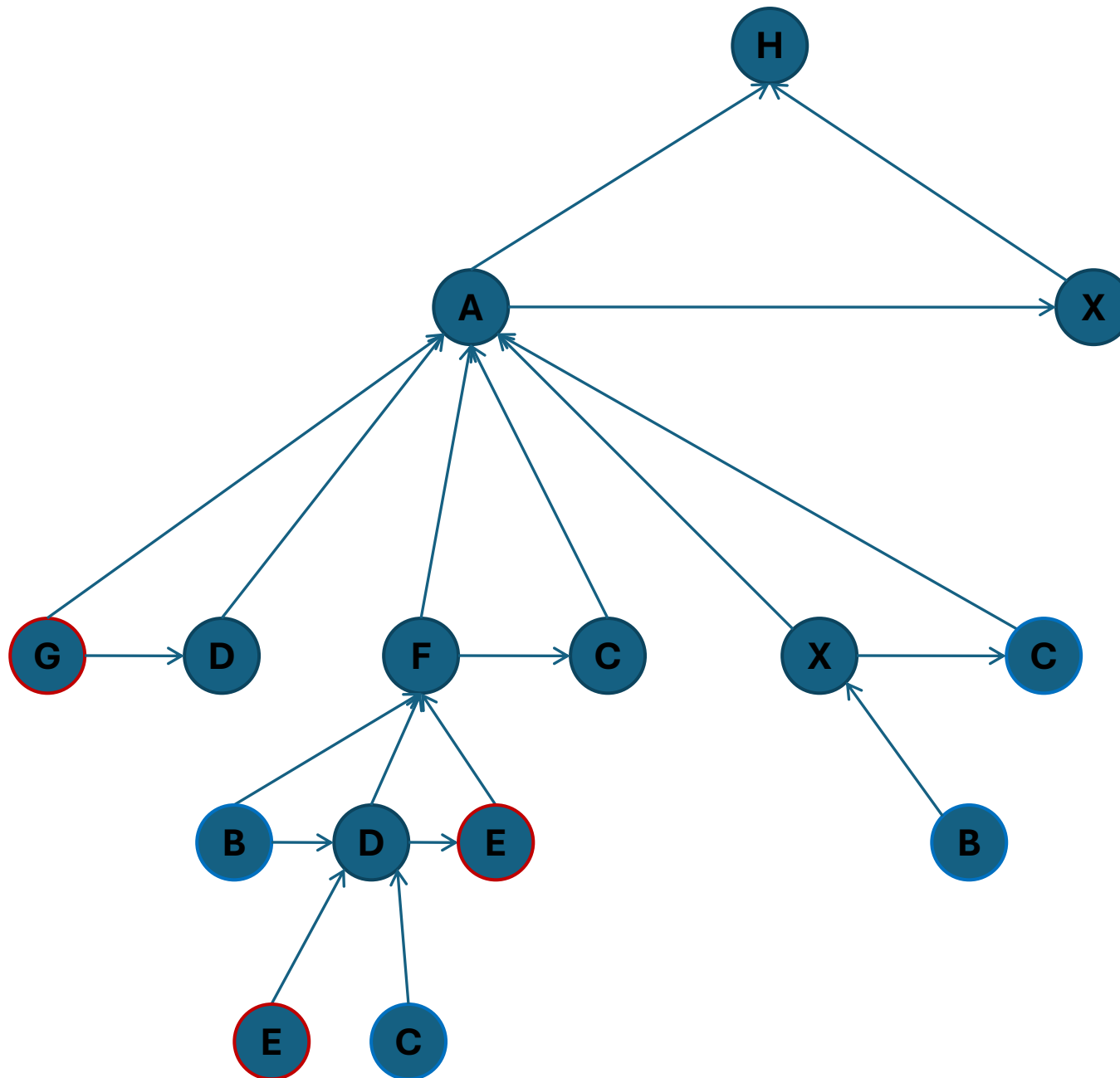
$b \ d \ e \rightarrow f$ ;  
 $g \ d \rightarrow a$ ;  
 $f \ c \rightarrow a$ ;  
 $b \rightarrow x$ ;  
 $e \rightarrow d$ ;  
 $a \ x \rightarrow h$ ;  
 $c \rightarrow d$ ;  
 $x \ c \rightarrow a$ ;  
 $x \ b \rightarrow d$ ;

### Faits

$b$ ;  
 $c$ ;





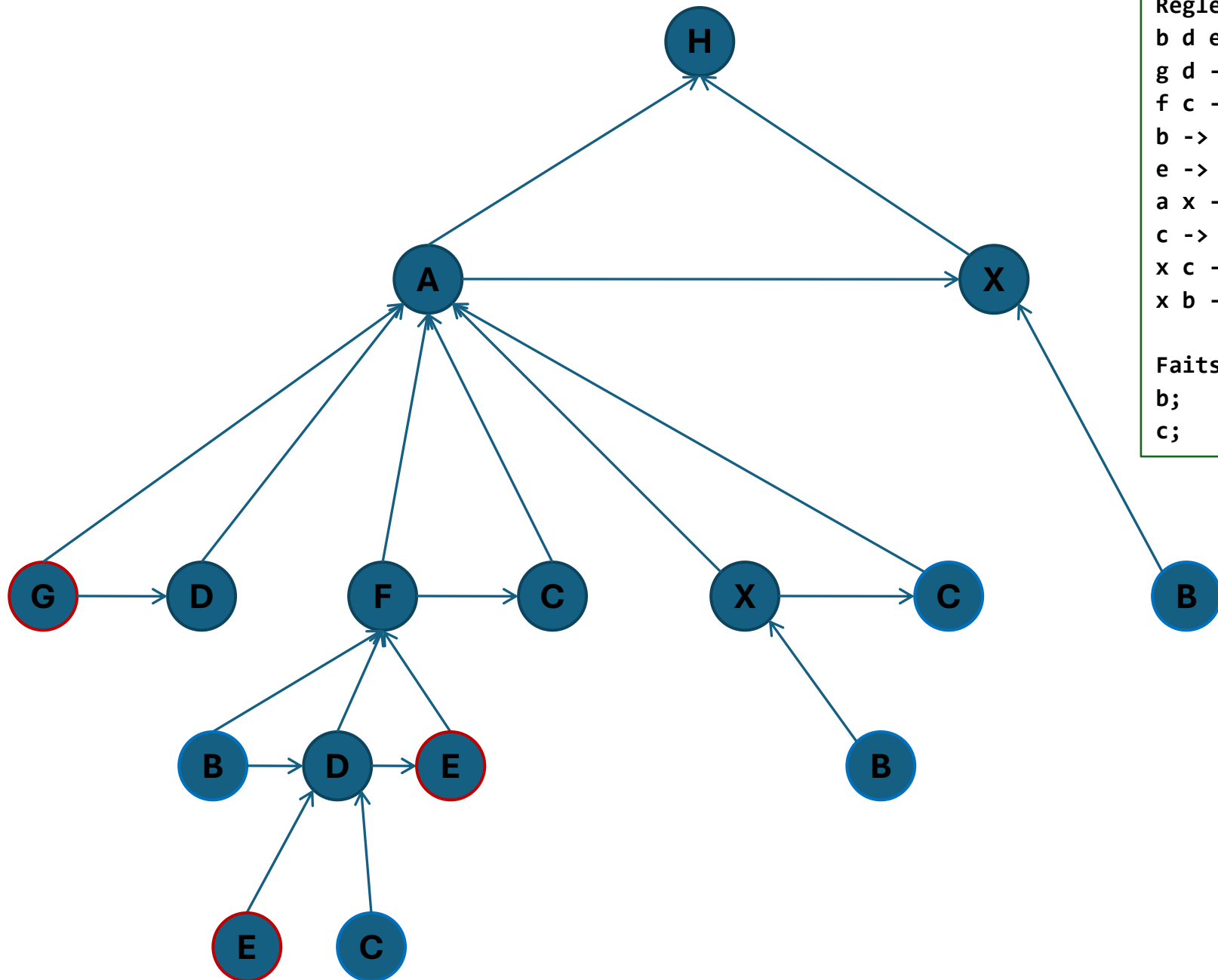


### Règles

$b \ d \ e \rightarrow f;$   
 $g \ d \rightarrow a;$   
 $f \ c \rightarrow a;$   
 $b \rightarrow x;$   
 $e \rightarrow d;$   
 $a \ x \rightarrow h;$   
 $c \rightarrow d;$   
 $x \ c \rightarrow a;$   
 $x \ b \rightarrow d;$

### Faits

$b;$   
 $c;$



### Règles

$b \ d \ e \rightarrow f;$   
 $g \ d \rightarrow a;$   
 $f \ c \rightarrow a;$   
 $b \rightarrow x;$   
 $e \rightarrow d;$   
 $a \ x \rightarrow h;$   
 $c \rightarrow d;$   
 $x \ c \rightarrow a;$   
 $x \ b \rightarrow d;$

### Faits

$b;$   
 $c;$

## Chaînage arrière (But, Base-de-règles, Base-de-faits)

### Début

Résultat  $\leftarrow$  faux

**Si** But n'est pas dans Base-de-faits **alors**

{ Règle = première(Base-de-règles)

**Tant que** Base-de-règles non vide **et** Résultat = faux **faire**

{ **Si** Conclusion (Règle) = But **alors**

{ */\*Vérifier que toutes les Hypothèses peuvent être vérifiées à partir de la Base-de-faits \*/*

Hypothèse  $\leftarrow$  première-hypothèse(Règle)

Continue  $\leftarrow$  vrai

**Tant que** Existe(Hypothèse) **et** Continue **faire**

{ Continue = Chaînage arrière (Hypothèse, Base-de-règles, Base-de-faits)

Hypothèse  $\leftarrow$  hypothèse-suivante(Règle)

**} fin tant que**

*/\* Si toutes les Hypothèses sont vérifiées la conclusion est vérifiée*

*Sinon la conclusion n'est pas vérifiée \*/*

Résultat = Continue

**} fin si**

Règle  $\leftarrow$  suivante(Base-de-règles)

**}fin tant que**

**} Sinon** */\* But est dans Base-de-faits \*/*

Résultat  $\leftarrow$  vrai

**Retourne** (Résultat)

**Algorithme de  
chainage arrière  
en pseudo-code**

### Fin

# POUR AUJOURD'HUI...

**L'équipe-projet choisit un sujet (une thématique d'expertise) et s'organise**

1. Mettez en forme l'algorithme de chainage avant
2. Réfléchissez au sujet de votre base de connaissance (diagnostic de pannes, météo, arbre de décision...)
3. Organiser les tâches à réaliser et planifier les réalisations...

# FONCTIONNALITES A METTRE EN OEUVRE

- Lecture (et intégration dans la base de règles) de fichiers de règles au format donné.
- Saisie de règles (intégration dans la base de règle)
- Saisie de faits (intégration dans la base de faits)
- Enregistrement de la base de règles dans un fichier texte (extension kbs)
- Appel de la fonctionnalité chainage avant, avec la question "**que voulez-vous démontrer ?**"
- Appel de la fonctionnalité chainage arrière, avec la question "**prouver que XXX est vrai**"

# PROCEDER PAR ETAPES

- Commencer par le projet *a minima*.
- Vous **devez** présenter un exécutable qui répond strictement au cahier des charge.
- Vous **pouvez** faire une version adaptée au problème spécifique que vous traitez.



# TYPE D'INTELLIGENCE ARTIFICIELLE

- IA générative (Machine Learning) : capable de résoudre tout type de problème pour peu qu'elle ait été entraînée. **Mais consommatrice de ressources, non stable pour les résultats, ne peut pas expliquer son raisonnement.**
- IA Symbolique (Système Expert) : permet des raisonnements de type déduction ou induction. **Base de connaissance doit être alimentée par un expert. Sait expliquer son raisonnement.**

# PROJET = ŒUVRE

- Le résultat peut être éloigné de l'objectif initial.
- Sur un projet normal cette dérive est interdite (coût).
- Sur le projet IA c'est encouragé dès lors que vous avez **(une version qui) rempli(t) le Cahier des charges.**



# AJOUT DE NOUVELLES FONCTIONNALITES

- Les fonctionnalités minimales sont inscrites au cahier des charges.
- Certains projets intègrent :
  - un module interactif (questions/réponses) avec l'utilisateur,
  - la trace (preuve de démonstration)