Modes de représentation et de raisonnement en théorie des possibilités quantitative.

Khellaf-Haned Hadja.Faiza

Motivations

- Connaissances hétérogènes ,
- Connaissances multi-sources (cas des applications multi-agents),
- Les connaissances sont représentées dans diffèrents formats.

$\Rightarrow$ Fusion

Motivations

Théorie des possibilités

-

Modes de représentation

- Représentation logique
- Représentation graphique

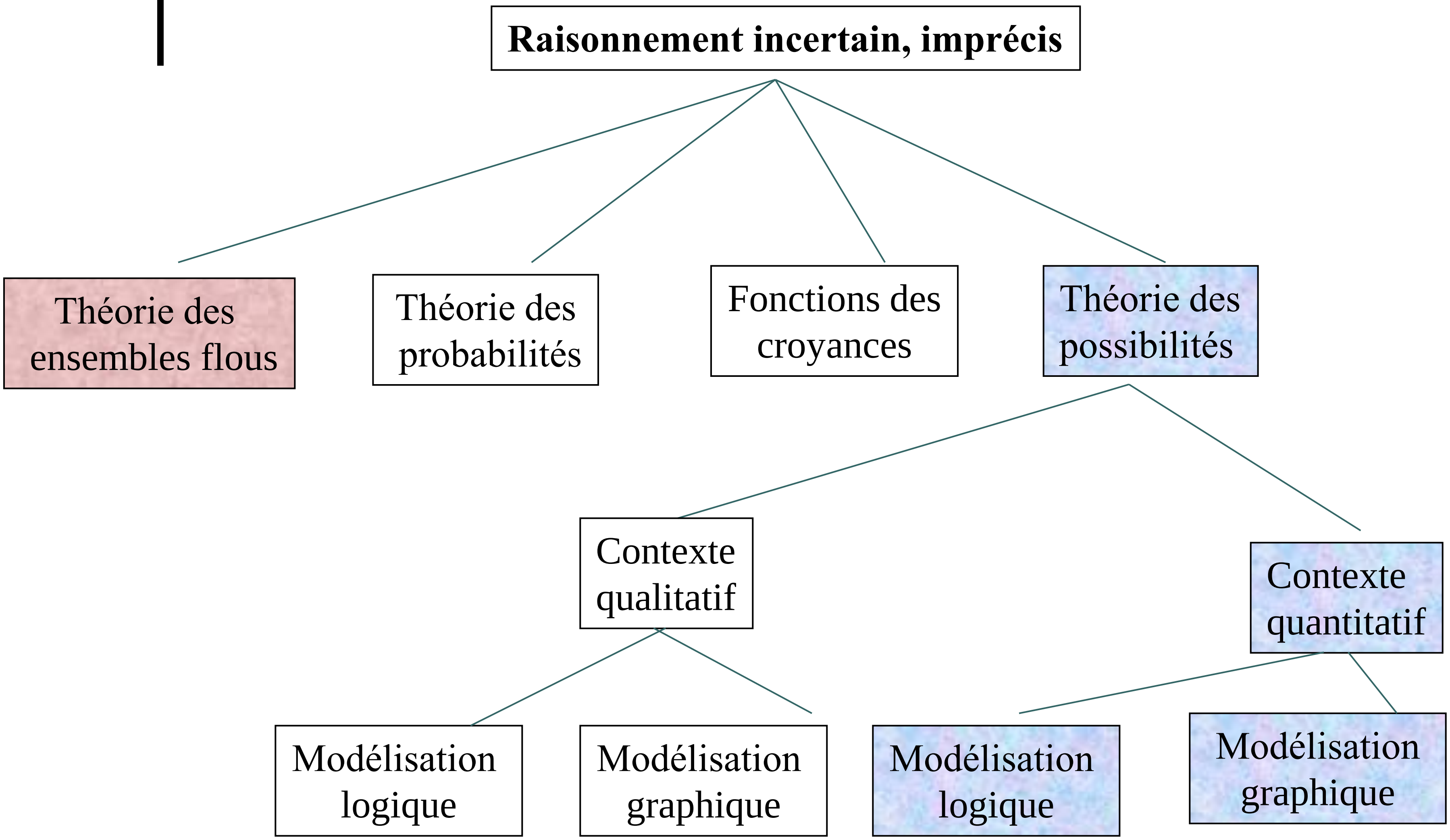
Différents outils de raisonnement:

- Extension de la logique classique
- Counterpartie des réseaux causaux bayésiens probabilistes

Plan

- 1-Introduction
- 2-Concepts de bases de la théorie des possibilités
  - 2-1 Modèle logique
  - 2-2 Modèle graphique
- 3- Transformations en théorie des possibilités quantitative
  - 3-1 d'un graphe causal vers une base
  - 3-2 d'une base vers un graphe causal
  - 3-3 base à pénalités et Bases possibiliste
  - 3-4 réseaux Kappa et réseaux possibilistes
- 4- Modes de raisonnement en théorie des possibilités quantitative
- 5- Implémentation
- 6- Expérimentation
- 7- Conclusion et perspectives

1-Introduction



- Montrer l'équivalence entre:

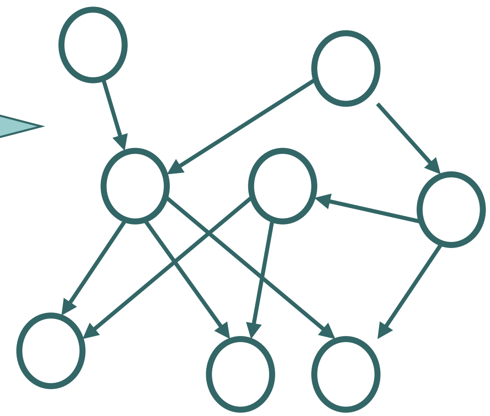
Représentations logiques

(φ_1, α_1)

⋮

(φ_n, α_n)

Représentations Graphiques



- Développement d'un algorithme d'inférence

2-Théorie des possibilités: Concepts de base

Notations:

- $\mathcal{L}$ un langage propositionnel fini
- Ω l'ensemble des interprétations associées à $\mathcal{L}$.
- $\varphi, \Psi, \ldots$ des formules propositionnelles.
- ω est une interprétation.
- $\omega \models \varphi$ ou $\omega \in [\varphi]$ signifie que ω satisfait φ.

2-Théorie des possibilités: Concepts de base

- **Une distribution possibiliste π** $: \Omega \rightarrow [0,1]$.
  - $\pi(\omega)=0$ ω est impossible
  - $\pi(\omega)=1$ ω est totalement possible
  - $\pi(\omega) > \pi(\omega')$ ω est préférée à ω'
  - $\forall \omega, \pi(\omega) = 1$ Ignorance
  - $\exists \omega / \pi(\omega)=1$ Normalisation

- **La mesure de possibilité** d'une formule φ:
  - $\Pi(\varphi) = max\{ \pi(\omega) : \omega \in [\varphi] \}$

- **La mesure de nécessité** d'une formule φ:
  - $\mathcal{N}(\varphi) = 1 - \Pi(\neg\varphi)$

2- Théorie des possibilités: Modèle logique

- Une base de connaissances possibiliste Σ est un ensemble de formules pondérées :

$$\Sigma = \{(\varphi_i, \alpha_i)\ i=1,n\},$$

où φ_i est une formule propositionnelle

et $\alpha_i [0,1]$ représente le degré de certitude minimal de φ_i.

Exemple:

$$\Sigma=\{(\neg b,0.7),(c,0.3),(\neg a\vee \neg b\vee \neg c,0.4), (a\vee \neg b,0.9) , (d\vee \neg a,0.6)\}$$

2-Théorie des possibilités: Modèle logique

- Distribution de possibilité associée à:

  - une formule élémentaire (φ_i,α_i):

$$\forall\ \omega \in \Omega,\ \pi_{(\varphi i,\alpha i)}(\omega) = \begin{cases} 1-\alpha_i & \text{si } \omega \in [\varphi_\iota] \\ 1 & \text{sinon} \end{cases}$$

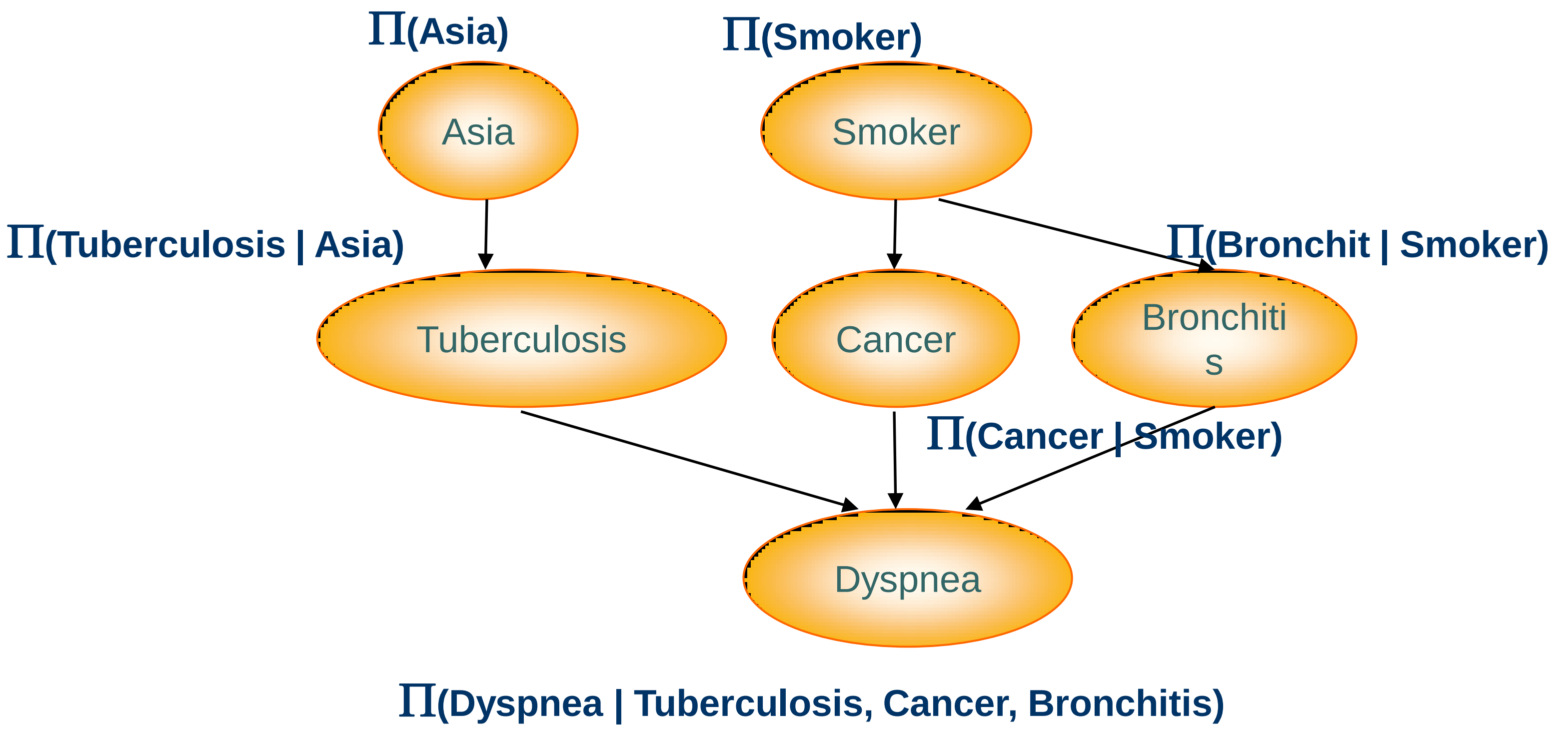
  - une base possibiliste quantitative Σ :

$$\forall\ \omega \in \Omega,\ \pi_\Sigma(\omega) = \begin{cases} 1 & \text{si } \forall\ (\varphi_i,\alpha_i) \in \Sigma,\ \omega \in [\varphi_\iota] \\ *\{(1-\alpha_i) : (\varphi_i,\alpha_i) \in \Sigma,\ \ \omega \notin [\varphi_\iota]\} & \text{sinon} \end{cases}$$

2-Théorie des possibilités: Modèle graphique

1. Composante Graphique (Directed Acyclic Graph)

2. Composante numérique : Possibilités conditionnelles



2-Théorie des possibilités: Modèle graphique

Règle de chaînage

$\omega=(a_1,a_2,\ldots.\ a_n)$, une interprétation.

1- Basée sur le produit:

$\pi_*(\omega)= *_{i=1..n}\Pi(a_i|u_i) \quad : \omega \in [a_i \wedge u_i]$

où a_i est l'instance de A_i

et u_i est l'instance des parents de A_i.

2- Basée sur le minimum:

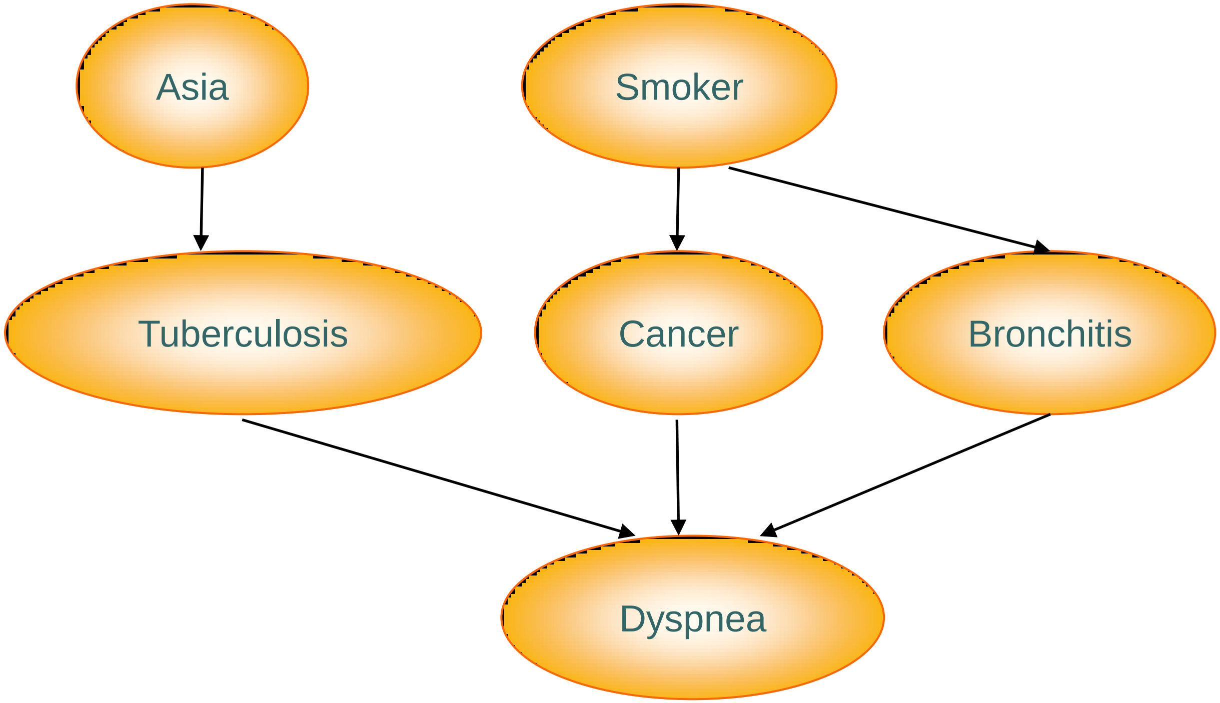
$\pi_m(\omega)= \min_{i=1..n} \Pi(a_i|u_i) \quad : \omega \in [a_i \wedge u_i]$

où a_i est l'instance de A_i

et u_i est l'instance des parents de A_i.

2-Théorie des possibilités: Modèle graphique

Exemple:



$\pi_*(a, s, \neg t, c, b, \neg d) = \Pi(a) * \Pi(s) * \Pi(\neg t \mid a) * \Pi(b \mid s) * \Pi(c \mid c) * \Pi(\neg d \mid \neg t, c, b)$

$\pi_m(a, s, \neg t, c, b, \neg d) = \min(\Pi(a), \Pi(s), \Pi(\neg t \mid a), \Pi(b \mid s), \Pi(c \mid c), \Pi(\neg d \mid \neg t, c, b))$

2-Théorie des possibilités: Modèle graphique

- Conditionnement basé sur le produit:

$$\pi_* (\omega|\varphi)= \begin{cases} \pi(\omega)/\ \Pi\ (\varphi) & \text{si } \omega\in[\varphi] \\ 0 & \text{sinon} \end{cases}$$

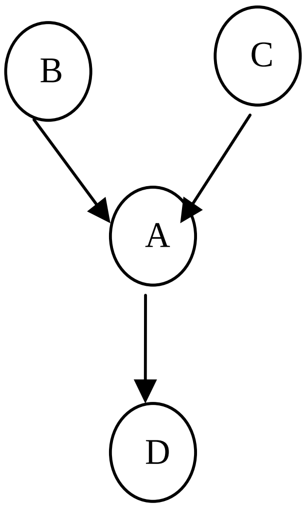
- Conditionnement basé sur le minimum:

$$\pi_m (\omega\ |\varphi)= \begin{cases} 1 & \text{si } \pi(\omega) = \Pi(\varphi) \text{ et } \omega\in[\varphi] \\ \pi(\omega) & \text{si } \pi(\omega) < \Pi(\varphi) \text{ et } \omega\in[\varphi] \\ 0 & \text{sinon} \end{cases}$$

2-Théorie des possibilités: Modèle graphique

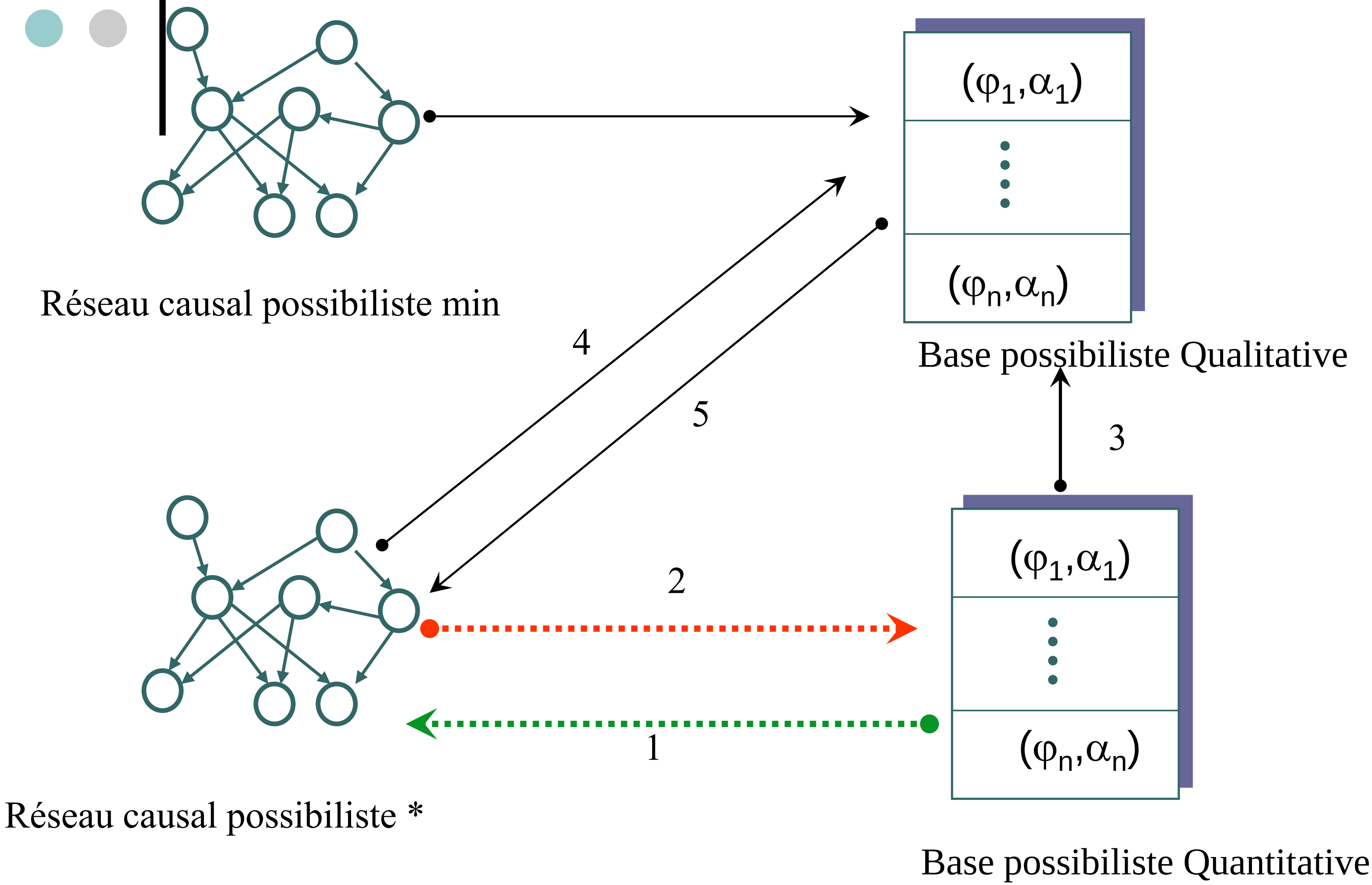
- <u>Réseau causal possibiliste</u>

  $\Pi G = \{(a_i,u_i, \alpha_i): \alpha_i = \Pi(a_i|u_i) \neq 1\}$

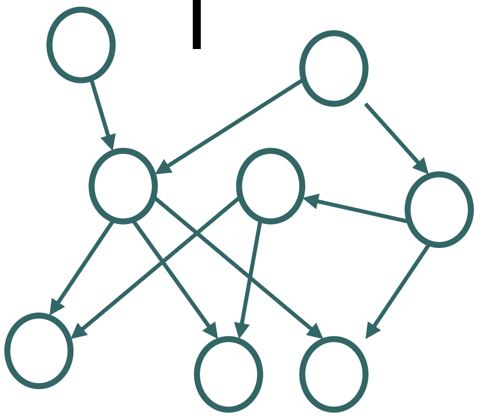


$\Pi G=\{(b,\emptyset,0.3), (\neg c,\emptyset,0.7), (a, b,c,0.6), (\neg a, b , 0.1), (\neg d, a, 0.4)\}$

Problématique:



3- Les transformations



(φ_1, α_1)

$\vdots$

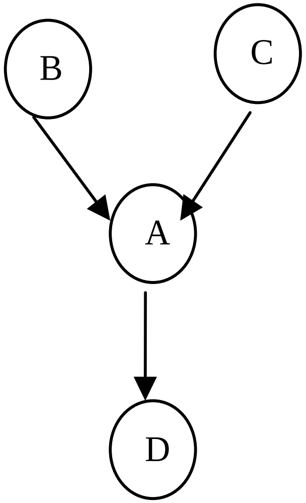
(φ_n, α_n)

$$\prod(a_i \mid u_i) = \alpha_i$$

$$(\neg\, a_i \vee \neg u_i) = 1 - \alpha_i$$

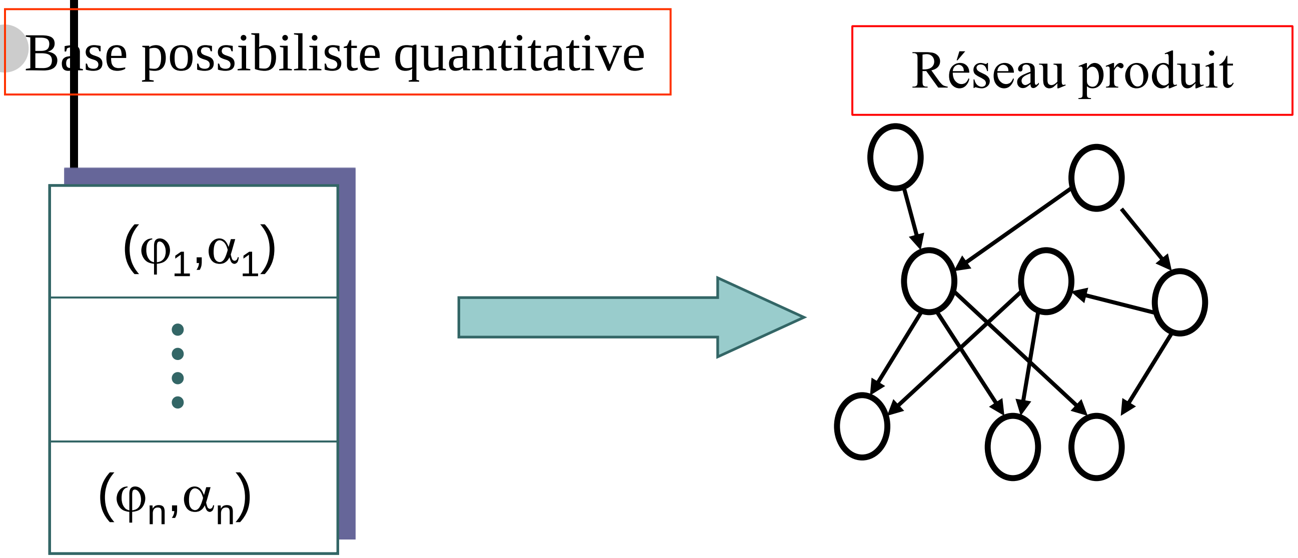
3- Les transformations

$\Pi G=\{(b,\varnothing,0.3), (\neg c,\varnothing,0.7), (a, b,c,0.6), (\neg a, b, 0.1), (\neg d, a, 0.4)\}$



$\Sigma_{DAG}=\{(\neg b,0.7),(c,0.3),(\neg a\vee \neg b\vee \neg c,0.4), (a\vee \neg b,0.9), (d\vee \neg a,0.6)\}$

3- Les transformations



Etape 1 - Suppression des tautologie
- Réductions

Etape 2 - Construction du graphe à partir d'un ordre arbitraire des variables.

Etape 3 - Détermination des possibilités conditionnelles:

3- Les transformations

<u>Etape 1:</u>

- Suppression des tautologies

- Réductions

$$(a \vee p, \alpha) \text{ et } (a \vee \neg p, \alpha)$$

$$\downarrow$$

$$(a, \alpha)$$

3- Les transformations

Etape 2:

Ordonner les $A_1, A_2, \ldots. A_n$

$\downarrow$

$\text{Parents}(A_i) \subseteq \{ A_{i+1} \ldots A_n \}$

$\downarrow$

Décomposer Σ en $\Sigma_{A1}, \Sigma_{A2} \ldots, \Sigma_{An}$

- $\Sigma_{A1} \subseteq \Sigma$ contient instances de A_1
- $\Sigma_{A2} \subseteq \Sigma - \Sigma_{A1}$ contient instances de A_2 , ...
- $\Sigma_{An} \subseteq \Sigma - \{\Sigma_{A1} \ldots \Sigma_{An-1}\}$ contient instances de A_n.

Etape 2:

Principe:

A partir de Σ_{A_i} calculer $\prod(a_i \mid u_i)$

Problème:

$\prod(a_i \mid u_i)$ est donnée pour toute instance de a_i et pour toute instance de Parent(A_i)

Transformer Σ_{A_i}

3- Les transformations

Etape 2: b-Extension (parents)

(a, α)

$\downarrow$

$(a \vee p, \alpha) \wedge (a \vee \neg p, \alpha)$

c-Redondances

$(a, \alpha) \wedge (a, \beta)$

$\downarrow$

$(a, \alpha+\beta-\alpha*\beta)$

3- Les transformations

Etape 2:

d-Réduction (cas de poids diffèrents)

$(a \vee p, \alpha) \wedge (a \vee \neg\, p, \beta)$ avec $\alpha < \beta$

$(\neg a \vee p,((1-\alpha)-(1-\beta))/(1-\alpha)$ à Σ_{Ai}

(p, α) à Σ

Etape 3 : calcul des possibilités conditionnelles

A partir des Σ_{Ai}

$$\Pi(a|u) = \begin{cases} 1-\alpha_i & \text{si } (\neg a \vee \neg u, \alpha_i) \in \Sigma_{Ai} \\ 1 & \text{sinon} \end{cases}$$

3- Les transformations

Exemple:

$$\Sigma=\{\cancel{(a\vee b \vee c,0.2)},\ \cancel{(a\vee b \vee \neg c,0.2)},\quad \boxed{(a\vee b,0.2),}$$
$$(\neg a\vee d,0.5),\quad (\neg c\vee d,0.1),\cancel{(\neg a\vee b\vee a,1)}\}.$$

Etape1:

- Suppression des tautologies

- Réduction des formules (poids identiques)

3- Les transformations

$$\Sigma=\{(a\vee b,0.2), (\neg a\vee d,0.5), (\neg c \vee d ,0.1),(a \vee b \vee \neg d,0.8)\}$$

Etape 2:

a- - Ordonner les variables A, B, C

- Décomposer la base

$$\Sigma_A=\{(a\vee b,0.2), (\neg a\vee d,0.5),(a \vee b \vee \neg d,0.8)\}$$

$$\Sigma_B=\{\}$$

$$\Sigma_C=\{(\neg c \vee d ,0.1)\}$$

$$\Sigma_D=\{\}$$

Traitement de la variable A:

b-Extension:

Σ_A ={(a∨b,0.2), (¬ a∨d,0.5),(a ∨ b ∨ ¬ d,0.8)} d'où Par(A)={B,D}

Σ_A={(a∨b ∨ d,0.2), (a∨b ∨ ¬ d,0.2),(¬ a∨b ∨ d,0.5), (¬ a ∨ ¬ b∨d, 0.5),(a ∨ b ∨ ¬ d,0.8)}

c-Redondance:

Σ_A={(a∨b ∨ d,0.2), (a∨b ∨ ¬ d,0.2), (a ∨ b ∨ ¬ d,0.8), (¬ a∨b ∨ d,0.5), (¬ a ∨ ¬ b∨d, 0.5)}

Σ_A={(a∨b ∨ d,0.2), (a∨b ∨ ¬ d,0.84), (¬ a∨b ∨ d,0.5), (¬ a ∨ ¬ b∨d, 0.5)}

3- Les transformations

d-Réduction (cas de poids diffèrents)

$\Sigma_A=\{\boxed{(a \vee b \vee d, 0.2), (\neg a \vee b \vee d, 0.5)}, (a \vee b \vee \neg d, 0.84), (\neg a \vee \neg b \vee d, 0.5)\}$

$\downarrow$

$\Sigma_A=\{\boxed{(\neg a \vee b \vee d, 0.375)}, (a \vee b \vee \neg d, 0.84), (\neg a \vee \neg b \vee d, 0.5)\}$

$\Sigma_B=\{(b \vee d, 0.2)\}$

3- Les transformations

Etape 3:

$\Sigma_A=\{(\neg a \vee b \vee d, 0.375), (a \vee b \vee \neg d, 0.84), (\neg a \vee \neg b \vee d, 0.5)\}$
$\Sigma_B=\{(b \vee d, 0.2)\}$
$\Sigma_C=\{(\neg c \vee d, 0.1)\}$
$\Sigma_D=\{\}$

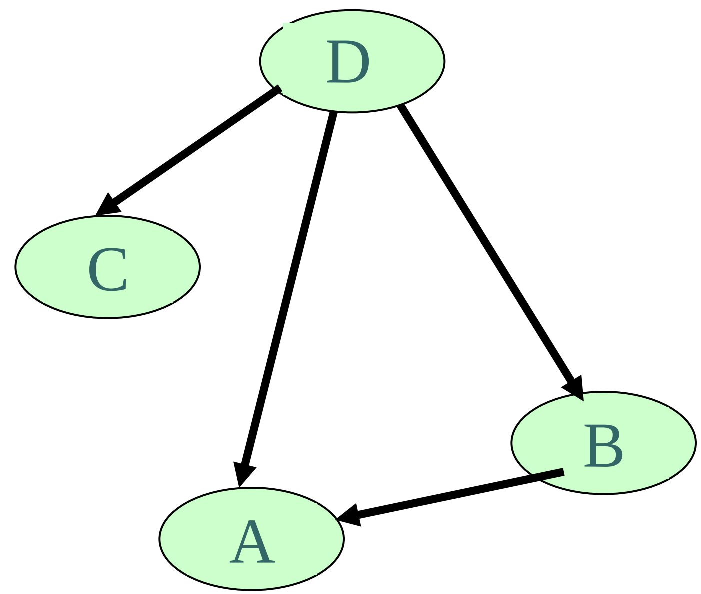
$$\Pi(a|u) = \begin{cases} 1-\alpha_i & \text{si } (\neg a \vee \neg u, \alpha_i) \in \Sigma_{Ai} \\ 1 & \text{sinon} \end{cases}$$

	d	¬d
b	1	1
¬b	1	.8

	d	¬d
c	1	.9
¬c	1	1

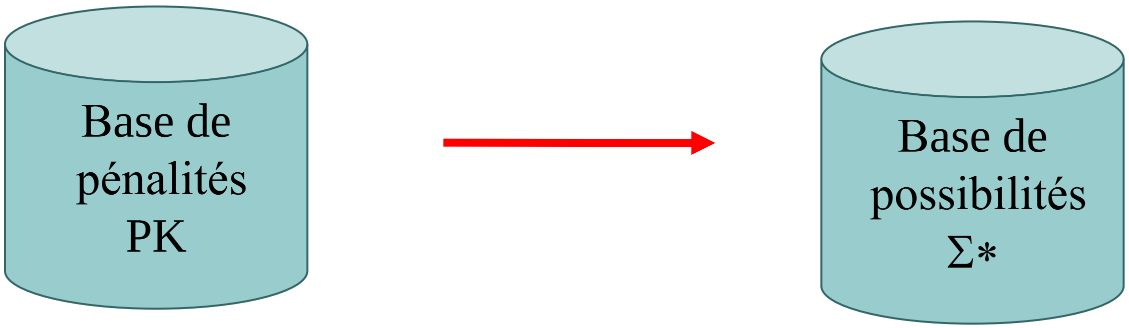
d	1
¬d	1

	b d	b¬d	¬b d	¬b ¬d
a	1	.5	1	.625
¬a	1	1	.16	1



3- Les transformations

3-3



- Base pénalités:

  $PK=\{(\varphi_i, \kappa_i)\ i=1..n\}$ où $k_i \in N \cup \{+\infty\}$

- Coût d'une interprétation:

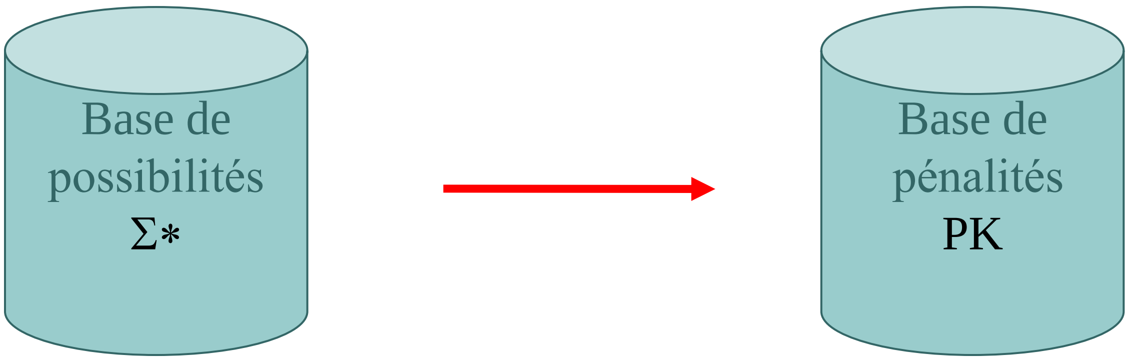
$$\kappa_{PK}(\omega)=\begin{cases} 0 & \text{si } \forall\ (\varphi_i, \kappa_i) \in P_K, \omega \in [\varphi_i] \\ \Sigma\kappa_i & \text{sinon} \end{cases}$$

- Relation syntaxique entre Σ^* et Σ_{PK}:

  $\Sigma_* = \{(\varphi_i, \alpha_i) : (\varphi_i, \kappa_i) \in PK \text{ et } \alpha_i = 1 - e^{-\kappa i}\}$

3- Les transformations

3-3



Définir des échelles d'incertitude spécifiques.

Une échelle possible est de la forme:

$\{e^{0}, e^{-1}, e^{-2}, \ldots, e^{-\infty}\}$

3-5 Réseaux possibilistes basés sur OCF

- Ordinal Conditional Function: $\Omega \rightarrow N$

$\forall \varphi \;\; \kappa(\omega)=\min\{\kappa(\varphi): \omega \in [\varphi]\}$
$\kappa(\varphi)$ représente le degré d'impossibilité de φ

- Conditionnement:

$\forall \omega \in \Omega, \;\; \kappa(\omega \mid p)= \kappa(\omega)- \kappa(p)$

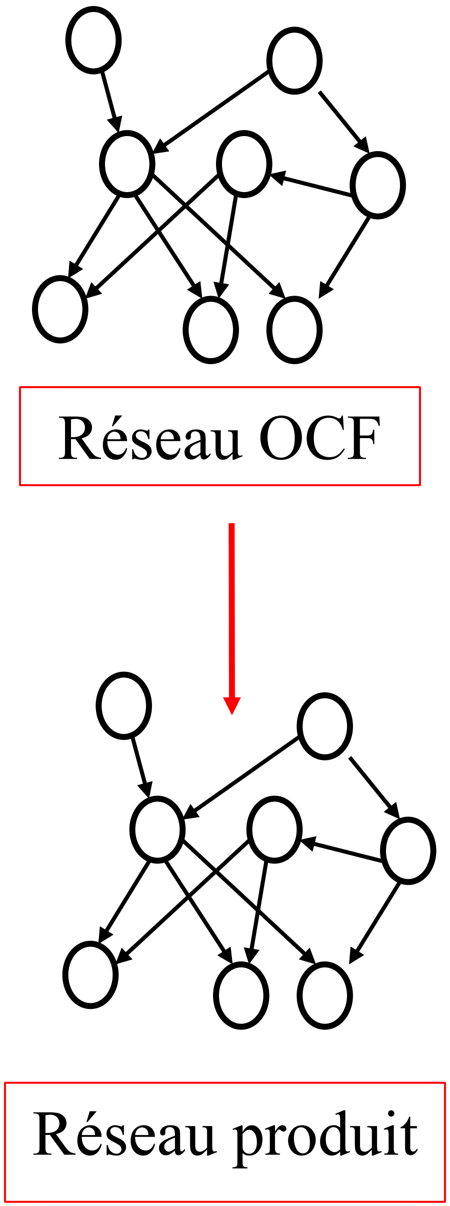
- Règle de chaînage:

$\kappa_G(\omega)= \quad \Sigma\{\kappa(a_i|u_i): \omega \in [a_i \wedge u_i], i=1\ldots n\}$
Où a_i est une instance de A_i
u_i est une instance de parents de A_i

3-5

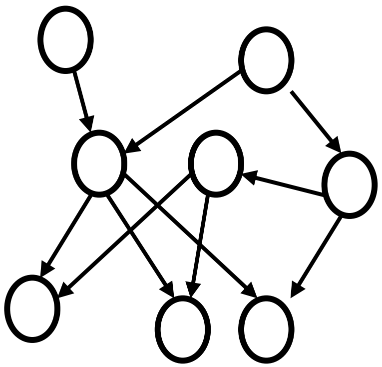
DAGs identiques

$\Pi(a_i|u_i)= e^{-K(ai|ui)}$:

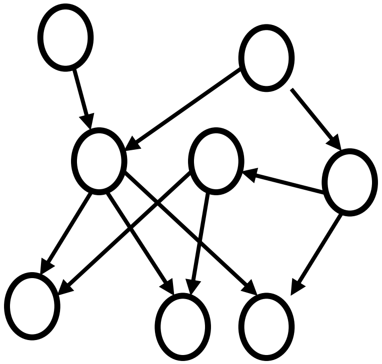


3- Les transformations

3-6



Réseau possibiliste *



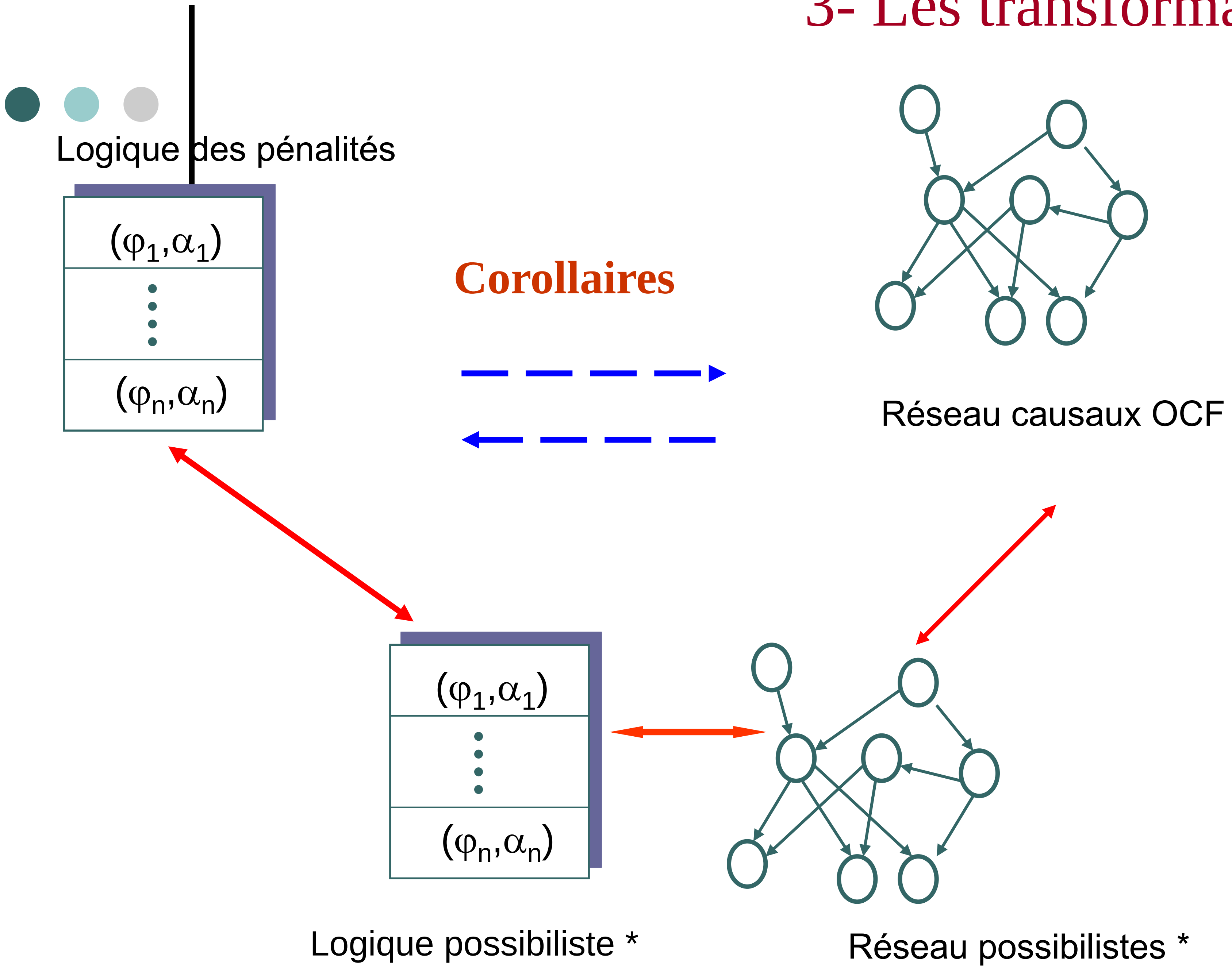
Réseau OCF

Utilisation d'échelles spécifiques.

En terme de distributions, il faudrait que:

$k(\omega) = -\log(\pi(\omega))$

3- Les transformations



4-1 Algorithmes d'inférence logique

Principe:

- Générer aléatoirement une base possibiliste quantitative en format WCNF(Weighted Conjonctive Normal Form) **Σ,**
- Transformer la base possibiliste **Σ** en une base à pénalités $\mathbf{P_K}$
- Générer aléatoirement une évidence (conjonction de littéraux) (**E,** ∞**)**
- Générer aléatoirement une variable d'intérêt A avec une instance **φ**

But:

$$\mathbf{P_K} \cup (E,\infty) \overset{?}{\models} (\varphi,\alpha)$$

Outils:

- Utilisation d'un prouver WMAXSAT (Weighted MAXSAT) qui fournit l'interprétation minimisant la somme des coûts des formules falsifiées.

4-1 Algorithme d'inférence en logique possibiliste quantitative

Algorithme PEN-WMAXSAT(PK,E,φ, α):

début

1- Genere(Σ,E, φ);

2- PK $\leftarrow$ Transform(Σ);

3- Wmaxsat(**PK** $\cup$ (E, ∞) $\cup$(φ, α),ω',k_{PK}(ω'));

4- Si ($k_{PK}(\omega') = +\infty$);

alors **PK** $\cup$ (E,+∞) $\not\models$ φ ;

sinon

PK $\cup$ (E,+∞) $\models$ (φ, $k_{PK}(\omega')$);

$k_{PK}(\varphi) = k_{PK}(\omega')$;

$\alpha = e^{-k_{PK}(\varphi)}$;

fsi

fin

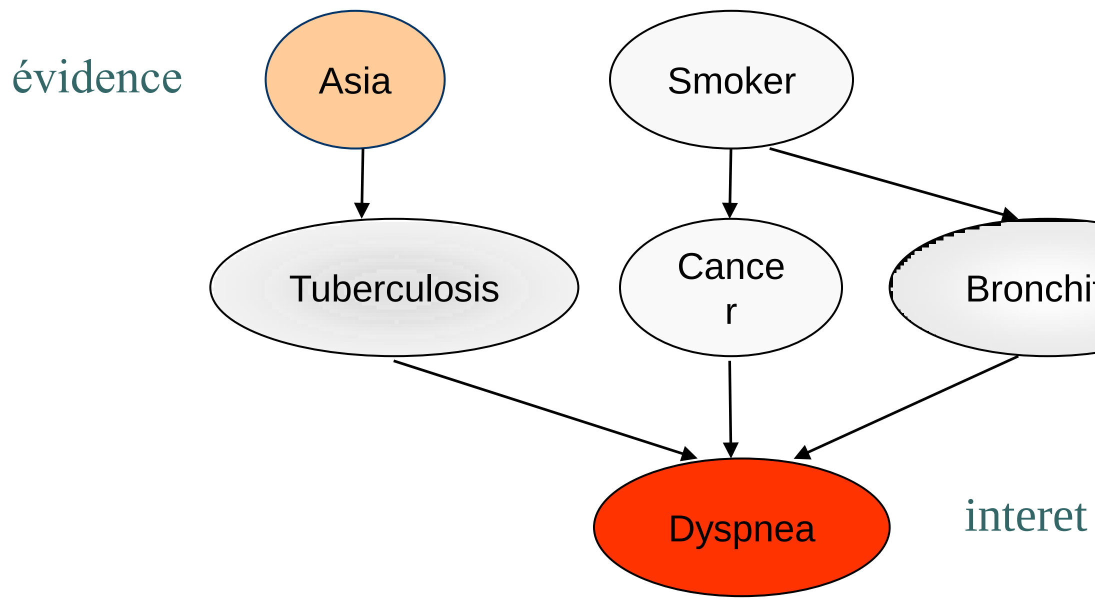
Algorithme de complexité Δ^p_2

4- Modes de Raisonnement

4-2 Les algorithmes de propagation en mode graphique

Principe:

1- Propagation de l'évidence sur l'ensemble des nœuds du réseau



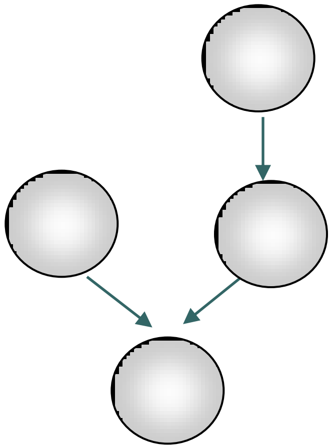
2– Calcul de Π(interet|evidence)

4- Modes de Raisonnement

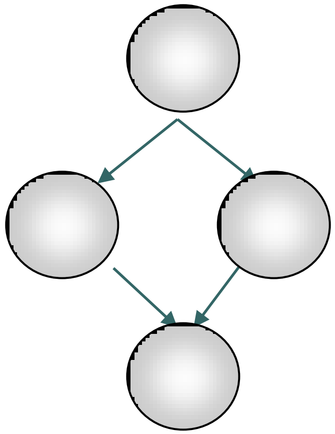
4-2 Les algorithmes de propagation en mode graphique

- ✓ Calculs locaux via les algorithmes de propagation

  -Méthodes exactes : Pearl (1988), Jensen (1996)



Simply connected DAG



Multiply connected DAG

Propagation dans un réseau causal simply connected

Cet algorithme est basé sur la communication locale via deux types:
- des messages λ circulant des nœuds enfants vers les nœuds parents
-des messages μ circulant des nœuds parents vers les nœuds enfants.

$$\forall a \in D_A : Bel(a) = \Pi(a|e) = \alpha.\lambda(a).\mu(a)$$

représente la mesure de possibilité conditionnelle de a à partir d'une évidence totale e,

où $\alpha = \dfrac{1}{\max_a Bel(a)}$

Propagation dans un réseau causal simply connected basé sur le produit:

Begin

Choosing the root of propagation

- Let S be the smallest connected set which contains the observed nodes ;
- Let pivot be an arbitrary node within S representing the root of propagation ;
- Let Postorder be the vector containing the order in which messages are sent in the distribute-evidence phase (the last node is the pivot) ;
- Let Preorder be the vector containing the order in which messages are sent in the collect-evidence phase (the first node is the pivot) ;

Initialization

- Set all λ and μ values and messages to 1 ;
- For each root A, $\mu(a) \leftarrow \prod(a)$, $\forall\ a \in D_A$;
- For each observed node A, set $\lambda_A(a)$ to 1 if A is instanciated to a and to 0 otherwise ;

Collect-evidence

For i $\leftarrow$ 1 to length (postorder)-1 **do**

A $\leftarrow$ postorder[i] ;   B $\leftarrow$ adjacent node of A in postorder ;

Compute $\lambda(A)$; Compute $\mu(A)$;

if B is a parent of A **then** post a λ message from A to B **else** post a μ message from A to B;

Distribute-evidence

For i $\leftarrow$ 1 to length (preorder) **do**

A $\leftarrow$ preorder[i] ;

Below $\leftarrow$ adjacent node of A in preorder ;

Compute $\lambda(A)$; Compute $\mu(A)$;

For j $\leftarrow$ 1 to length (Below) **do**

B $\leftarrow$ Below[j] ;

if B is a parent of A **then** post a λ message from A to B **else** post a μ message from A to B;

Marginalization

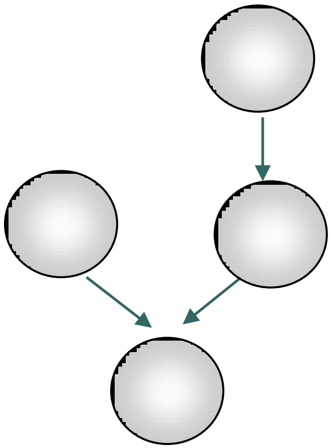
For each node A, compute $Bel(A) = \prod_p(a|e)$;

End

4- Modes de Raisonnement

4-2 Les algorithmes de propagation en mode graphique DAG Singly connected

Algorithme de propagation pour un polyarbre (basé sur le passage de messages)

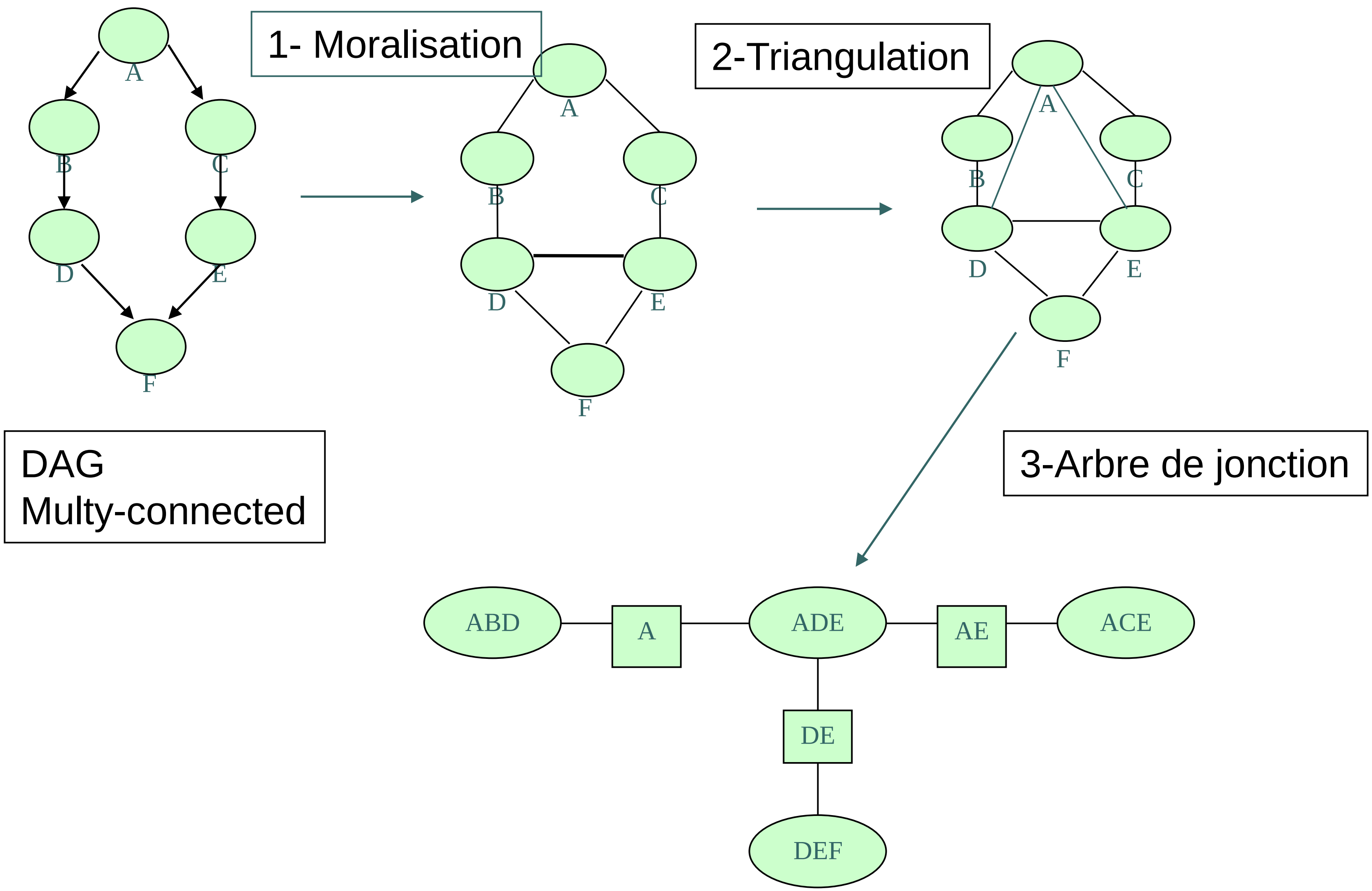


- Choisir de la racine de propagation
- Initialisation
- Collecte de l’évidence
- Distribution de l’évidence
- Marginalisation

Complexité Polynomiale

4- Modes de Raisonnement

4-2 Les algorithmes de propagation en mode graphique
DAG Multy connected



4- Modes de Raisonnement

4-2 Les algorithmes de propagation en mode graphique DAG Multy connected

Algorithme:

a- Initialisation de JT .

b- Propagation globale:

- ➢ Choisir de la racine de propagation
  - ➢ Collecte de l'évidence e
  - ➢ Distribution de l'évidence e

c- Marginalisation :

La prise en compte de l'évidence consiste à calculer Π (A|e) où A∈V et e est une évidence totale

Algorithme NP Complet

5- TP

•Exploitation de la Toolbox PNT(Possibilistic Networks Toolbox) de Benamor sous Matlab qui est une adaptation de la BNT (Bayesian Netaworks Toolbox) de Kevin Murphy incluant une procédure de génération automatique de graphe causal possibiliste basé sur le produit.

• Utilisation de la Toolbar incluant un prouver WMAXSAT de Scheix et al élue comme meilleur prouver MAXSAT 2006.

•Utilisation de l'environnement CYGWIN (Emulateur Linux sous Windows).

•Exploitation des procédures de passage avec C sous CYGWIN:

- •D'un graphe possibiliste * vers une base possibiliste *.
- •D'une base possibiliste * vers une base de pénalités.

• Exploitation de l'algorithme d'inférence pour la logique des pénalités basé sur WMAXSAT (Weighted MAXSAT).

6-Conclusions…

Mode logique

+Puissance du mode de représentation des connaissances

+ Puissance du mécanisme d'inférence

- Complexité:

Δ_2^p (WMAXSAT)

Mode graphique

+ Expressivité des connaissances

+Puissance des algorithmes de propagation

+ Existence des algorithmes d'apprentissage

- Complexité:

Polynomial (polyarbre)

NP-complet (Multy-connected

- Limites de l'algorithme de JT pour des graphes très denses