Concepts de plus proches voisins dans des graphes de connaissances

Sébastien Ferré
SemLIS, IRISA/Université Rennes 1, France

PFIA/IC 2017, Caen

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

UMR IRISA LIS UNIVERSITÉ DE RENNES 1

Motivation

- La méthode des *k* plus proches voisins (k-NN)
  - est une forme d'apprentissage paresseux
  - qui permet la classification supervisée
  - et est une composante importante du raisonnement à partir de cas

Question

Comment l'adapter à des graphes de connaissances :

- où les objets sont les nœuds du graphe
- où la description de chaque objet est le graphe centré sur cet objet
- sans se ramener à des descriptions à plat

Motivation

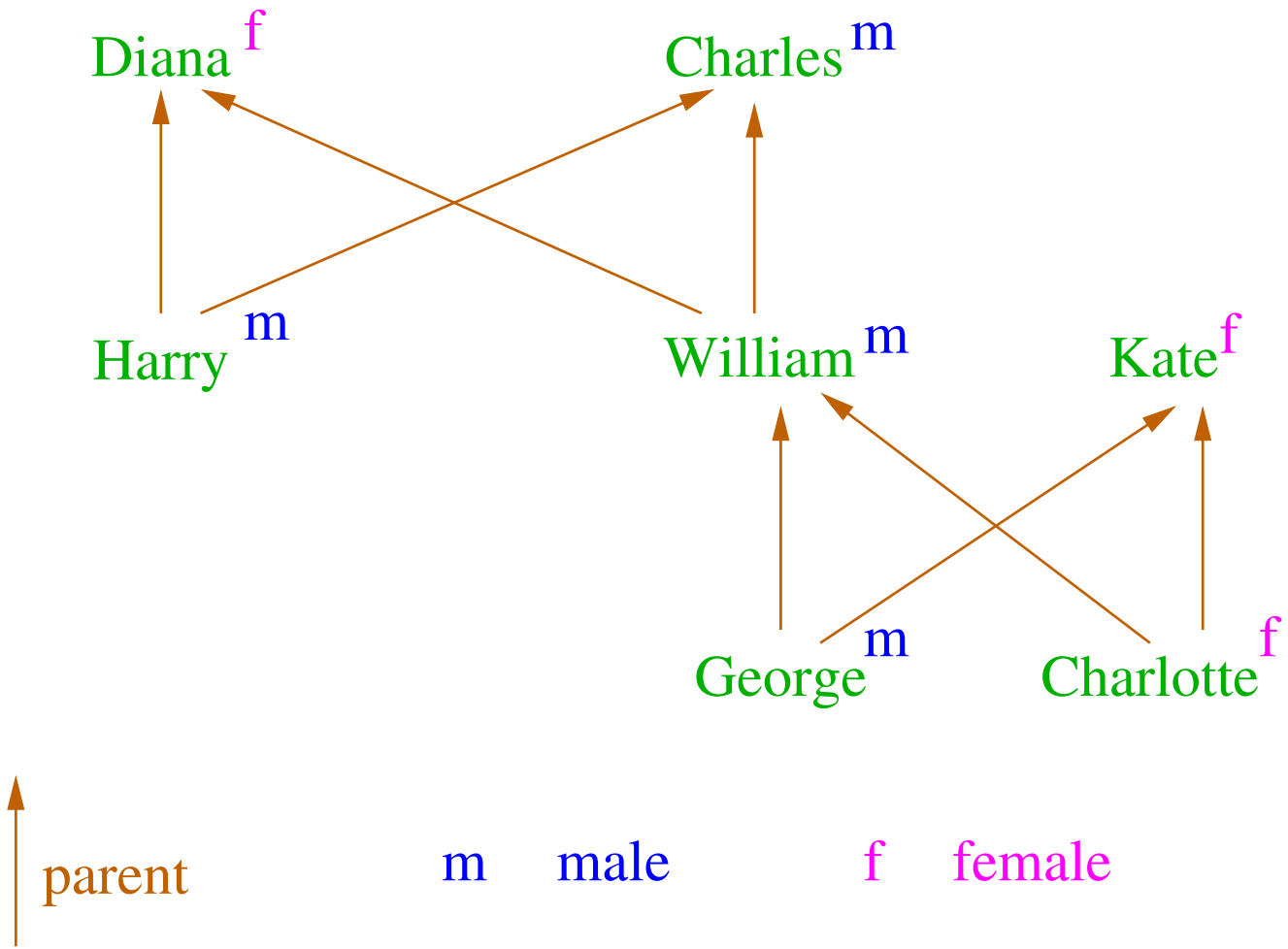
- La méthode des *k* plus proches voisins (k-NN)
  - est une forme d'apprentissage paresseux
  - qui permet la classification supervisée
  - et est une composante importante du raisonnement à partir de cas

Question

Comment l'adapter à des graphes de connaissances :

- où les objets sont les nœuds du graphe
- où la description de chaque objet est le graphe centré sur cet objet
- sans se ramener à des descriptions à plat

Un exemple de graphe de connaissances



Travaux existants

- L'approche k-NN suppose une mesure de distance (ou de similarité)
- On distingue deux types de distance/similarité [Bisson'00]
  - numérique: définissant un ordre total
    - PRO souple et économique
    - CONS une même valeur peut cacher des similarités très différentes
    - CONS rarement défini sur des descriptions complexes
      ex: RIBL (Relational Instance-Based Learning) [Horvath'01]
  - symbolique: définissant un ordre partiel (ex., subsomption)
    - ★ par calcul de plus petites généralisations communes (*lgg*)
      ex: découverte de concepts (FCA), de règles de classif. (PLI)
    - PRO plus précis et offrant une explication
    - CONS plus couûteux à calculer
    - CONS rarement appliqué pour k-NN
      ex: classification à base de FCA [Kuznetsov'13]

Choix du formalisme

Besoins

1. un cadre de type FCA définissant l'espace des similarités symboliques (concepts) à partir d'un ensemble d'objets et leurs descriptions
2. les objets (instances) sont en relation les uns avec les autres formant un graphe de connaissances
3. la similarité entre objets prend la forme de patterns de graphes
   - sans limite a priori de profondeur de traversée
   - avec possibilité de cycles

Options

- RCA: Relational Concept Analysis [Rouane et al'07]
- G-FCA: Graph-FCA [Ferré'15]

Choix du formalisme

Besoins

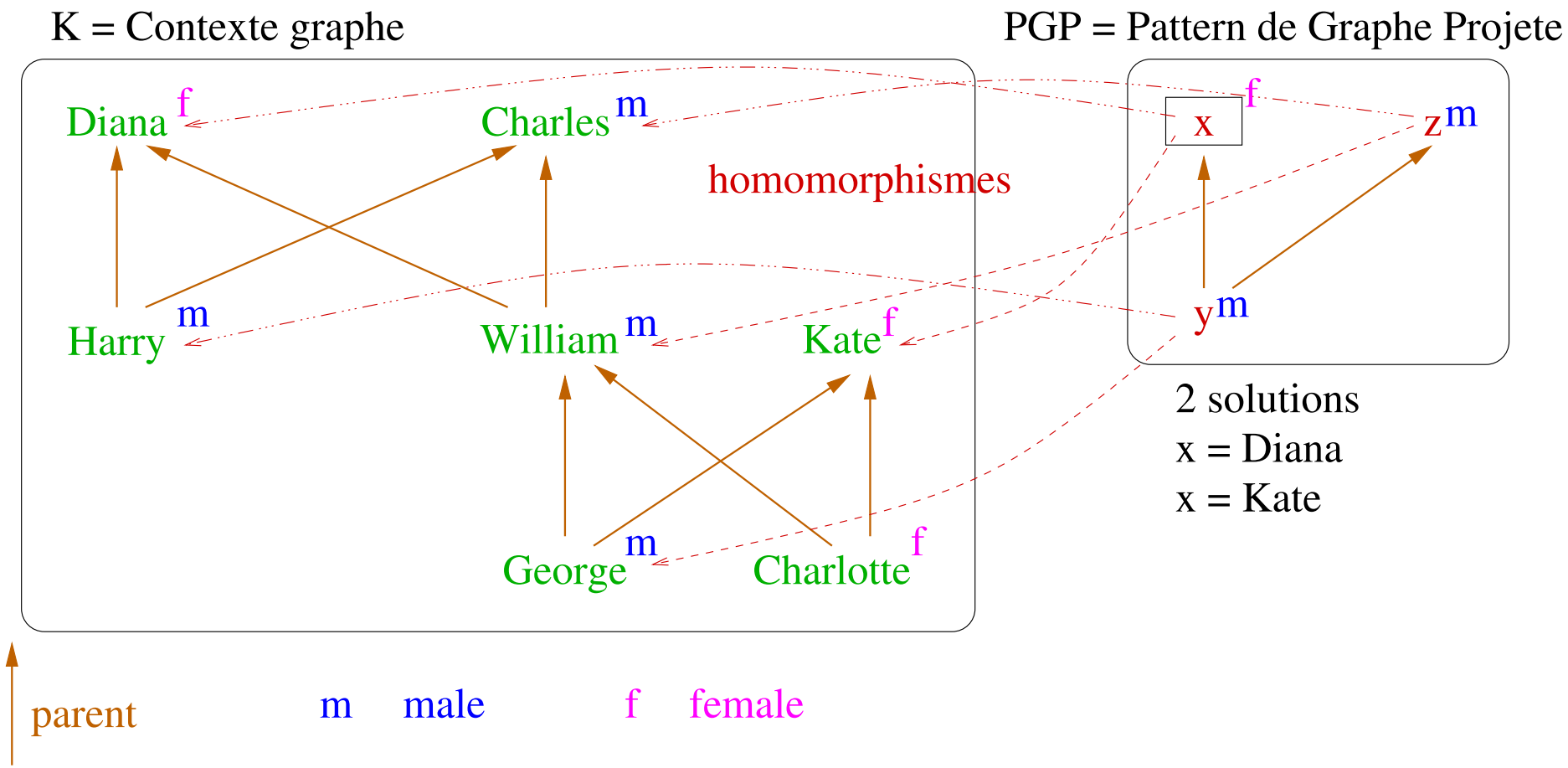
1. un cadre de type FCA définissant l'espace des similarités symboliques (concepts) à partir d'un ensemble d'objets et leurs descriptions
2. les objets (instances) sont en relation les uns avec les autres formant un graphe de connaissances
3. la similarité entre objets prend la forme de patterns de graphes
   - sans limite a priori de profondeur de traversée
   - avec possibilité de cycles

Choix

Nous choisissons ici Graph-FCA car ses patterns de graphes

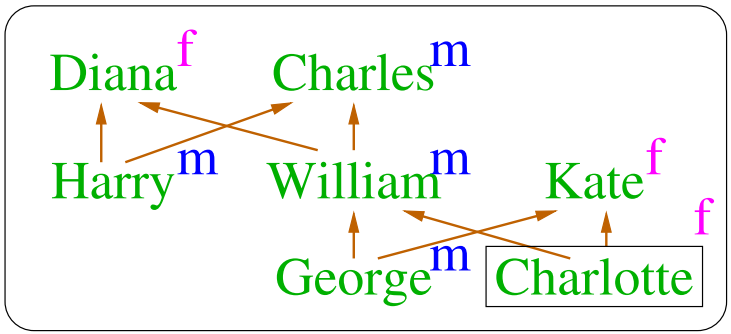
1. sont plus faciles à lire
2. peuvent exprimer des cycles entre objets
3. peuvent utiliser des relations n-aires
4. sont définis de façon directe plutôt qu'itérative

Graph-FCA en bref (1/3)



Graph-FCA en bref (2/3)

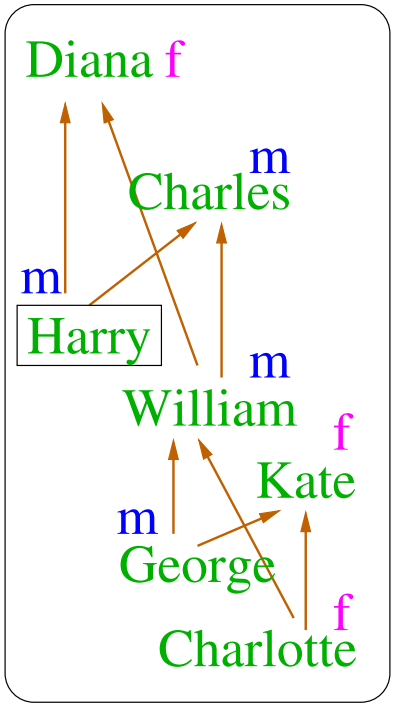
Q(Charlotte)
inter
Q(Harry)



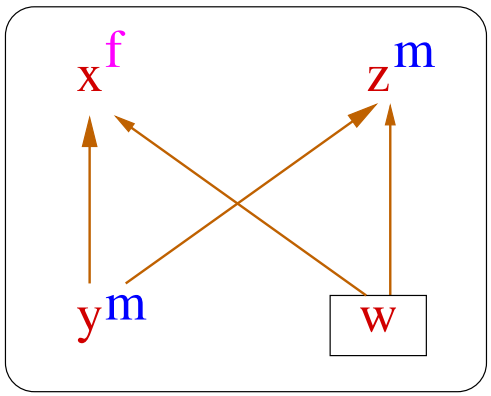
parent

m male

f female



concept graphe



intension

(4 solutions)
Charlotte, Harry,
George, William

extension

Graph-FCA en bref (1/3)

Quelques notations utiles:

- K: contexte graphe
- $Q = (x, P)$: pattern de graphe P projeté en x (PGP)
  - $Q(o) = (o, K)$: PGP décrivant o dans K
  - $Q_1 \subseteq_q Q_2$: inclusion par homomorphisme
  - $Q_1 \cap_q Q_2$: intersection de 2 PGP, plus petite généralisation commune
  - $ext(Q) = \{o \mid Q \subseteq_q Q(o)\}$: extension d'un PGP
- $C = (Q, R)$: concept (intension,extension)
  tel que $C.int = Q = \bigcap_{o \in R} Q(o)$ et $C.ext = R = ext(Q)$
  - partial order: $C_1 \leq C_2$ ssi $C_1.ext \subseteq C_2.ext$ ssi $C_2.int \subseteq_q C_1.int$
  - supremum: $(C_1 \vee C_2).int = C_1.int \cap_q C_2.int$

Distance conceptuelle

Nous définissons une distance symbolique entre objets d'un graphe.

Definition

La distance conceptuelle entre deux objets u et v d'un contexte graphe K est définie comme le concept $\delta(u, v) = (Q, R)$ où :

- $\delta(u, v).int = Q = Q(u) \cap_q Q(v)$
  la description que u et v ont en commum
- $\delta(u, v).ext = R = ext(Q)$
  les objets qui partagent cette description commune

Deux distances/similarité numériques peuvent être dérivées :

1. $dist(u, v) := |\delta(u, v).ext|$: nombre d'objets séparant u et v
2. $sim(u, v) := |\delta(u, v).int|$: nombre d'arcs en commun

Les ordres totaux sur ces mesures sont tout deux compatibles avec l'ordre partiel sur les distances conceptuelles.

$$\delta(u, v) \leq \delta(u, w) \Rightarrow dist(u, v) \leq dist(u, w) \wedge sim(u, v) \geq sim(u, w)$$

Distance conceptuelle

Nous définissons une distance symbolique entre objets d'un graphe.

Definition

La distance conceptuelle entre deux objets u et v d'un contexte graphe K est définie comme le concept $\delta(u, v) = (Q, R)$ où :

- $\delta(u, v).int = Q = Q(u) \cap_q Q(v)$
  la description que u et v ont en commum
- $\delta(u, v).ext = R = ext(Q)$
  les objets qui partagent cette description commune

Deux distances/similarité numériques peuvent être dérivées :

1. $dist(u, v) := |\delta(u, v).ext|$: nombre d'objets séparant u et v
2. $sim(u, v) := |\delta(u, v).int|$: nombre d'arcs en commun

Les ordres totaux sur ces mesures sont tout deux compatibles avec l'ordre partiel sur les distances conceptuelles.

$$\delta(u, v) \leq \delta(u, w) \Rightarrow dist(u, v) \leq dist(u, w) \wedge sim(u, v) \geq sim(u, w)$$

Concepts de (plus proches) voisins

Définition

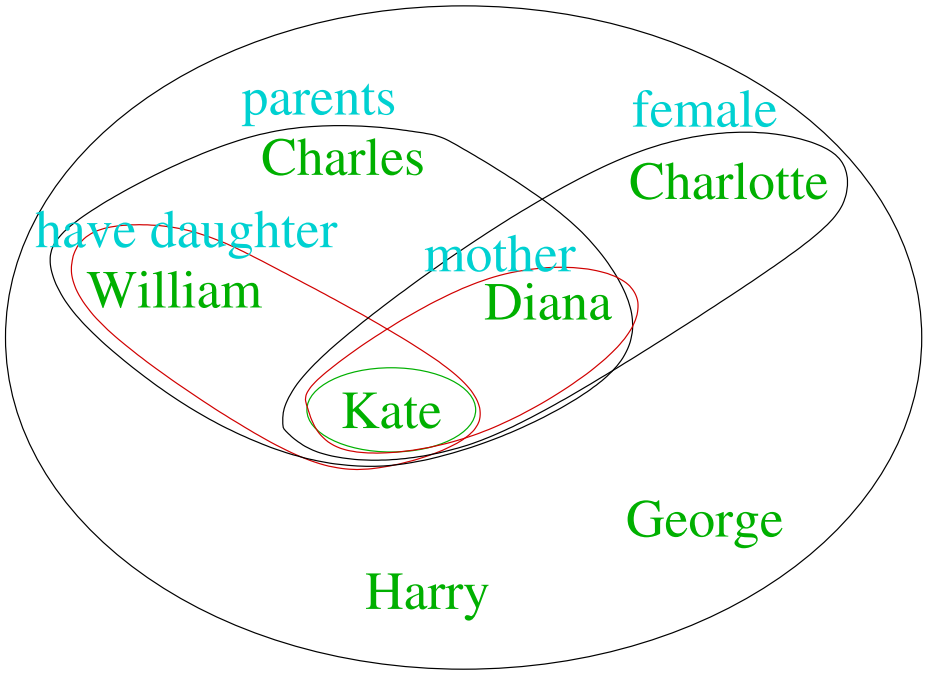
Les concepts de voisins d'un objet u sont l'ensemble des distances conceptuelles partant de u.

$$CN(u) := \{\delta(u, v) \mid v \in K\}$$

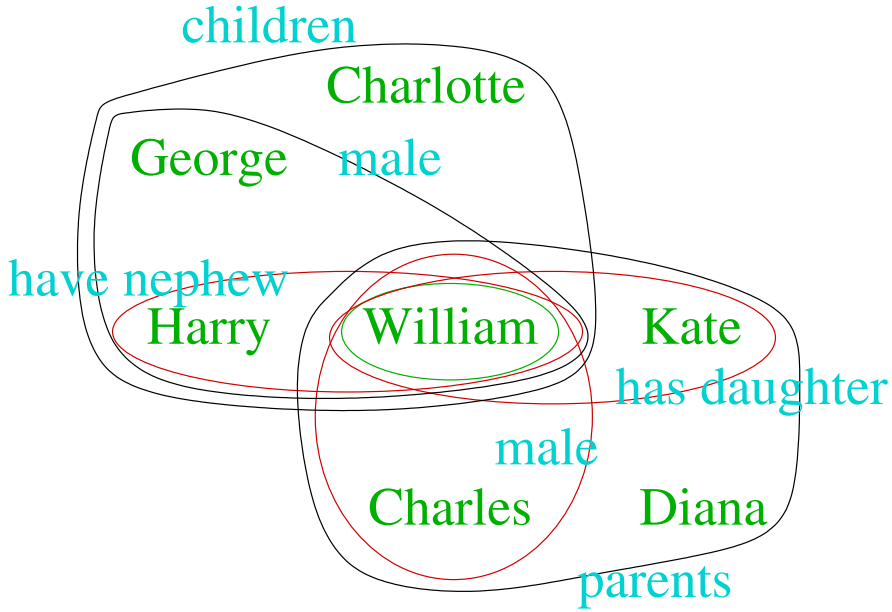
- $\delta.proper = \{v \in \delta.ext \mid \delta(u, v) = \delta\}$
  l'extension propre d'un concept de voisin est le sous-ensemble de ses instances qui sont exactement à cette distance
- $CNN(u) = \{\delta \in CN(u) \mid \delta(u, u) \prec \delta\}$
  les concept de plus proches voisins sont ceux qui sont immédiatement supérieurs à la distance conceptuelle "zéro" $\delta(u, u)$

Examples de concepts de voisins

CN(Kate)
Voisins de Kate



CN(William)
Voisins de William



Algorithme naïf

Principe: itération sur l'ensemble des objets

1. $CN(u) \leftarrow \emptyset$
2. pour tout $v \in K$
   1. $Q \leftarrow Q(u) \cap_q Q(v)$ // calcul intension
   2. $R \leftarrow \{o \in K \mid Q \subseteq_q Q(o)\}$ // calcul extension
   3. $CN(u) \leftarrow CN(u) \cup \{\delta\}$ where $\delta = (R, Q)$
   4. $\delta.proper \leftarrow \delta.proper \cup \{v\}$ // v est dans l'extension propre de δ

Problèmes

- les opérations $\cap_q$ et $\subseteq_q$ sont très coûteuses
  $\Rightarrow$ *homomorphismes de graphes*
- le nombre d'objets peut être beaucoup plus grand que le nombre de concepts de voisins (redondance)
- donc d'un objet à l'autre, beaucoup de calculs sont répétés

Solutions trouvées

Nous avons trouvé deux techniques algorithmiques complémentaires:

1. Partitionnement de l'ensemble d'objets
   - plutôt qu'itération sur les objets
   - chaque arc dans la description de u est susceptible de couper une partie (un pré-concept de voisins) en deux
   - avantages
     - ★ le nombre de parties est borné par le nombre d'objets *alors que l'espace des patterns est exponentiel*
     - ★ le processus est très flexible ⇒ heuristiques
     - ★ le processus est *anytime* ⇒ *timeout* plutôt que paramètres
2. Jointures paresseuses de relations
   - représentation compacte d'un ensemble d'homomorphismes pour le calcul des extensions de concepts de voisins
   - avantages
     - ★ calcul incrémental des extensions, au gré des partitionnements
     - ★ complexité mémoire en $O(Nm^2)$ au lieu de $O(Nm^m)$ pour N films de m acteurs, soit un facteur de 10^8 pour 10 acteurs !

Solutions trouvées

Nous avons trouvé deux techniques algorithmiques complémentaires:

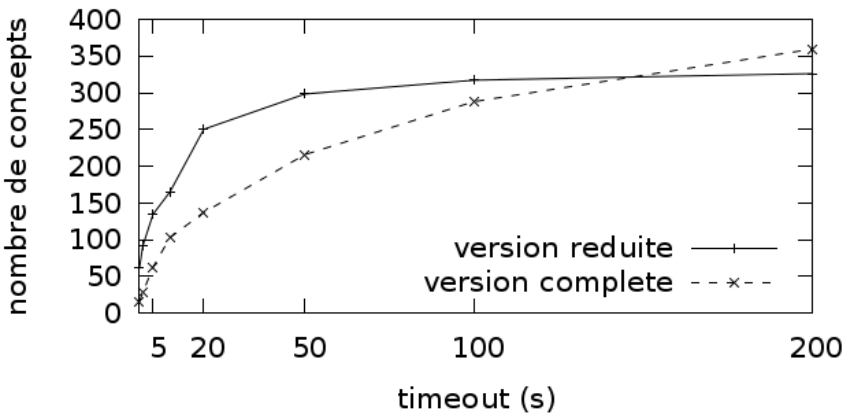
1. Partitionnement de l'ensemble d'objets
   - plutôt qu'itération sur les objets
   - chaque arc dans la description de u est susceptible de couper une partie (un pré-concept de voisins) en deux
   - avantages
     - ★ le nombre de parties est borné par le nombre d'objets *alors que l'espace des patterns est exponentiel*
     - ★ le processus est très flexible ⇒ heuristiques
     - ★ le processus est *anytime* ⇒ *timeout* plutôt que paramètres
2. Jointures paresseuses de relations
   - représentation compacte d'un ensemble d'homomorphismes pour le calcul des extensions de concepts de voisins
   - avantages
     - ★ calcul incrémental des extensions, au gré des partitionnements
     - ★ complexité mémoire en $O(Nm^2)$ au lieu de $O(Nm^m)$ pour N films de m acteurs, soit un facteur de 10^8 pour 10 acteurs !

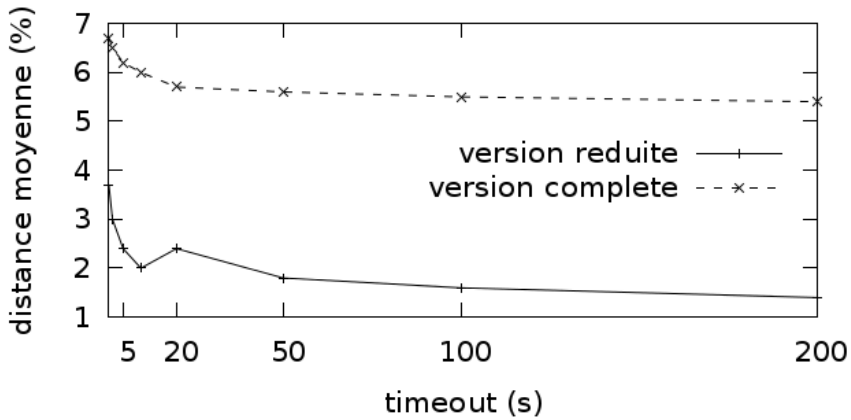
Implémentation

- Implémentation: dans SEWELIS
  - un système d'exploration et d'édition de graphes RDF
  - en environ 700 lignes de code OCaml
- Application: guidage lors de la saisie de données RDF
  - contexte graphe: graphe RDF + description RDF en cours
  - objet central u: nœud RDF à compléter (focus utilisateur)
  - usage des objets voisins: saisie prédictive
- Heuristiques:
  - stratégie partitionnement: parcours en largeur
  - choix des arcs: compromis entre proximité de u et diversité

Expérimentation: données et résultats

- Données: base géographique MONDIAL
  - pays, populations, langues, villes, rivières, régimes politiques
  - 2 versions
    1. complète: 41577 nœuds et 120546 arcs
    2. réduite: 9692 nœuds et 11691 arcs (- données num.)
- Validation de l'algorithme *anytime* sur ces grands graphes





Expérimentation: examples de voisinages

Quelques plus proches voisins (avec explications)

- France
  - Italie: république, en Europe, parle français
  - Nouvelle-Zélande: a une dépendance, parle français
  - Pologne: république, en Europe, voisin Allemagne
  - Espagne: en Europe, sur l'Atlantique, a une dépendance
  - ...
- Pologne
  - Roumanie: république, en Europe, ethnie allemande
  - Lituanie: en Europe, parle allemand
  - Allemagne: en Europe, ethnie allemande, sur l'Oder, voisin Tchéquie
- Chine
  - Corée du Nord: en Asie, voisin Russie, sur Mer Jaune
  - Norvège: voisin Russie, a une dépendance
  - Vietnam: en Asie, état communiste, a des musulmans
  - Géorgie: en Asie, voisin Russie, a des musulmans
  - ...

Conclusion

- distance symbolique à base de concepts de Graph-FCA
- sur des graphes de connaissances avec relations n-aires
  RDF, graphes conceptuels
- algorithme *anytime* de génération de concepts de voisins
  - sans paramètre autre qu'un *timeout* (ex., profondeur, taille patterns)
  - donnant de bons résultats en quelques secondes sur +100.000 arcs
- application à la saisie prédictive de données RDF
  - prédiction de valeurs, classes et propriétés sur la base des plus proches voisins

Travaux futurs

- l'approche se généralise à des tuples d'objets
  - ex: quels sont les plus proches voisins de (France,Paris) où (Einstein,relativité, 1905) ?
  - forme de recherche d'analogies
- amélioration des algorithmes
  - il reste des marges d'optimisation
  - traitement des valeurs numériques (discrétisation localisée)
- application à la classification automatique
  - et comparaison avec approches RIBL et PLI
  - avantage sur PLI: flux d'instances, nombreuses classes

Fin

Merci de votre attention!

Graph FCA (G-FCA)

Key ideas

G-FCA is an extension of FCA satisfying:

1. direct application to Knowledge Graphs (KG)
2. expressive KGs: n-ary predicates, directed/multiple edges
3. expressive concepts: mixed arities, n-ary concepts
4. self-contained representations of extents/intents
5. natural extension of FCA, FCA as a special case

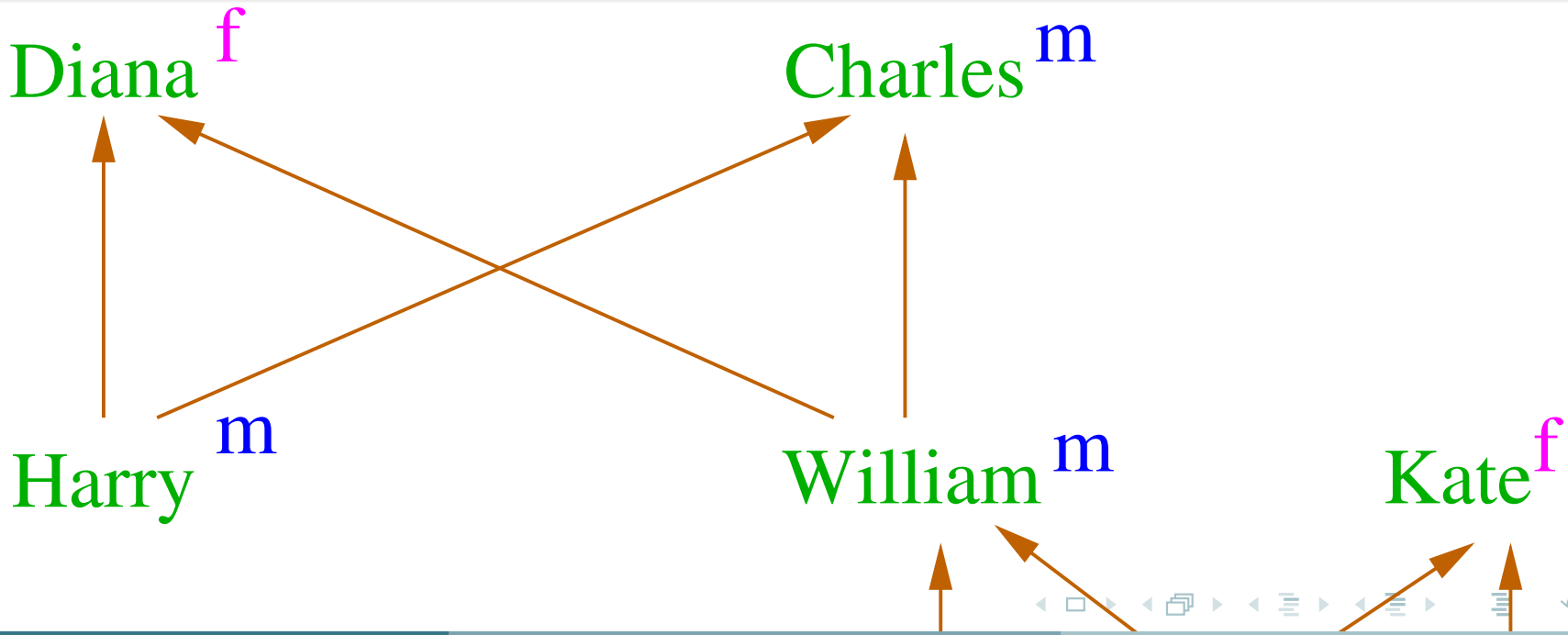
Hint 1: object $\rightsquigarrow$ tuple of objects
Hint 2: object $\rightsquigarrow$ variable

G-FCA Formal Context

Definition

A G-FCA formal context, or graph context, is a triple $K = (O, A, I)$:

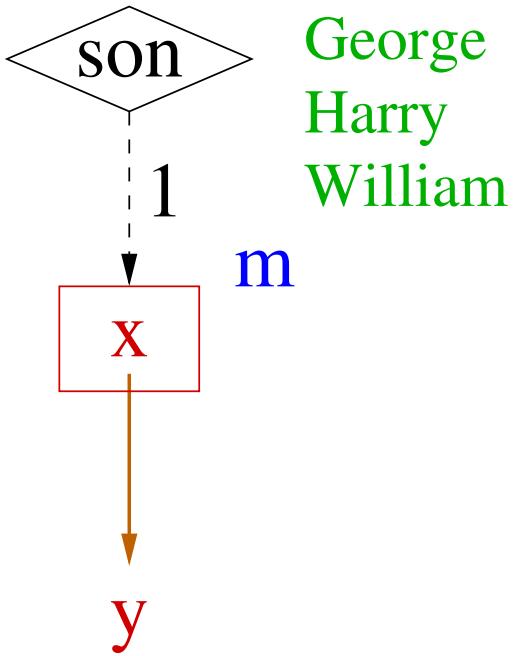
- O is a set of objects (*nodes*)
- A is a set of attributes (*edge labels*)
- $I \subseteq O^* \times A$ is a set of relationships (*edges*) with O^* tuples of objects

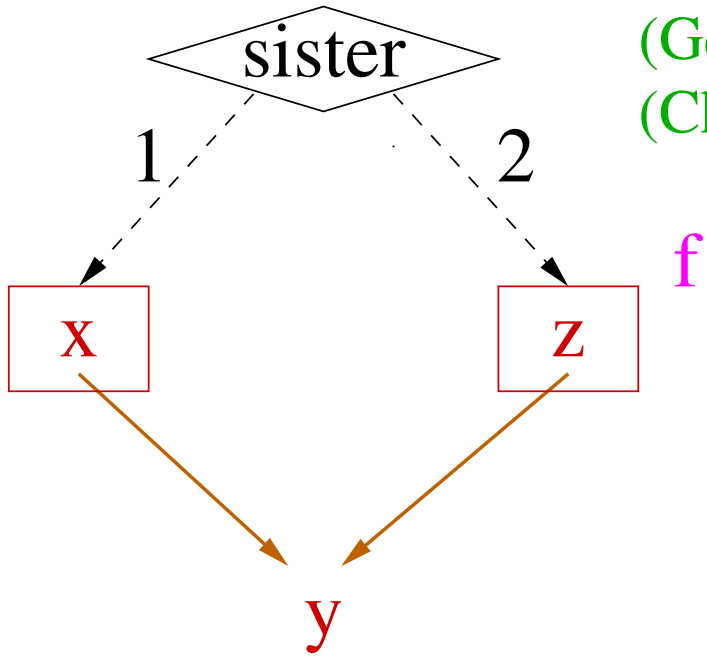


Projected Graph Patterns (PGP)

Definition

A *projected graph pattern* (PGP) is a couple $Q = (\overline{x}, P)$ where $P \subseteq \mathcal{V}^* \times A$ is a graph pattern, and $\overline{x} \in \mathcal{V}^*$, called *projection tuple*, is a tuple of variables.

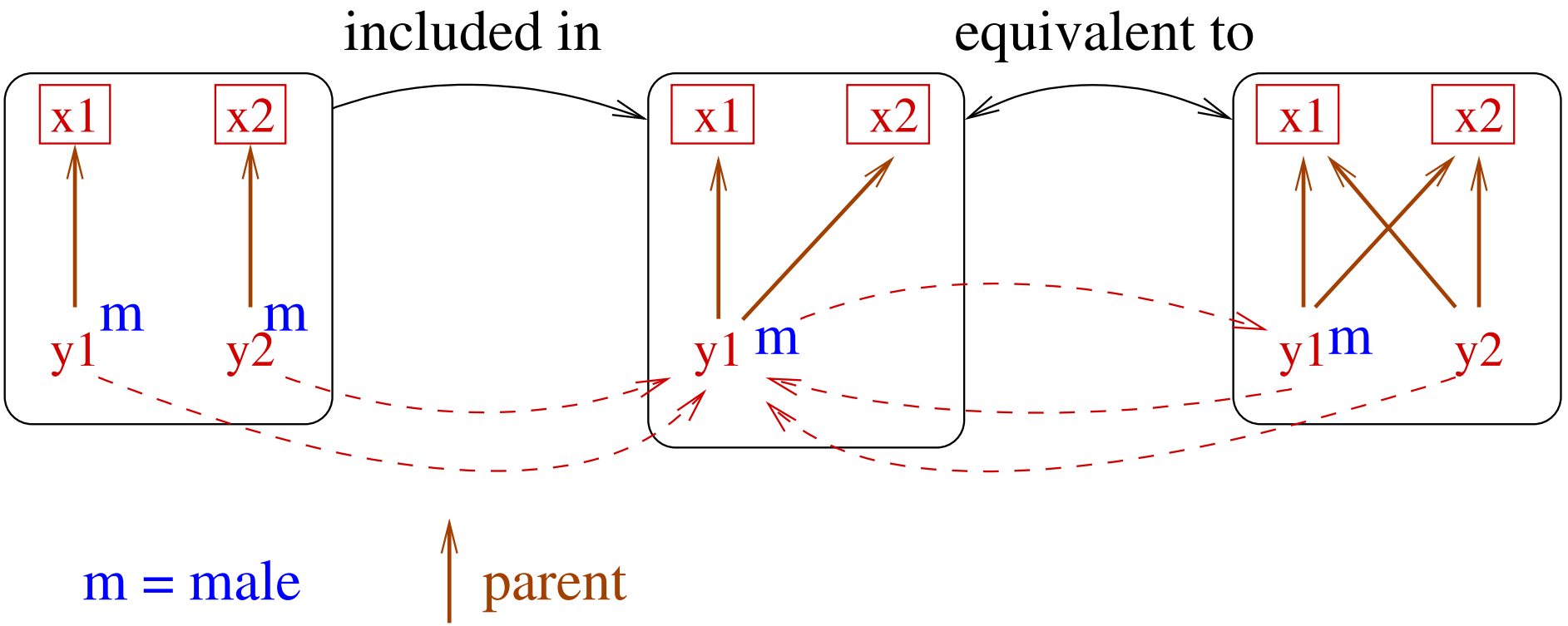




PGP Inclusion $\subseteq_g$

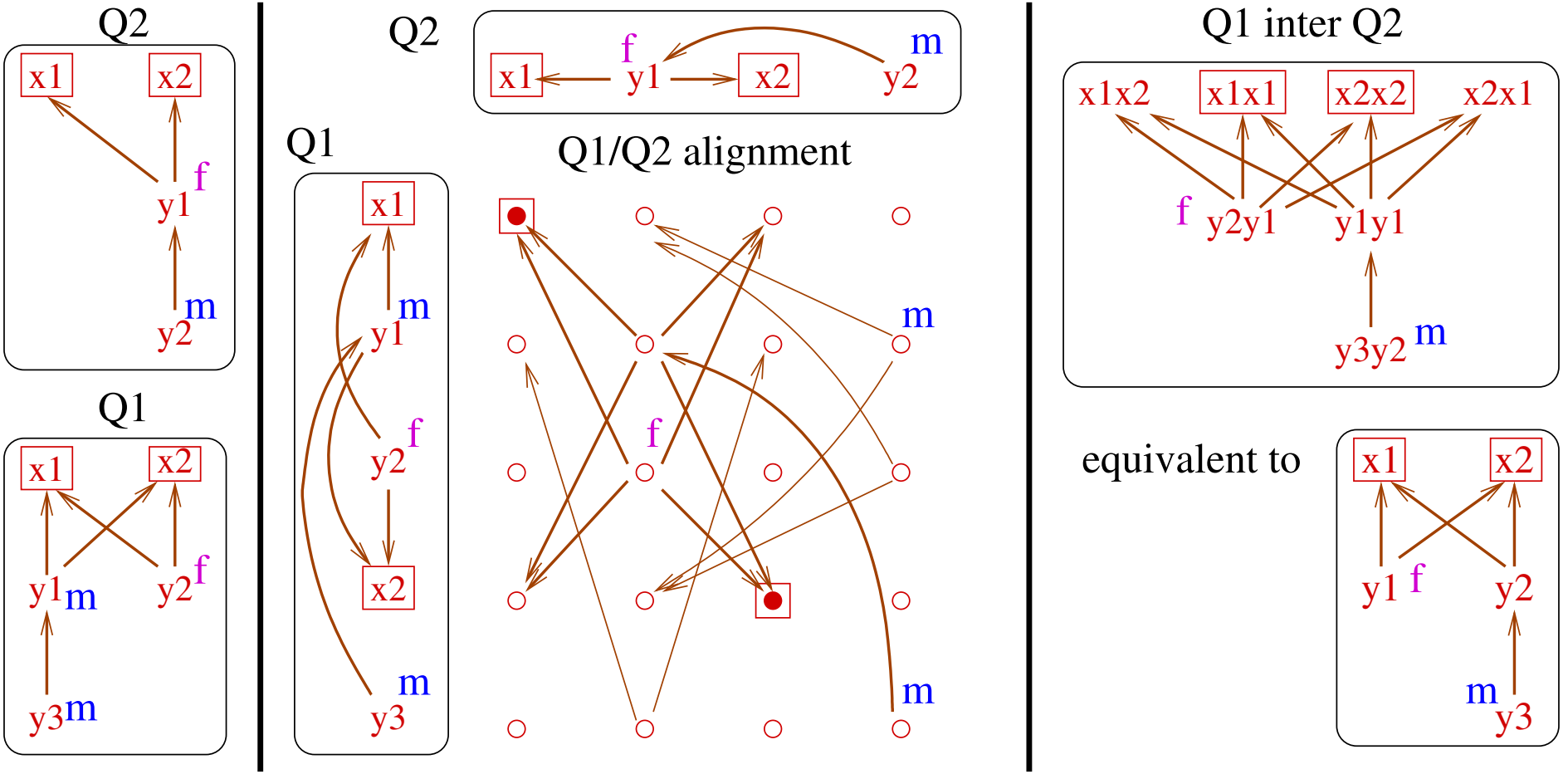
Definition

Q_1 is included in Q_2 iff there is a homomorphism ϕ from P_1 to P_2 such that $\phi(\overline{x}_1) = \overline{x}_2$.



PGP Intersection $\cap_q$

PGP intersection can be defined and computed similarly to a sequence alignment.



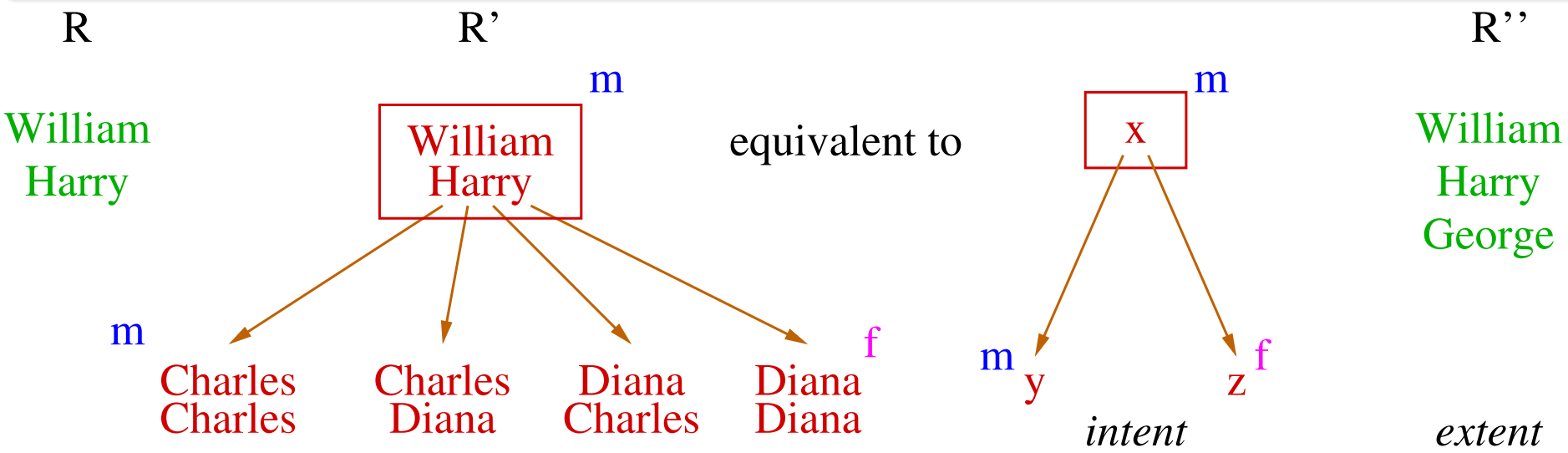
G-FCA Galois Connection

Theorem

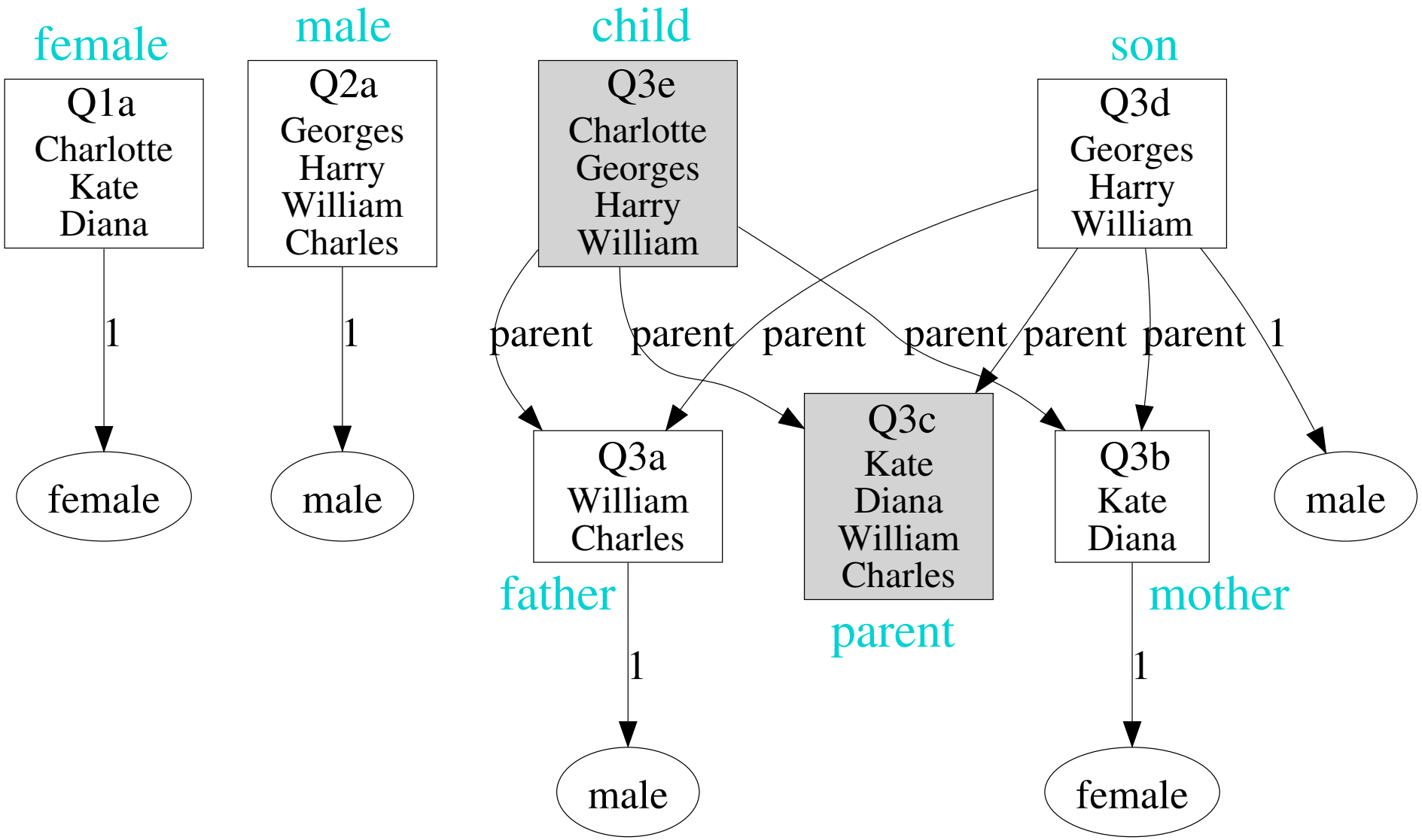
For every graph context $K = (O, A, I)$, and arity $k \geq 0$, the pair of mappings (ext, int), defined by

1. $ext(Q) = Q' = \{\overline{o} \in O^k \mid Q \subseteq_q (\overline{o}, I)\}$, for $Q \in \mathcal{Q}_k$
2. $int(R) = R' = \bigcap_q \{(\overline{o}, I)\}_{\overline{o} \in R}$, for $R \in \mathcal{R}_k$

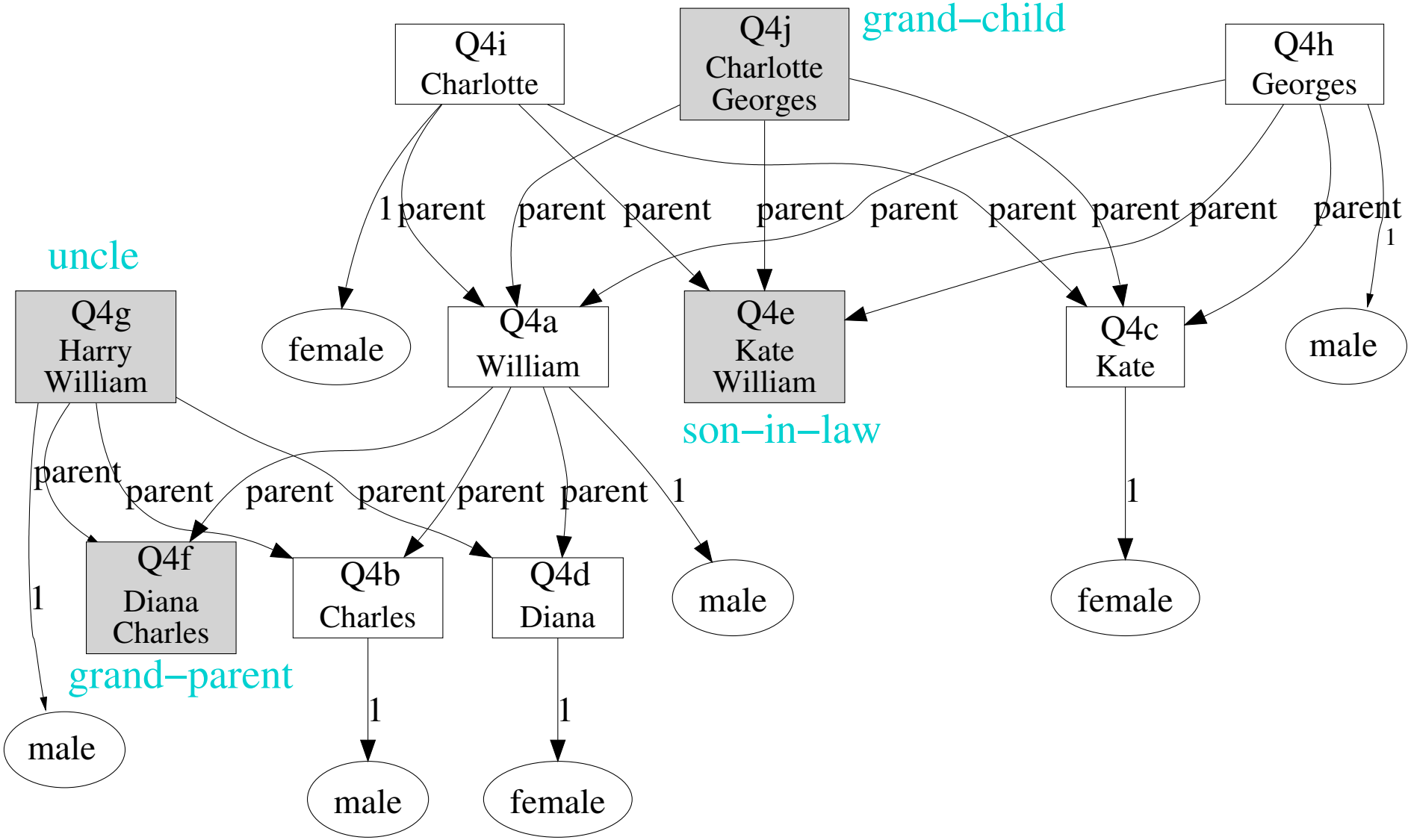
forms a Galois connection.



All Concepts about the British Royal Family (1/2)



All Concepts about the British Royal Family (2/2)



Lattice of Unary Concepts about the British Royal Family