Vues logique / graphe des faits et des requêtes conjonctives

HAI933I

RAPPELS : BASES DE FAITS ET REQUÊTES CONJONCTIVES

- Une **base de faits F** est généralement vue comme un ensemble d'atomes instanciés (« ground »)
- D'un point de vue logique, **F** est vue comme la conjonction de ces atomes
- On peut étendre la notion de base de faits pour prendre en compte des valeurs inconnues : variables quantifiées existentiellement
  => une base de faits est alors vue comme une **conjonction d'atomes dont toutes les variables sont quantifiées existentiellement**

BD relationnelle

Movie

m_id	
m1	…
m2	…
?x	

Actor

a_id	
a	…
b	…
c	

Play

m_id	a_id
a	m1
a	m2
c	?x

Base de faits

{ movie(m1), movie(m2), movie(x), actor(a), actor(b), actor(c), play(a,m1), play(a,m2), play(c,x) }

Formule logique associée à une base de faits

$$\exists x \ (\ movie(m1) \wedge movie(m2) \wedge movie(x) \quad actor(a) \wedge actor(b) \wedge actor(c) \quad play(a,m1) \wedge play(a,m2) \wedge play(c,x)\ )$$

Rappels : bases de faits et requêtes conjonctives

Une **requête conjonctive (CQ)** $Q(x_1 \dots x_k)$ est de la forme

$$\exists x_{k+1},\dots,x_m \; (A_1 \wedge \dots \wedge A_p)$$

où $A_1,\dots,A_p$ sont des atomes ayant pour variables $x_1,\dots,x_m$

Autrement dit, une requête conjonctive est une **conjonction d'atomes quantifiée existentiellement** (mais pas forcément close)

Notation simplifiée $\quad Q(x_1 \dots x_k) = \{ A_1, \dots, A_p \}$

$q(x) = \exists y \; (\text{movie}(y) \wedge \text{play}(x, y))$

$q(x) = \{ \text{movie}(y), \text{play}(x,y) \}$

Fragment existentiel conjonctif positif : FOL(∃,∧)

Formules construites avec le quantificateur existentiel (∃) et la conjonction (∧)

- Forme normalisée (« prénexe ») :

  $\exists x_1 \ldots \exists x_n (A_1 \wedge \ldots A_p)$ où les A_i sont des atomes
  et chaque x_j apparait dans un A_i
- Permettent de représenter des bases de faits (et bases de données relationnelles) et des requêtes conjonctives
- Pour des formules closes (bases de faits et CQs booléennes) :

  $f1 \models f2$ ssi il existe un homomorphisme de $f2$ dans $f1$

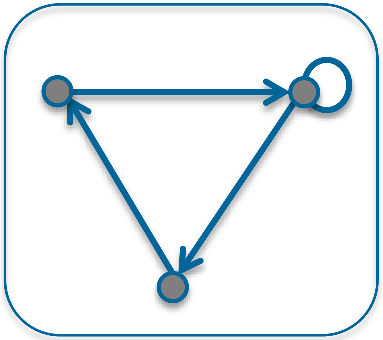
Dans ce cours :

- Vision « graphe » de ces formules (=> homomorphisme de graphe)
- Notion de minimalité : peut-on supprimer des atomes en gardant une formule équivalente ? Deux formules équivalentes sont-elles « identiques » ?

- On note G = (V,E) un graphe orienté où V est l'ensemble des sommets (*vertices*)
  et E est l'ensemble des arcs (*edges*)

- Un ensemble d'atomes avec un seul prédicat binaire et sans constantes peut être vu comme un graphe orienté

{ p(x,y), p(y,z), p(z,x), p(y,y) }



(et réciproquement pour les graphes sans sommets isolés – ou alors on introduit un prédicat unaire pour typer les sommets)

Etant donné un tel ensemble d'atomes A, on construit un graphe (V,E) tel que :

- V est en bijection avec les termes de A (des variables ici) (soit b cette bijection)

- E est en bijection avec les atomes de A :

  E contient un arc (b(x),b(y)) si et seulement si p(x,y) ∈ A

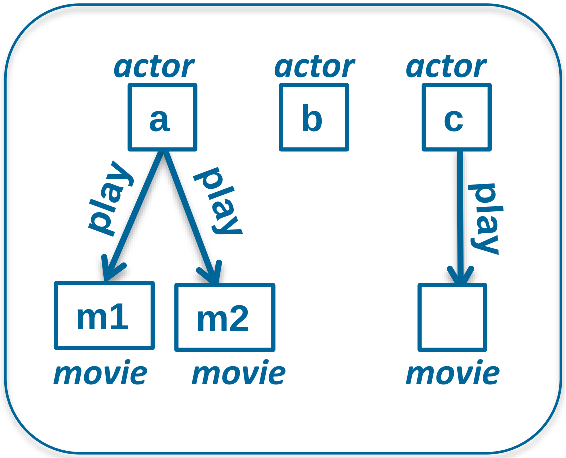
- On étiquette :

  les arcs si on a plusieurs prédicats binaires

  les sommets si on a des constantes

- On peut ajouter un 2ème type d'étiquette sur les sommets pour représenter les prédicats unaires

movie(m1), movie(m2), movie(x),
actor(a), actor(b), actor(c),
play(a,m1), play(a,m2), play(c,x)



Et si on a des prédicats
d'arité supérieure à 2 ?

Ensemble d'atomes encodé par un graphe

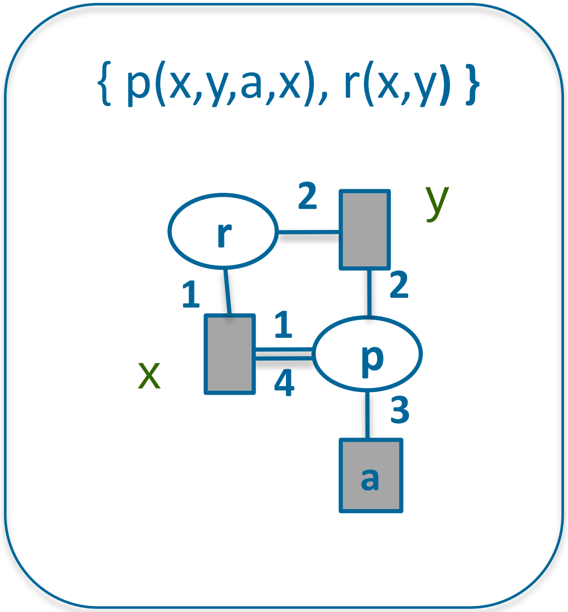
- À un ensemble d'atomes, on associe naturellement un hypergraphe « orienté ».

  La notion d'hyperarc généralise la notion d'arc : n-uplet (n > 0) de sommets

- Il est pratique de considérer le graphe d'incidence associé à l'hypergraphe.
  C'est un multi-graphe biparti non-orienté

  « biparti » : l'ensemble des sommets est partitionné en 2 classes, tel qu'il n'y a aucun arc entre 2 sommets de la même classe

  « multi-graphe » : il peut y avoir plusieurs arêtes entre deux sommets



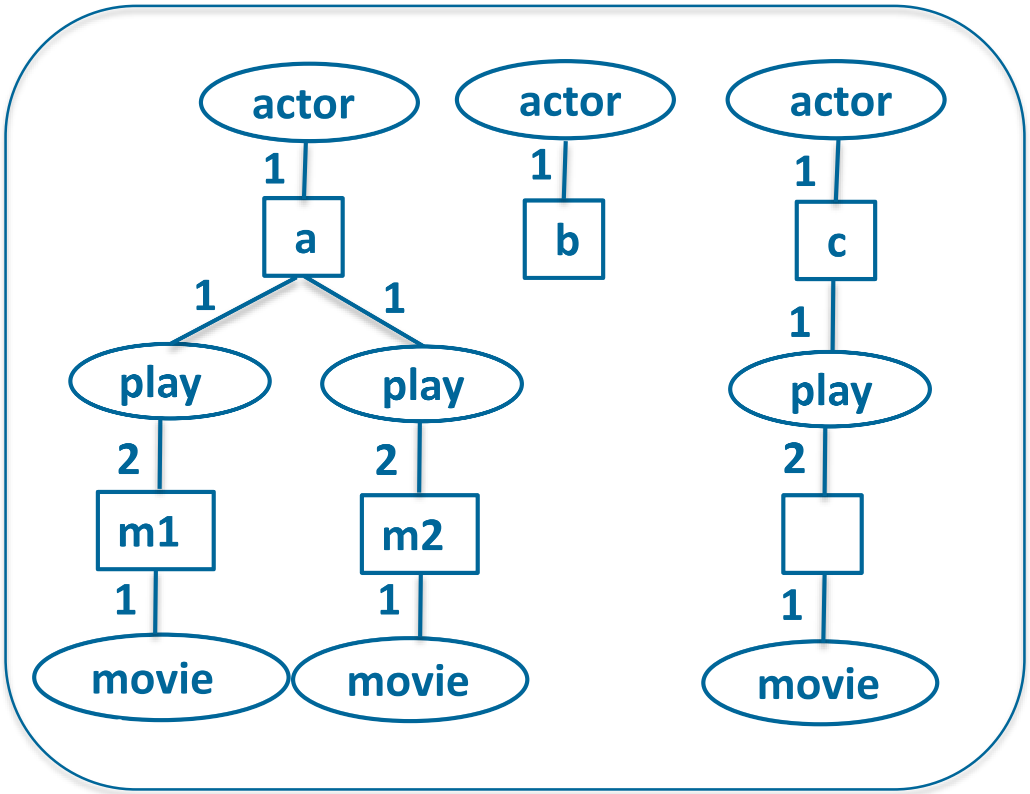
- 1 sommet par terme
  étiqueté par le terme si c'est une constante
- 1 sommet par atome
  étiqueté par le prédicat de l'atome
- les arêtes lient chaque sommet atome aux sommets termes qui représentent ses arguments
- les arêtes incidentes à un sommet atome sont totalement ordonnées (ce qu'on peut représenter par une numérotation)

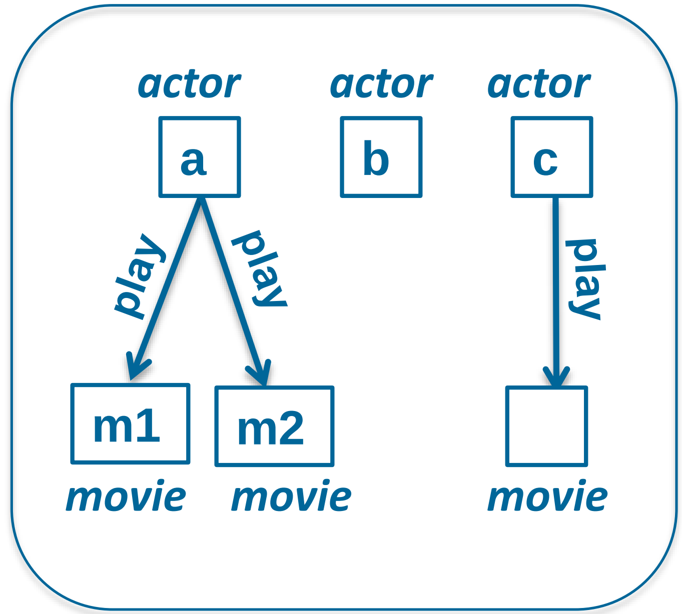
Plus précisément :

À un ensemble d'atomes F, on associe un (multi-)graphe biparti $(V_T, V_A, E, label)$ tel que :

- V_T : ensemble des sommets termes
  (on a une bijection b_T de l'ensemble des termes de F vers V_T)
- **V_A** : ensemble des sommets atomes
  (on a une bijection b_A de l'ensemble des atomes de F vers V_A)
- **E** : multi-ensemble des arêtes : pour chaque atome $A = p(t_1, ..., t_k)$ de F,
  on a k arêtes entre $b_A(A)$ et chacun des $b_T(t_i)$
- **label** : fonction d'étiquetage qui vérifie :
  - tout sommet terme $b_T(t)$ est étiqueté par t si t est une constante, sinon il n'est pas étiqueté
  - tout sommet atome $b_A(A)$ avec $A = p(t_1, ..., t_k)$ est étiqueté par p
    et toute arête $(b_A(A), b_T(t_i))$ est étiquetée par i

movie(m1), movie(m2), actor(a), actor(b), actor(c), play(a,m1), play(a,m2), movie(x), play(c,x)



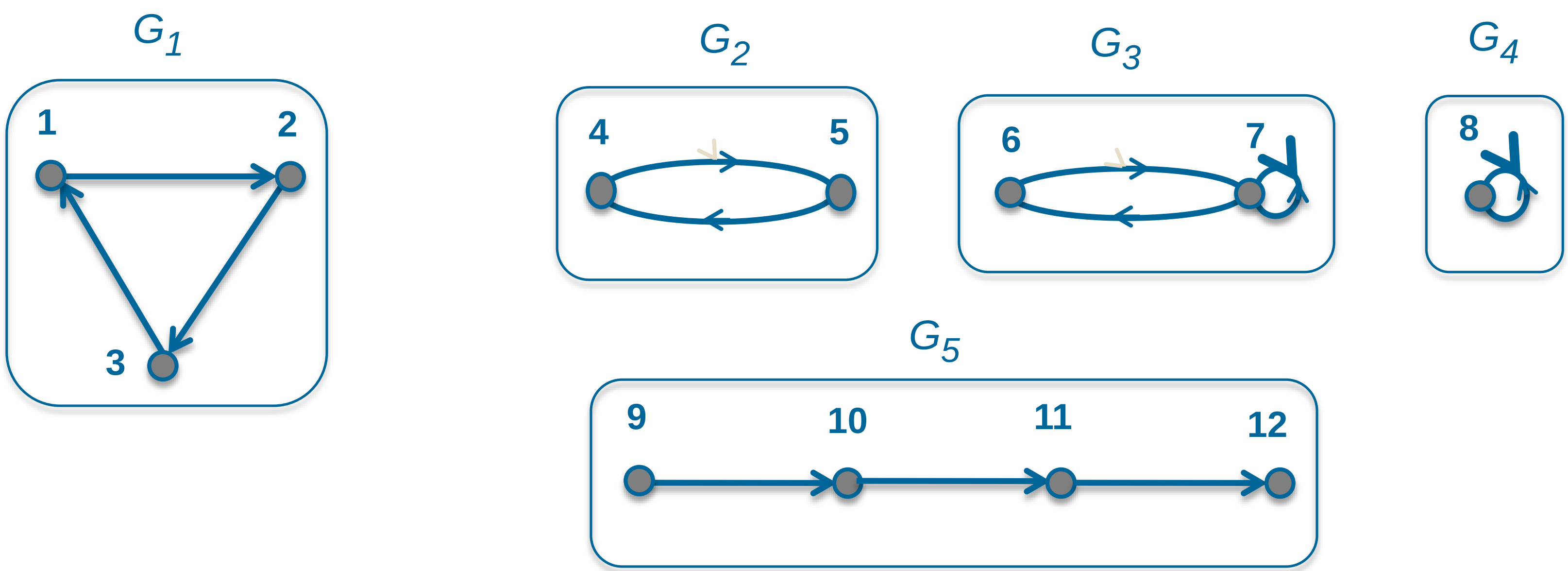


graphe plus simple
(car prédicats d'arité ≤ 2)

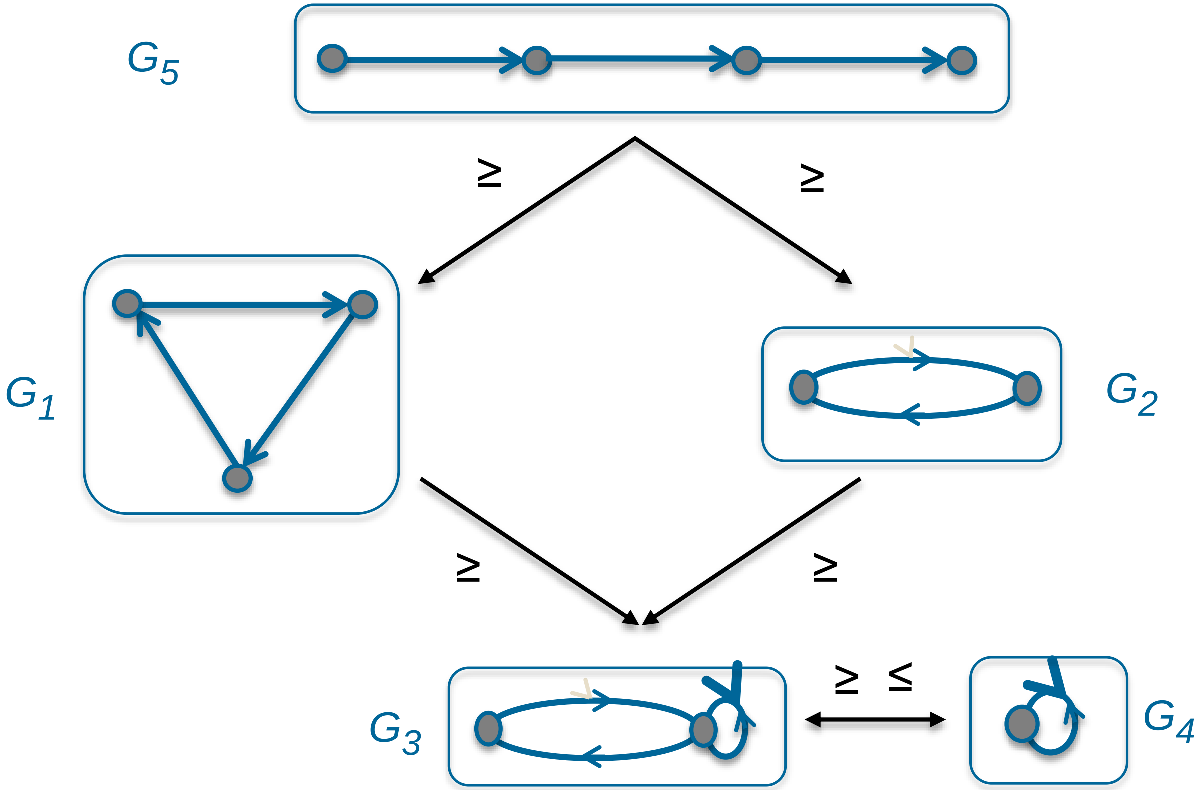
graphe biparti associé à la vision « hypergraphe »

GRAPH HOMOMORPHISMS (1)

- Let $G_1=(V_1,E_1)$ to $G_2=(V_2,E_2)$ be classical graphs.
  Homomorphism *h* from G_1 to G_2: mapping from V_1 to V_2 s. t.
  for every edge (u,v) in E_1, (h(u),h(v)) is in E_2



Find the homomorphisms between these graphs



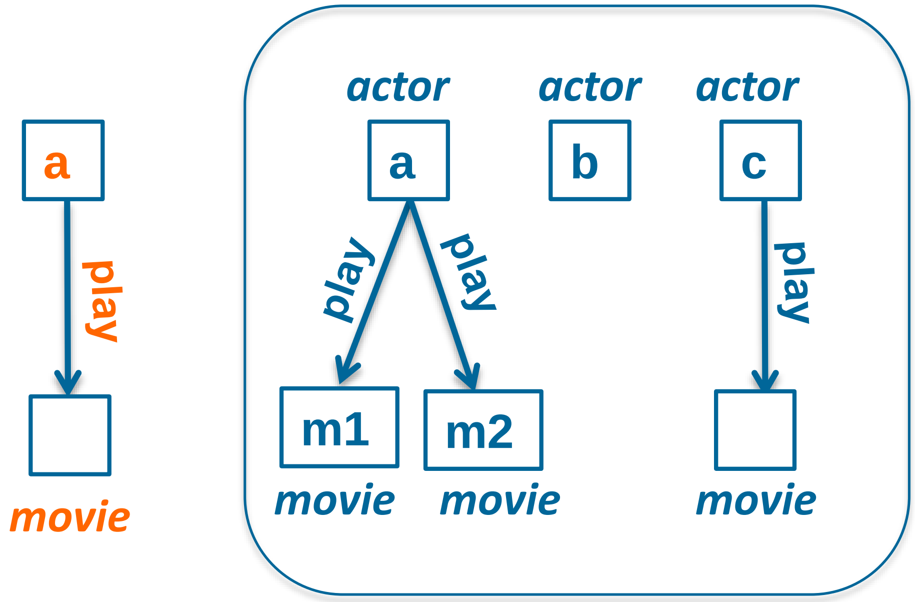
Here, $F \geq G$ means

« F maps to G by homomorphism »

GRAPH HOMOMORPHISMS (2)

- Let $G_1=(V_1,E_1)$ to $G_2=(V_2,E_2)$ be classical graphs.
  Homomorphism *h* from G_1 to G_2: mapping from V_1 to V_2 s. t.
  for every edge (u,v) in E_1, (h(u),h(v)) is in E_2

- If there are labels: they have to be ``kept'' as well

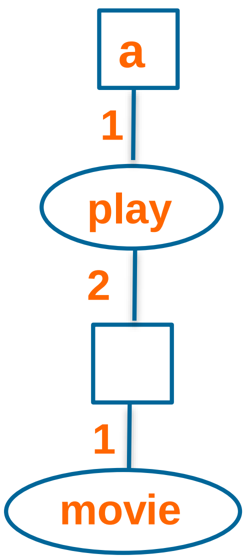


q *F*

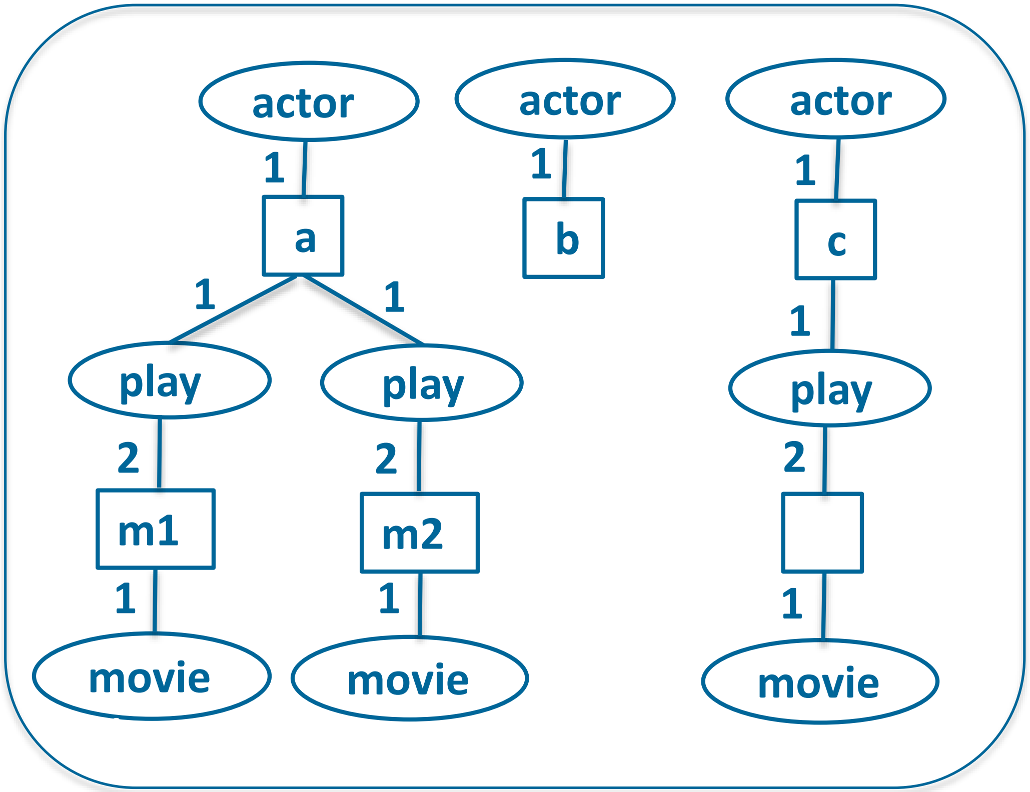
GRAPH HOMOMORPHISMS (3)

- Let $G_1=(V_1,E_1)$ to $G_2=(V_2,E_2)$ be classical graphs.
  Homomorphism *h* from G_1 to G_2: mapping from V_1 to V_2 s. t.
  for every edge (u,v) in E_1, (h(u),h(v)) is in E_2

- If there are labels: they have to be ``kept'' as well



q



F

Homomorphisms on graphs associated with atom sets

- Let *graph* be the translation from a set of atoms to a bipartite (multi-)graph
- Let F_1 and F_2 be two sets of atoms, and: $G_1 = graph(F_1) = (V_{T1}, V_{A1}, E_1, label_1)$

  $G_2 = graph(F_2) = (V_{T2}, V_{A2}, E_2, label_2)$

- A homomorphism from G_1 to G_2 is a mapping h from $V_{T1} \cup V_{A1}$ to $V_{T2} \cup V_{A2}$ such that:
  - for all $v \in V_{T1}$, $h(v) \in V_{T2}$; and for all $v \in V_{A1}$, $h(v) \in V_{A2}$
  - for all $v \in V_{T1} \cup V_{A1}$, $label(h(v)) = label(v)$ [when label(v) is defined]
  - for all $e = (a,t) \in E_1$, $h(e) = (h(a),h(t)) \in E_2$, and $label(e) = label(h(e))$

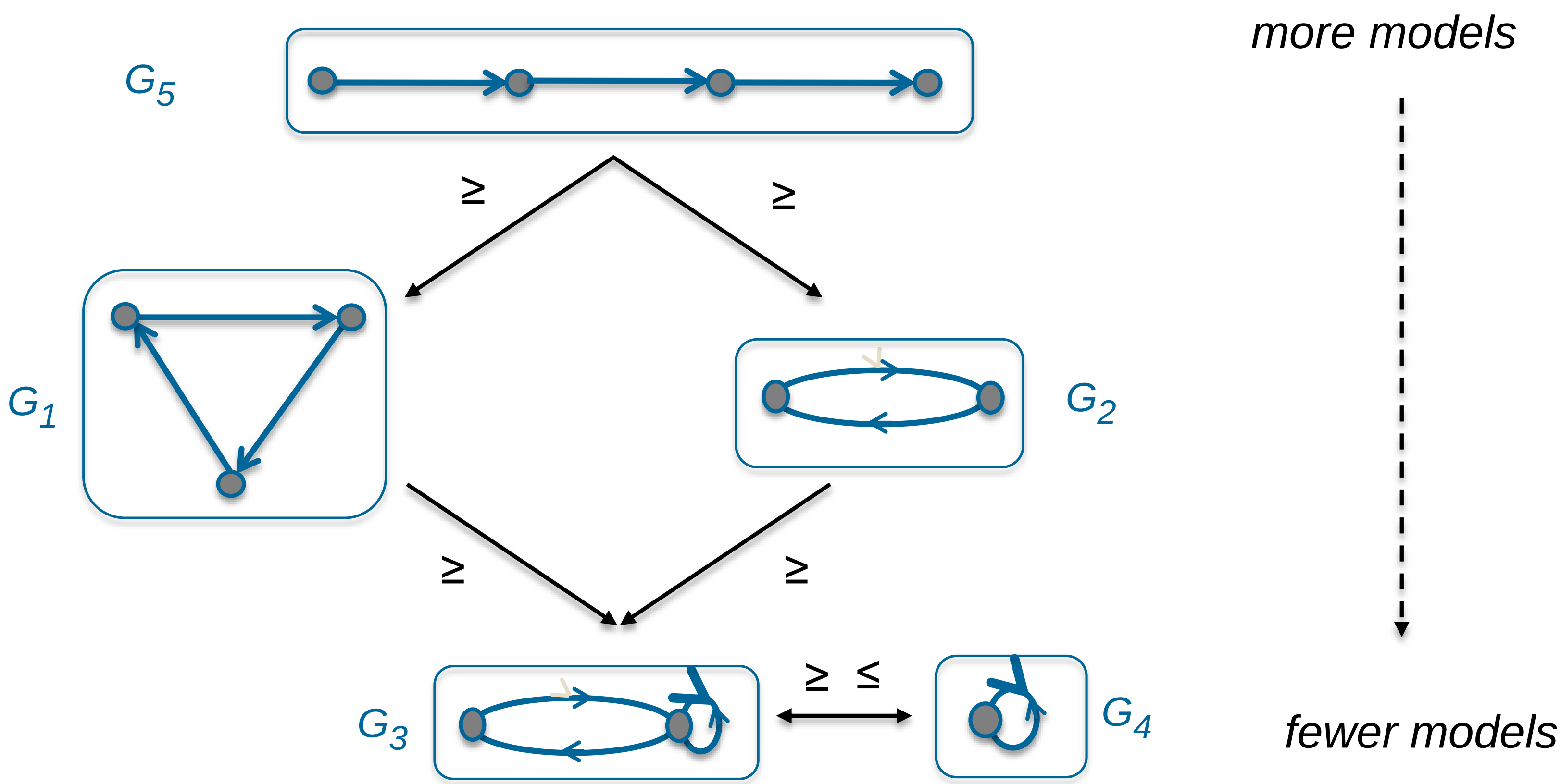
- Reminder: a homomorphism from F_1 to F_2 is a mapping h from variables(F_1) to terms(F_2) such that: for all $p(t_1,\ldots,t_k) \in F_1$, $h(p(t_1,\ldots,t_k)) \in F_2$

  [where $h(p(t_1,\ldots,t_k)) = p(h(t_1),\ldots,h(t_k))$]

There is a homomorphism from F_1 to F_2

if and only if

there is a [graph] homomorphism from graph(F_1) to graph(F_2)



Here, F ≥ G means « F maps to G by homomorphism »

If F maps to G *and* G maps to F, we say that they are homomorphically equivalent
This notion exactly corresponds to logical equivalence

Isomorphism of sets of atoms / Graphs

- Let f and g be sets of atoms
  Isomorphism h from f to g: bijective mapping from *variables(f)* to *variables(g)* such that $h(f) = g$

  When f and g are isomorphic :
  we also say that f and g are ``equal up to a bijective variable renaming''

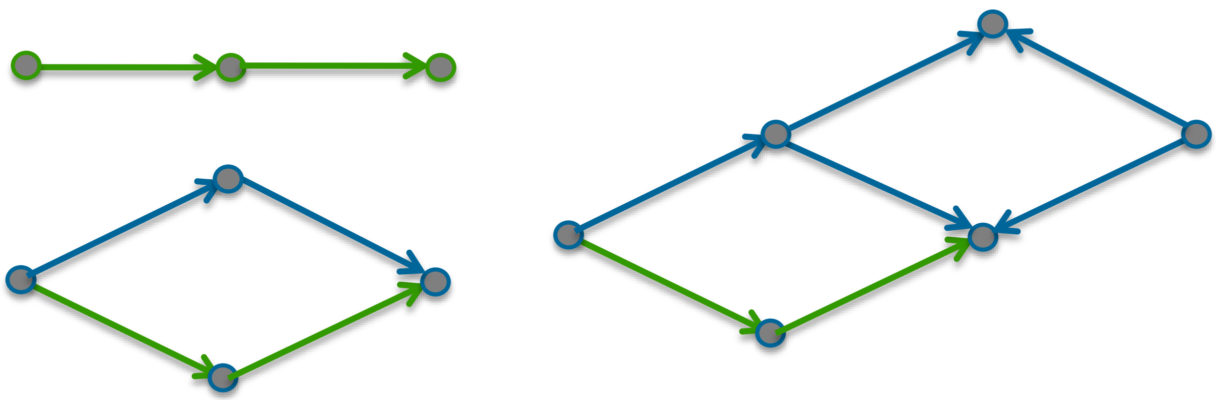
- Let $G_1=(V_1,E_1)$ to $G_2=(V_2,E_2)$ be classical graphs
  Isomorphism h from G_1 to G_2 : bijective mapping from V_1 to V_2 such that for all vertices u,v in V_1, $(u,v) \in E_1$ if and only if $(h(u),h(v)) \in E_2$

  If the graphs are labeled:
  $label(u) = label(h(u))$ for all $u \in V_1$
  $label((u,v)) = label(h(u),h(v))$ for all $(u,v) \in E_1$

It may happen that f and g are equivalent *but not* isomorphic
Intuitively, it means that at least one of them contains « redundant » information

REMOVING REDUNDANCIES: GETTING TO THE *CORE*

- A **core** is a set of atoms that is not equivalent to any of its strict subsets (i.e., it does not map by homomorphism to one of its strict subsets).
- Given *f* a (finite) set of atoms,
  the **core of *f*** is a minimal subset of *f* equivalent to *f*.
  It may happen that *f* has several cores, but they are all isomorphic.
  Hence, we can say « the » core of *f*.
- If *f* and *g* are equivalent, the core of *f* is isomorphic to the core of *g*.



Exercice : find an algorithm to compute the core of an atom set

Conclusion

- Le fragment logique existentiel conjonctif est fortement lié aux (hyper)graphes (donc aux modèles de données graphes, comme RDF)

- Les notions d'*homomorphisme* et de *core* sont fondamentales pour de nombreux problèmes sur les bases de faits et les requêtes

  Par exemple : optimisation de requêtes conjonctives
  - Déterminer si deux requêtes sont équivalentes
    - but : exécuter la plus simple pour le SGBD
  - Minimiser une requête : supprimer toutes ses redondances
    - but : accélérer l'évaluation de la requête
  - Déterminer si une requête Q_1 est plus spécifique qu'une requête Q_2
    - but : accélérer l'évaluation de Q_1 (connaissant les réponses à Q_2)

  [On parle de problèmes d'optimisation « statique » de requêtes, au sens où ils sont indépendants d'une base de données (ou base de faits) particulière]