VUE LOGIQUE (ET GRAPHE) DES FAITS ET DES REQUÊTES CONJONCTIVES

THÉORIE DES BASES DE CONNAISSANCES
HMIN312M

DATA / FACTS

Relational database

Movie

m_id	
m1	…
m2	…
?x	

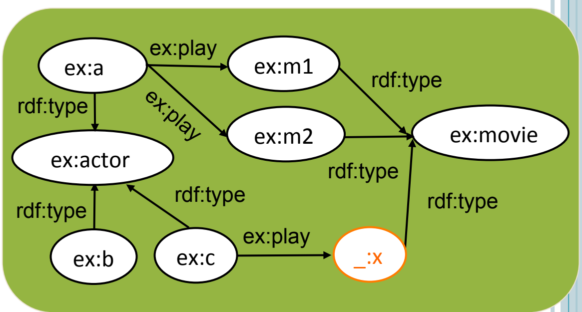
Actor

a_id	
a	…
b	…
c	

Play

a_id	m_id
a	m1
a	m2
c	?x

RDF (and other kinds of labeled graphs) ***Etc.***



Abstraction in first-order logic (FOL)

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

Fact base

Normalized form:
$\exists x_1..x_n$
(*conjunction of atoms with all variables in* $x_1..x_n$)

Conjunctive Queries (CQ) – The Basic Database Queries

« find those who play in a movie »

$q(x) = \exists y\ (movie(y) \wedge play(x, y))$ — First Order Logic

A conjunctive query (**CQ**) $q(x_1 \dots x_k)$ has the form $\exists x_{k+1},\dots,x_m\ A_1 \wedge \dots \wedge A_p$
where $A_1,\dots,A_p$ are atoms over variables $x_1,\dots,x_m$
and $x_1 \dots x_k$ are free variables (defining the answer part)

If $k = 0$, q is a Boolean conjunctive query (**BCQ**)
(thus has the same form as our notion of a fact base)

answer(x) ← movie(y), play(x,y) — Datalog notation

SELECT ... FROM ... WHERE <*join conditions*> — SQL

SELECT ... WHERE <*graph pattern*> — SPARQL

Key notion: Homomorphism

$q(x) = \exists y\ (movie(y) \wedge play(x, y))$

movie(y)
play(x, y)

F: movie(m1), movie(m2), movie(x0), actor(a), actor(b), actor(c), play(a,m1), play(a,m2), play(c,x0)

Homomorphism *h* from *q* to *F*:
substitution of var(*q*) *by terms(F)*
such that h(q) ⊆ *F*

h1 : x → a, y → m1	h1(q) = movie(m1) ∧ play(a, m1)
h2 : x → a, y → m2	h2(q) = movie(m2) ∧ play(a, m2)
h3 : x → c, y → x0	h3(q) = movie(x0) ∧ play(c, x0)

Answers: obtained by restricting the domains of homomorphisms to the variables of interest
(usually, only mappings of these variables to constants are kept)

x = a
x = c

Answers to a Conjunctive Query

- The answer to a BCQ Q in F is *yes* if $F \models Q$
  yes = ()

- A tuple $(a_1, \ldots, a_k)$ of *constants* is an answer to $Q(x_1,\ldots,x_k)$ with respect to F

if $F \models Q[a_1,\ldots,a_k]$, where $Q[a_1,\ldots,a_k]$ is obtained from $Q(x_1,\ldots,x_k)$ by replacing each x_i by a_i.

- Let F and Q be seen as sets of atoms. A homomorphism *h* from Q to F is a mapping from *variables*(Q) to *terms*(F) such that $h(Q) \subseteq F$

> $F \models Q()$ iff *Q* can be mapped by homomorphism to *F*
>
> $(a_1, \ldots, a_k)$ is an answer to $Q(x_1,\ldots,x_k)$ on *F* iff there is a homomorphism from *Q* to *F* that maps each x_i to a_i

Exercice 1 : CQ en SQL et en logique sur un exemple

On considère une base de données relationnelle qui gère des abonnés, qui peuvent avoir des cartes d'accès, en cours de validité ou pas. Le schéma de la base a deux relations :

Coords [id_abonné, nom, prénom, date_naissance, ville], où id_abonné est une clé
Cartes [id_abonné, id_carte, validité]

Pour trouver les dates de naissance de tous les abonnés de Montpellier qui ont une carte d'accès en cours de validité, on fait la requête SQL suivante :

```
SELECT DISTINCT Coords.date_naissance
FROM Coords, Cartes
WHERE Coords.ville = MPL AND Cartes.validité = true
                    AND Coords.id_abonné = Cartes.id_abonné
```

1. Traduire les relations du schéma en prédicats et la requête en une requête conjonctive.
2. Soit l'instance de base données suivante :

   Coords = [[1, N1,P1,D1,MPL],[2,N2,P2,D2,MPL],[3,N3,P3,D3,MPL],[4,N4,P4,D4,MARS]]
   Cartes = [[1,401,false],[1,502,true],[1,503,true],[2,404,false],[4,509,true]]

   Définir la base de faits associée et déterminer les réponses à la requête

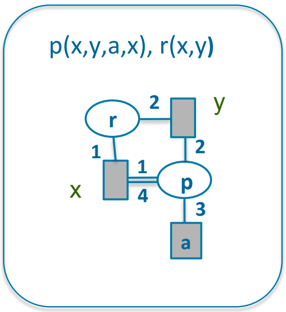
LABELLED HYPERGRAPH / GRAPH REPRESENTATION

- A fact base (or a BCQ) can be seen as a **set of atoms**

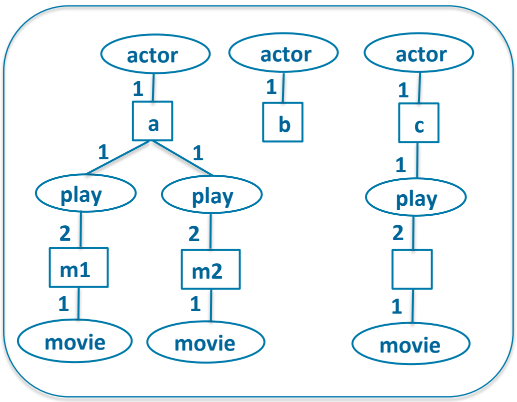
movie(m1), movie(m2), actor(a), actor(b), actor(c),
play(a,m1), play(a,m2), movie(x), play(c,x)

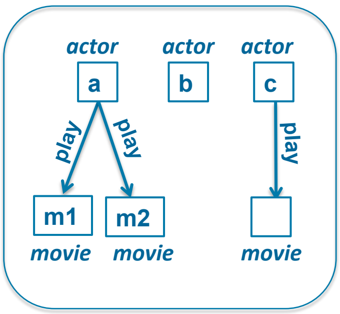
→ and a set of atoms is naturally seen as a **bipartite (multi-)graph**

- one (labelled) node per term
  variable: no label
  constant: labelled by itself
- one (labelled) node per atom
  label: the atom's predicate
- totally ordered edges



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

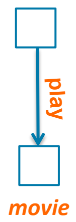




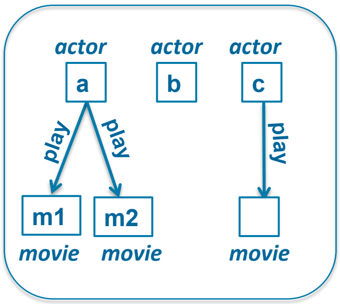
If predicates are at most binary:
atom nodes can be replaced by **labelled directed edges**

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



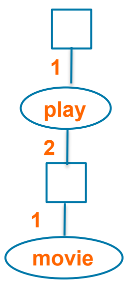
q



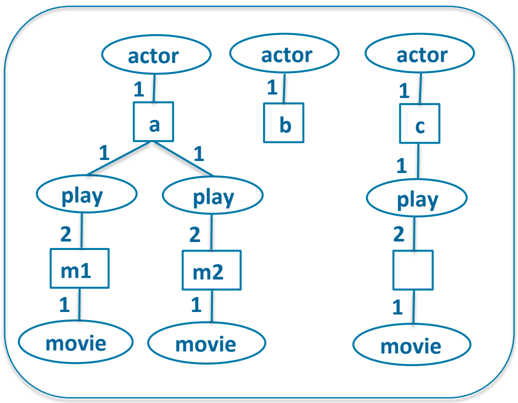
F

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

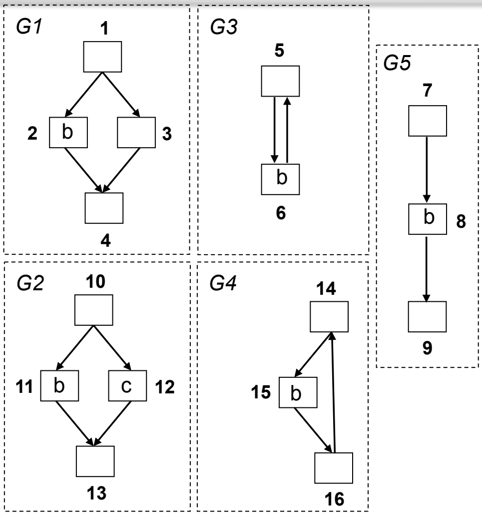


q



F

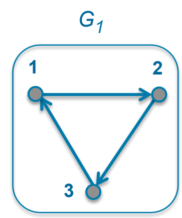
Exercice 2 : Homomorphismes



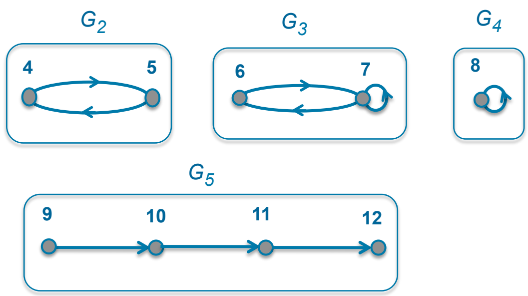
1. En supposant qu'on n'ait que le prédicat binaire *p*, quelles sont les formules logiques associées à ces graphes ?
2. Les classifier par homomorphisme (en utilisant la vue logique ou graphe)

Positive conjunctive existential fragment of FOL: FOL(∃,∧)

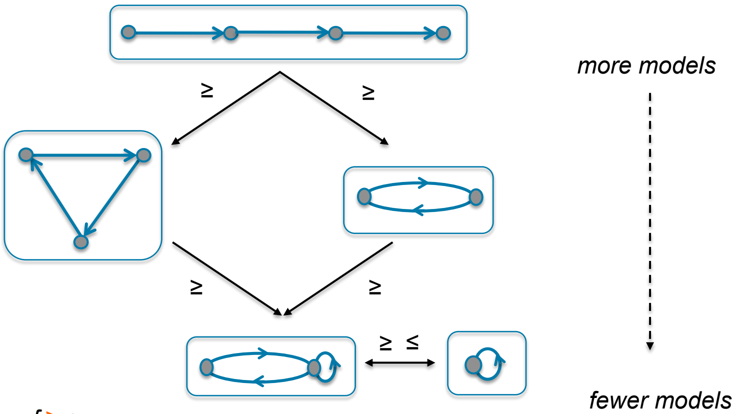
- Allows to express **facts** and (Boolean**)** **conjunctive queries**
- Such formulas can be seen as sets of atoms, labelled graphs, relational structures, ...
- **Homomorphism** is a fundamental notion in this fragment



$\exists x\ \exists y\ \exists z$
$(p(x,y) \wedge p(y,z) \wedge p(z,x))$



Find the homomorphisms between these graphs



f ≥ *g*

f maps to *g* by homomorphism

f is « more general » than *g*

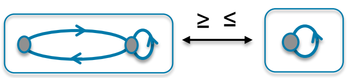
g is « more specific » than *f*

≥ is not an order
but a preorder
(a reflexive and transitive relation)

ISOMORPHISM ON SETS OF ATOMS

- Let *f* and *g* in FOL(∃,∧) seen as sets of atoms
- Isomorphism *h* from *f* to *g*: bijective mapping from *var(f)* to *var(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''
- Equivalent definition of isomorphism: homomorphism *h* from *f* to *g* such that h^{-1} is a homomorphism from *g* to *f*

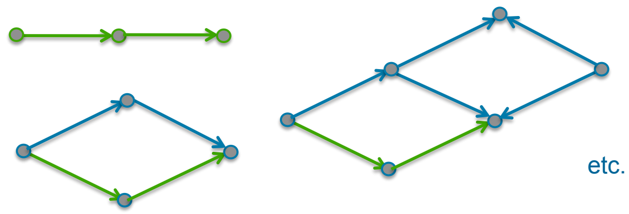


equivalent but not isomorphic

Equivalence / Core

- If $f \geq g$ and $g \geq f$, they are called (homomorphically) equivalent
- A core is a set of atoms that is not equivalent to any of its strict subsets
- Given f seen as a set of atoms, the core of f is a minimal subset of f equivalent to f

- Consider the (infinite) set of all possible finite sets of atoms on a vocabulary, structured by $\leq$. Then each equivalence class has a unique minimal element (up to bijective variable renaming)



Interpretations / Models (1)

- **Vocabulary** $\mathcal{V} = (\mathcal{P}, \mathcal{C})$, where $\mathcal{P}$ = finite set of predicates
  $\mathcal{C}$ = set of constants
- **Interpretation** $\mathcal{I} = (D_\mathcal{I}, .^\mathcal{I})$ of $\mathcal{V}$, where
  $D_\mathcal{I} \neq \emptyset$ (domain)
  for all c in $\mathcal{C}$, $c^\mathcal{I} \in D_\mathcal{I}$
  for all p in $\mathcal{P}$ with arity k, $p^\mathcal{I} \subseteq D_\mathcal{I}^k$
- Simplifying assumption (which has no incidence in the following):
  $\mathcal{C} \subseteq D_\mathcal{I}$ and for all c in $\mathcal{C}$, $c^\mathcal{I} = c$

> $\mathcal{V} = (\{p_{/2}, r_{/3}\}, \{a, b\})$
>
> $\mathcal{I}$: $D_\mathcal{I} = \{a, b, d_1\}$ $p^\mathcal{I} = \{(b, a), (b, d_1), (d_1, b)\}$
> $r^\mathcal{I} = \{(d_1, d_1, a)\}$

- $\mathcal{I}$ is a **model** of f (built on $\mathcal{V}$) if f is true in $\mathcal{I}$

By default, assume that formulas are closed

INTERPRETATIONS / MODELS (2)

- Let f in FOL($\exists$,$\wedge$). $\mathcal{I}$ is a model of f iff there is a mapping v from var(f) to $D^{\mathcal{I}}$

  such that for all $p(e_1, \ldots, e_k)$ in f, $(v(e_1), \ldots, v(e_k))$ in $p^{\mathcal{I}}$

  --- where $v(c) = c$ for each constant c ---

  v is called a ``good assignement''

$\mathcal{I}$: $D_{\mathcal{I}} = \{a, b, d_1\}$ $p^{\mathcal{I}} = \{(b, a), (b, d_1), (d_1, b)\}$

$r^{\mathcal{I}} = \{(d_1, d_1, a)\}$

$f = \exists x \exists y \exists z\ (\ p(x, y) \wedge p(y, z) \wedge r(x, z, a)\ )$

v: $x \mapsto d_1$ $y \mapsto b$ $z \mapsto d_1$

- Interpretations can be seen as **sets of atoms** (with elements from $D^{\mathcal{I}} \setminus \mathcal{C}$ seen as variables)

$p(b,a)$, $p(b,x_1)$, $p(x_1,b)$, $r(x_1, x_1,a)$

- $\mathcal{I}$ is a **model** of f iff there is a **homomorphism** from f to $\mathcal{I}$ (seen as a set of atoms)

HOMOMORPHISMS AGAIN AND AGAIN

- One can also define **homomorphisms** between **interpretations**

  Homomorphism h from $\mathcal{I}_1=(D_1, .^{\mathcal{I}1})$ to $\mathcal{I}_2 = (D_2, .^{\mathcal{I}2})$:

  mapping from D_1 to D_2 such that:

  for all c in $\mathcal{C}$, $h(c^{\mathcal{I}1}) = c^{\mathcal{I}2}$ with our assumption: $h(c) = c$

  for all p in $\mathcal{P}$ and $(t_1 \ldots t_k)$ in $p^{\mathcal{I}1}$, $(h(t_1) \ldots h(t_k))$ in $p^{\mathcal{I}2}$

- We have:

  If $\mathcal{I}_1 \geq \mathcal{I}_2$ then, for any f in FOL($\exists$,$\wedge$), $\mathcal{I}_1$ model of $f \Rightarrow \mathcal{I}_2$ model of f

  Indeed: $f \geq \mathcal{I}_1$ and $\mathcal{I}_1 \geq \mathcal{I}_2$, hence $f \geq \mathcal{I}_2$

MODEL "ISOMORPHIC" TO A FOL(∃ ,∧) FORMULA

To a formula f *in* FOL(∃ ,∧) , we assign its **isomorphic model** (also called **canonical model**) M_f:

- the domain is in bijection with *terms(f)* $\cup\, \mathcal{C}$
  (to simplify, we can consider that this bijection is the identity)
- for all c in $\mathcal{C}$, $c^{Mf} = c$
- for all p in $\mathcal{P}$, if p occurs in f then $p^{Mf} = \{(t_1 \ldots t_k) \mid p(t_1 \ldots t_k) \text{ in } f\}$
  otherwise $p^{Mf} = \emptyset$

Remark: the sets of atoms associated with M_f and f are isomorphic

We check that M_f is indeed a model of f

$f = \exists x \exists y \exists z\ (\ p(x,y) \wedge p(y,z) \wedge r(x,z,a)\ )$

M_f: $D = \{a, d_x, d_y, d_z\}$
$p^{Mf} = \{\ (d_x,d_y), (d_y,d_z)\ \}$
$r^{Mf} = \{\ (d_x, d_z, a)\ \}$

NICE SEMANTIC PROPERTIES OF FOL(∃ ,∧)

- M_f is **universal**, i.e., for all M' model of f, $M_f \geq M'$

Proof: Let M' model of f. Then, $f \geq M'$. Since M_f « isomorphic » to f, $M_f \geq M'$

- $g \vDash f$ (i.e., every model of g is a model of f) iff
  $f \geq M_g$ (the canonical model of g is a model of f) iff
  $f \geq g$ (there is a homomorphism from f to g)

Proof:

(1 ⇒ 2): Assume $g \vDash f$. In particular M_g is a model of f, hence $f \geq M_g$

(2 ⇒ 1): Assume $f \geq M_g$. Since M_g is universal: for any M' model of g, $f \geq M'$, i.e., M' is a model of f, hence $g \vDash f$

(3 ⇒ 2) and (2 ⇒ 3) : since g and M_g are isomorphic, one has $M_g \geq g$ and $g \geq M_g$
We conclude by the transitivity of ≥

Complexity of deciding logical entailment in FOL(∃, ∧)

- FOL(∃, ∧) entailment is NP-complete:

  Input: *f* and *g* in FOL(∃, ∧)
  Question: ***g* ⊨ *f*** ?

  Membership to NP: polynomial certificate
  Hardness for NP: for instance by reduction from 3-coloring

- Hence BCQ answering is NP-complete:

  Input: a BCQ *Q* and a fact base *F*
  Question: ***F* ⊨ *Q*** ?

Naive algorithm exponential in |variables(Q)|

However, we can often assume *Q* is *very very* small with respect to *F*!

- Hence the distinction between two kinds of complexity:
  - Combined complexity: *Q* and *F* are both part of the input (= usual complexity)
  - Data complexity: only *F* is part of the input (*Q* is fixed)

BCQ answering is polynomial in data complexity (even in AC_0)

Exercice 3 : inclusion de requêtes

Etant données deux requêtes conjonctives booléennes, Q1 et Q2, on dit que Q1 est **incluse** dans Q2 (notation Q1 ⊑ Q2) si l'ensemble des bases de faits qui répondent oui à Q1 est inclus dans l'ensemble des bases de faits qui répondent oui à Q2.

Utiliser les notions vues dans ce cours pour prouver que :

Q1 ⊑ Q2

ssi

Q2 ≥ Q1

(il existe un homomorphisme de Q2 dans Q1)

Question subsidiaire : pour des requêtes quelconques, Q1 est incluse dans Q2 si pour toute base de faits, l'ensemble des réponses à Q1 est inclus dans l'ensemble des réponses à Q2. Comment étendre le résultat précédent ?