Jean Goubault-Larrecq

Bases de données

Bases de données, langages de requêtes, théorème de Codd

Bases de données relationnelles

Schémas de bases de données relationnelles

- On se donne un ensemble Attr d'**attributs**
  qu'on suppose **totalement ordonné**
  et **infini dénombrable** (oublié dans le Alice)
- Un **schéma** de relation est un ensemble fini d'attributs
- Un **schéma de base de données** est un couple (T, schema) où
  — T est un ensemble fini de **relations** (ou **tables**)
  — schema : $T \to \mathbf{P}_{\text{fin}}(\text{Attr})$
- Ex: T = {Movies, Location, Pariscope}   [Alice p.30]
  schema(Movies) = {Title, Director, Actor}
  schema(Location) = {Theater, Address, Phone}
  schema(Pariscope) = {Theater, Title, Schedule}

Bases de données relationnelles

- Pour chaque attribut a, dom(a) est un ensemble **fini non vide** (les **valeurs** possibles de l'attribut)
- On supposera que dom(a) est le même pour tout a, pour simplifier;
  soit dom cet ensemble
- Une **instance I** d'un schéma (T, schema) (= **base de données relationnelle** sur ce schéma) est la donnée, pour chaque relation R dans T, d'un ensemble **fini I**(R) de **rangées** (ou **uplets**)
  $$t : \text{schema}(R) \to \text{dom}$$
- Comme Attr est totalement ordonné, on identifie chaque rangée à un élément de dom^n, où $n = |\text{schema}(R)|$
  — mais attention à être clairs sur votre perspective, attributs nommés ou non

Exemple de BD relationnelle

Movies

Title	Director	Actor
The Trouble with Harry	Hitchcock	Gwenn
The Trouble with Harry	Hitchcock	Forsythe
The Trouble with Harry	Hitchcock	MacLaine
The Trouble with Harry	Hitchcock	Hitchcock
...	...	...
Cries and Whispers	Bergman	Andersson
Cries and Whispers	Bergman	Sylwan
Cries and Whispers	Bergman	Thulin
Cries and Whispers	Bergman	Ullman

Location

Theater	Address	Phone Number
Gaumont Opéra	31 bd. des Italiens	47 42 60 33
Saint André des Arts	30 rue Saint André des Arts	43 26 48 18
Le Champo	51 rue des Ecoles	43 54 51 60
...	...	...
Georges V	144 av. des Champs-Elysées	45 62 41 46
Les 7 Montparnassiens	98 bd. du Montparnasse	43 20 32 20

Pariscope

Theater	Title	Schedule
Gaumont Opéra	Cries and Whispers	20:30
Saint André des Arts	The Trouble with Harry	20:15
Georges V	Cries and Whispers	22:15
...	...	...
Les 7 Montparnassiens	Cries and Whispers	20:45

Figure 3.1: The **CINEMA** database

[Alice p.30]

Que faire avec une BD?

- L’**interroger** = évaluer des **requêtes**
  Ce sera l’essentiel de ce que nous étudierons
  Plusieurs langages: SQL, **RA**, **logique du premier ordre**, Datalog (avec ou sans négation), etc.
  Pouvoir expressif
- La **modifier**: insertion, suppression
  Questions d’implémentation, propriétés ACID
  Je n’insisterai pas là-dessus [Alice chapitre 22]

Requêtes et logique du premier ordre

Formules de la logique du premier ordre

- Fixons un schéma (T, schema)
- Une **formule** *F* de la logique du premier ordre (**FOL**), dans notre cadre, sera:
  — sur le langage de prédicats $T \cup \{=\}$
  arité de R dans T = |schema(R)|
  — à connecteurs logiques parmi $\wedge, \top, \vee, \bot, \neg, \exists$
  (d'autres si vous voulez, mais soyez précis: le Alice oublie le $\top$, par exemple, et il est nécessaire pour faire les conjonctions 0-aires)
  — **pas** de symboles de fonctions
  autres que les **constantes**
- On notera free(*F*) l'ensemble de ses **variables libres** (déf. usuelle)
- Les **termes** sont donc les constantes ou les variables

Requêtes

- Fixons un schéma (T, schema)
- Une **requête** $\{\underline{x} \mid F\}$ est formée:
  — d'une formule F de la logique du premier ordre
  — d'un **uplet libre** $\underline{x}$ = liste de variables distinctes 2 à 2
  ([Alice déf. 4.2.1 p.41] est plus libérale: un uplet libre
  y est formé de variables et de constantes)
- et les variables libres de F sont **exactement** celles de $\underline{x}$
- Le **domaine actif** adom(F), resp. adom($\{\underline{x} \mid F\}$)
  est l'ensemble des constantes apparaissant dans F

Exemples de requêtes

- $\{(x_{\text{th}}, x_{\text{ad}}) \mid \exists x_{\text{ti}}, x_{\text{ac}}, x_{\text{s}}, x_{\text{p}},$ [Alice ex. 4.1.1 p.41]
  $\text{Movies}(x_{\text{ti}}, \text{« Bergman »}, x_{\text{ac}}) \land$
  $\text{Pariscope}(x_{\text{th}}, x_{\text{ti}}, x_{\text{s}}) \land \text{Location}(x_{\text{th}}, x_{\text{ad}}, x_{\text{p}})\}$
  (requête **conjonctive**: n'utilise que $\land$, $\exists$)
- Son domaine actif est {« Bergman »}

Sémantique (1/2)

- Une **valuation** ν (**relative** au domaine dom et pour une formule F) est une fonction qui à chaque variable libre de F
  associe un élément de dom

- On note $t\nu$: $= \nu(t)$ si t variable, $= t$ si t constante
  Pour tout uplet $\underline{u} = (u_1, \ldots, u_n)$ de termes (=vars ou constantes),
  $\underline{u}\nu = (u_1\nu, \ldots, u_n\nu)$

- Une instance **I** satisfait F pour la valuation ν, en notation $\mathbf{I} \models_{\mathrm{dom}} F[\nu]$ ssi (définition par récurrence sur F usuelle en logique):
  voir transparent suivant

- Attention: (1) **dépend** du domaine dom — d'où le « dom » en indice
  (2) n'a de sens que si **adom(*F*)** $\subseteq$ **dom**

Sémantique (2/2)

- $\mathbf{I} \vDash_{\mathrm{dom}} \mathrm{R}(\underline{u})[\nu]$ ssi $\underline{u}\nu$ est dans $\mathbf{I}(\mathrm{R})$
- $\mathbf{I} \vDash_{\mathrm{dom}} (s{=}t)[\nu]$ ssi $s\nu{=}t\nu$, $\mathbf{I} \vDash_{\mathrm{dom}} \bot[\nu]$ jamais, $\mathbf{I} \vDash_{\mathrm{dom}} \top[\nu]$ toujours
- $\mathbf{I} \vDash_{\mathrm{dom}} (F \wedge G)[\nu]$ ssi $\mathbf{I} \vDash_{\mathrm{dom}} F[\nu_{|\mathrm{free}(F)}]$ et $\mathbf{I} \vDash_{\mathrm{dom}} G[\nu_{|\mathrm{free}(G)}]$
- $\mathbf{I} \vDash_{\mathrm{dom}} (F \vee G)[\nu]$ ssi $\mathbf{I} \vDash_{\mathrm{dom}} F[\nu_{|\mathrm{free}(F)}]$ ou $\mathbf{I} \vDash_{\mathrm{dom}} G[\nu_{|\mathrm{free}(G)}]$
- $\mathbf{I} \vDash_{\mathrm{dom}} (\neg F)[\nu]$ ssi non ($\mathbf{I} \vDash_{\mathrm{dom}} F[\nu]$)
- $\mathbf{I} \vDash_{\mathrm{dom}} (\exists x \,.\, F)[\nu]$ ssi il existe a **dans dom** tel que

  $$\mathbf{I} \vDash_{\mathrm{dom}} F[\nu[x{:=}a]]$$

  où $\nu[x{:=}a]$ envoie x vers a et tout $y{\neq}x$ vers $\nu(y)$

Sémantique des requêtes

- Soit q la requête $\{\underline{x} \mid F\}$
  son **domaine actif** adom(q) est adom(F), par définition
- Soit **I** une instance d'un schéma (T, schema),
  sur un domaine dom contenant adom(q)
- La **sémantique** de q relative à **I**
  est $q(\text{dom},\mathbf{I})$ = l'ensemble des $\underline{x}\nu$, lorsque ν parcourt les valuations (**sur dom**) telles que $\mathbf{I} \vDash_{\text{dom}} F[\nu]$
- Alice le note $q(\mathbf{I})$, mais vu les questions d'indépendance de domaine (plus tard), je préfère mentionner dom explicitement

Exemple

- **I** =

Movies	Title	Director	Actor
	The Trouble with Harry	Hitchcock	Gwenn
	The Trouble with Harry	Hitchcock	Forsythe
	The Trouble with Harry	Hitchcock	MacLaine
	The Trouble with Harry	Hitchcock	Hitchcock
	...	...	...
	Cries and Whispers	Bergman	Andersson
	Cries and Whispers	Bergman	Sylwan
	Cries and Whispers	Bergman	Thulin
	Cries and Whispers	Bergman	Ullman

Location	Theater	Address	Phone Number
	Gaumont Opéra	31 bd. des Italiens	47 42 60 33
	Saint André des Arts	30 rue Saint André des Arts	43 26 48 18
	Le Champo	51 rue des Ecoles	43 54 51 60
	...	...	...
	Georges V	144 av. des Champs-Elysées	45 62 41 46
	Les 7 Montparnassiens	98 bd. du Montparnasse	43 20 32 20

Pariscope	Theater	Title	Schedule
	Gaumont Opéra	Cries and Whispers	20:30
	Saint André des Arts	The Trouble with Harry	20:15
	Georges V	Cries and Whispers	22:15
	...	...	...
	Les 7 Montparnassiens	Cries and Whispers	20:45

Figure 3.1: The **CINEMA** database

- $q = \{(x_{th}, x_{ad}) \mid \exists x_{ti}, x_{ac}, x_s, x_p,$
  $\text{Movies}(x_{ti}, \text{« Bergman »}, x_{ac}) \wedge$
  $\text{Pariscope}(x_{th}, x_{ti}, x_s) \wedge$
  $\text{Location}(x_{th}, x_{ad}, x_p)\}$

- $q(\mathbf{I}) = \{$
  (« Gaumont Opéra »,
  « 31 bd. des Italiens »),
  (« Georges V »,
  « 144 av. des Champs-Elysées »),
  (« Les 7 Montparnassiens »,
  « 98 bd. du Montparnasse »)
  }

Les deux algèbres relationnelles

L'algèbre relationnelle RA (sans noms)

- Version **non nommée**:
  — constantes: les relations R de T, les uplets $\underline{u}$ (= $(u_1,\ldots,u_n)$, dans dom^n)
  — + 5 opérations:
  sélection σ_F (*F* conjonction d'égalités *i*=*j* ou *i*=cst., *i* et *j* entiers;
  Alice autorise aussi les ≠)
  projection π_L (*L* liste d'entiers **distincts deux à deux**)
  produit ×
  différence –
  union ∪
- Les 3 premières opérations forment l'algèbre SPC [Alice, Sec. 4.4, p.52] et ont le même pouvoir expressif que les requêtes conjonctives
- Intérêt: **implémentation**
- Attention: ces opérations sont (implicitement) typées…

L'algèbre relationnelle RA (sans noms): arités

- R : n (pour chaque relation R dans T, d'arité n)
  $(u_1,\ldots,u_n) : n$ (pour chaque uplet)
  $\sigma_F : n \to n$ (F conj. d'égalités i=j ou i=cst., **avec 1≤*i*,*j*≤*n***)
  $\pi_L : n \to$ len L (L liste d'éléments de {1,…,n})
  $\times : m \times n \to m+n$
  $-, \cup : n \times n \to n$ (avec **le même *n*** à gauche de la flèche)
- Les **requêtes RA** sont les expressions bien typées q formées à partir de ces constructions
- Le **domaine actif** adom(q) est l'ensemble des constantes qui apparaissent dans q (dans les uplets $(u_1,\ldots,u_n)$)

L'algèbre relationnelle RA (sans noms): sémantique

- Sémantique $\mathbf{I}[\![q]\!] \in \mathbf{P}_{\text{fin}}(\text{dom}^n)$, où $q : n$

  $\mathbf{I}[\![\text{R}]\!] = \mathbf{I}(\text{R})$

  $\mathbf{I}[\![(u_1,\ldots,u_n)]\!] = \{(u_1,\ldots,u_n)\}$

  $\mathbf{I}[\![\sigma_F(q)]\!] = \{\underline{x} \in \mathbf{I}[\![q]\!] \mid \bigwedge_{i=j\in F} x_i{=}x_j \wedge \bigwedge_{i=\text{cst}\in F} x_i{=}\text{cst}\}$

  $\mathbf{I}[\![\pi_L(q)]\!] = \{(\text{x}_{i[1]},\ldots,\text{x}_{i[k]}) \mid \exists x_{j[1]},\ldots,x_{j[n-k]} \,.\, \underline{x} \in \mathbf{I}[\![q]\!]\}$

  où $L{=}[i[1],\ldots,i[k]]$, et $j[1],\ldots,j[n{-}k]$ énumère les indices de $\{1, \ldots, n\}$ hors de L

  $\mathbf{I}[\![q \times q']\!] = \mathbf{I}[\![q]\!] \times \mathbf{I}[\![q']\!]$

  $\mathbf{I}[\![q{-}q']\!] = \mathbf{I}[\![q]\!] - \mathbf{I}[\![q']\!]$ (vous voyez l'intérêt du typage?)

  $\mathbf{I}[\![q{\cup}q']\!] = \mathbf{I}[\![q]\!] \cup \mathbf{I}[\![q']\!]$

L'algèbre relationnelle RA (avec noms)

- Version **nommée**:
  — constantes: les relations R de T, les uplets $\underline{u} : A \to \mathrm{dom}$
  ($A=\{a_1,\ldots,a_n\}$ ensemble fini d'attributs; on notera $\underline{u}$ $\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle$)
  — + 6 opérations:
  sélection σ_F (F conj. d'égalités $a=b$ ou $a=$cst, a et b **attributs**;
  Alice autorise aussi les $\neq$)
  projection π_A (A ensemble **d'attributs** — et **ensemble**, pas liste)
  jointure naturelle $\bowtie$
  renommage ϱ_r (r fct. injective partielle de Attr dans Attr)
  différence –
  union $\cup$
- De nouveau, restreintes par typage… [Alice p.50, sans – ni $\cup$]

L'algèbre relationnelle RA (sans noms): sortes

- Typage $q : B$, où B est un ensemble d'attributs (la **sorte**)
- R : schema(R)
  $\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle : \{a_1,\ldots,a_n\}$
  $\sigma_F : B \to B$ (F conj. d'égalités $a{=}b$ ou $a{=}$cst., **avec** $\boldsymbol{a,b \in B}$)
  $\pi_A : B \to A$ ($A \subseteq B$)
  $\bowtie : B \times C \to B \cup C$
  ϱ_r : dom $r \to$ im r
  $-, \cup : B \times B \to B$ (avec **le même** $\boldsymbol{B}$ à gauche de la flèche)

L'algèbre relationnelle RA (avec noms): sémantique

- Sémantique $\mathbf{I}[\![q]\!] \in \mathbf{P}_{\text{fin}}(\text{schema}(q) \to \text{dom})$

  $\mathbf{I}[\![\mathrm{R}]\!] = \mathbf{I}(\mathrm{R})$

  $\mathbf{I}[\![\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle]\!] = \{\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle\}$

  $\mathbf{I}[\![\sigma_F(q)]\!] = \{\underline{x} \in \mathbf{I}[\![q]\!] \mid \bigwedge_{a=b\in F} \underline{x}(a)=\underline{x}(b) \wedge \bigwedge_{a=\text{cst}\in F} \underline{x}(a)=\text{cst}\}$

  $\mathbf{I}[\![\pi_A(q)]\!] = \{\underline{x}_{\mid A} \mid \underline{x} \in \mathbf{I}[\![q]\!]\}$ (ens. des **restrictions** à A)

  $\mathbf{I}[\![q \bowtie q']\!] = \{\underline{x} : \text{schema}(q) \cup \text{schema}(q') \to \text{dom}$

  $\mid \underline{x}_{\mid \text{schema}(q)} \in \mathbf{I}[\![q]\!]$ et $\underline{x}_{\mid \text{schema}(q')} \in \mathbf{I}[\![q']\!]\}$

  $\mathbf{I}[\![\varrho_r(q)]\!] = \{\underline{x} \circ r^{-1} \mid \underline{x} \in \mathbf{I}[\![q]\!]\}$

  $\mathbf{I}[\![q - q']\!] = \mathbf{I}[\![q]\!] - \mathbf{I}[\![q']\!]$

  $\mathbf{I}[\![q \cup q']\!] = \mathbf{I}[\![q]\!] \cup \mathbf{I}[\![q']\!]$

Exemple

- $\mathbf{I} =$

Movies	Title	Director	Actor
	The Trouble with Harry	Hitchcock	Gwenn
	The Trouble with Harry	Hitchcock	Forsythe
	The Trouble with Harry	Hitchcock	MacLaine
	The Trouble with Harry	Hitchcock	Hitchcock
	...	...	...
	Cries and Whispers	Bergman	Andersson
	Cries and Whispers	Bergman	Sylwan
	Cries and Whispers	Bergman	Thulin
	Cries and Whispers	Bergman	Ullman

Location	Theater	Address	Phone Number
	Gaumont Opéra	31 bd. des Italiens	47 42 60 33
	Saint André des Arts	30 rue Saint André des Arts	43 26 48 18
	Le Champo	51 rue des Ecoles	43 54 51 60
	...	...	...
	Georges V	144 av. des Champs-Elysées	45 62 41 46
	Les 7 Montparnassiens	98 bd. du Montparnasse	43 20 32 20

Pariscope	Theater	Title	Schedule
	Gaumont Opéra	Cries and Whispers	20:30
	Saint André des Arts	The Trouble with Harry	20:15
	Georges V	Cries and Whispers	22:15
	...	...	...
	Les 7 Montparnassiens	Cries and Whispers	20:45

Figure 3.1: The **CINEMA** database

- Movies $\bowtie$ Pariscope :
  {Theater, Title, Director, Actor, Schedule}

- sémantique:
  (« G.Opéra », « C&W », « Bergman », « Andersson », « 20:30 »),
  (« G.Opéra », « C&W », « Bergman », « Sylwan », « 20:30 »),
  « Georges V », « C&W », « Bergman », « Andersson », « 22:15 »),
  …

RA

- Les deux versions de RA ont un pouvoir expressif équivalent
- On verra ça comme conséquence du théorème de Codd

Traduction de RA en FOL

RA → FOL

- **Prop [Alice, Lemme 5.3.11 p.80].** Pour toute requête q de RA (sans noms), on peut calculer une requête (de FOL) $\{\underline{x} \mid F[q]\}$ telle que (en la notant q')
  $\mathbf{I}[\![q]\!] = q'(\text{dom},\mathbf{I})$ pour toute instance $\mathbf{I}$ sur dom,
  pour tout domaine dom contenant adom(q)

- On se fixe une énumération de variables $x_1, x_2, \ldots$ toutes distinctes

- $F[R] = R(x_1,\ldots,x_n)$ où $R{:}n$
  $F[(u_1, \ldots, u_n)] = \bigwedge_{i=1}^{n} x_i{=}u_i$
  $F[\sigma_F(q)] = F[q] \wedge \bigwedge_{i=j\in F} x_i{=}x_j \wedge \bigwedge_{i=\text{cst}\in F} x_i{=}\text{cst}$
  $F[\pi_L(q)] = (\exists x_{j[1]},\ldots,x_{j[n-k]} \,.\, F[q])\,[x_{i[1]}{:=}x_1, \ldots, x_{i[k]}{:=}x_k]$
  où $L{=}[i[1],\ldots,i[k]]$, et $j[1],\ldots,j[n{-}k]$ énumère les indices de $\{1, \ldots, n\}$ hors de L
  $F[q \times q'] = F[q] \wedge (F[q'][x_1{:=}x_{m+1},\ldots,x_n{:=}x_{m+n}])$ où $q{:}m$, $q'{:}n$
  $F[q - q'] = F[q] \wedge \neg F[q']$ $\qquad$ $F[q \cup q'] = F[q] \vee F[q']$

Indépendance de domaine (1/5)

- Le **domaine actif** d'une instance **I** est
  adom(**I**) = {valeurs apparaissant dans au moins un uplet de **I**}
- On a adom(**I**) ⊆ dom par définition de **I**
- Alice note adom(q,**I**) = adom(q) ∪ adom(**I**) et demande juste que dom contienne adom(q,**I**)
- Pour toute requête RA q,
  I⟦q⟧ **ne dépend pas** de dom
  ⊇ adom(q,**I**)

$F[R] = R(x_1,\ldots,x_n)$ où $R{:}n$

$F[(u_1, \ldots, u_n)] = \wedge_{i=1}^{n}\, x_i{=}u_i$

$F[\sigma_F(q)] = F[q] \wedge \wedge_{i=j\in F}\, x_i{=}x_j \wedge \wedge_{i=\mathrm{cst}\in F}\, x_i{=}\mathrm{cst}$

$F[\pi_L(q)] = (\exists \mathrm{x}_{j[1]},\ldots,\mathrm{x}_{j[n-k]} \,.\, F[q])\, [x_{i[1]}{:=}x_1, \ldots, x_{i[k]}{:=}x_k]$

où $L{=}[i[1],\ldots,i[k]]$, et $j[1],\ldots,j[n{-}k]$ énumère les indices de $\{1, \ldots, n\}$ hors de L

$F[q \times q'] = F[q] \wedge (F[q'][x_1{:=}x_{m+1},\ldots,x_n{:=}x_{m+n}])$ où $q{:}m$, $q'{:}n$

$F[q - q'] = F[q] \wedge \neg F[q']$ $\qquad$ $F[q \cup q'] = F[q] \vee F[q']$

Indépendance de domaine (2/5)

- En revanche, pour une requête (FOL) q'
  $q'(\mathrm{dom},\mathbf{I})$ peut dépendre de dom, au sens où
  il peut exister deux ensembles finis dom, dom′ $\supseteq$ adom(q',**I**) tels que $q'(\mathrm{dom},\mathbf{I}) \neq q'(\mathrm{dom}',\mathbf{I})$
- $\{x \mid \neg R(x)\}$
  $\{(x,y) \mid R(x) \vee R(y)\}$
  $\{x \mid R(x) \vee \neg R(x)\}$
  $\{x \mid \forall y \,.\, x{=}y\}$
  $\{x \mid \exists y \,.\, (R(x) \vee \neg R(x)) \wedge S(y)\}$

- $\mathbf{I} \vDash_{\mathrm{dom}} R(\underline{u})[\nu]$ ssi $\underline{u}\nu$ est dans $\mathbf{I}(R)$
- $\mathbf{I} \vDash_{\mathrm{dom}} (s{=}t)[\nu]$ ssi $s\nu{=}t\nu$, $\mathbf{I} \vDash_{\mathrm{dom}} \bot[\nu]$ jamais, $\mathbf{I} \vDash_{\mathrm{dom}} \top[\nu]$ toujours
- $\mathbf{I} \vDash_{\mathrm{dom}} (F \wedge G)[\nu]$ ssi $\mathbf{I} \vDash_{\mathrm{dom}} F[\nu_{\mid \mathrm{free}(F)}]$ et $\mathbf{I} \vDash_{\mathrm{dom}} G[\nu_{\mid \mathrm{free}(G)}]$
- $\mathbf{I} \vDash_{\mathrm{dom}} (F \vee G)[\nu]$ ssi $\mathbf{I} \vDash_{\mathrm{dom}} F[\nu_{\mid \mathrm{free}(F)}]$ ou $\mathbf{I} \vDash_{\mathrm{dom}} G[\nu_{\mid \mathrm{free}(G)}]$
- $\mathbf{I} \vDash_{\mathrm{dom}} (\neg F)[\nu]$ ssi non ($\mathbf{I} \vDash_{\mathrm{dom}} F[\nu]$)
- $\mathbf{I} \vDash_{\mathrm{dom}} (\exists x \,.\, F)[\nu]$ ssi il existe a **dans dom** tel que
  $\mathbf{I} \vDash_{\mathrm{dom}} F[\nu[x{:=}a]]$

Indépendance de domaine (3/5)

- On dit qu'une requête FOL q' est **indépendante du domaine** ssi pour toute instance **I**,pour tous
  dom, dom' $\supseteq$ adom(q',**I**), on a q'(dom,**I**) = q'(dom',**I**)
- La requête FOL $\{\underline{x} \mid F[q]\}$ construite à partir d'une requête RA q **est indépendante du domaine**
- … car **I**$[\![q]\!] = \{\underline{x} \mid F[q]\}$ (dom,**I**)
  et **I**$[\![q]\!]$ ne dépend pas de dom

Indépendance de domaine (4/5)

- Dans le Alice, ils étendent ça au cas où dom peut être **infini** (ce que j'ai exclu depuis le début)
- En général, une requête FOL q non indépendante du domaine a la particularité que $q(\mathrm{dom}, \mathbf{I})$ (où dom est infini) est lui-même **infini**
- (Je pense que c'est toujours vrai. Probablement parce que toute formule FOL qui a des modèles finis de cardinalité arbitrairement grande a un modèle infini.)

Indépendance de domaine (5/5)

- Le problème:
  ENTREE: une requête FOL $q = \{\underline{x} \mid F\}$
  Q: q est-elle indépendante du domaine?
- est indécidable (si T contient au moins une relation binaire)
- C'est une conséquence du **théorème de Trakhtenbrot**: l'existence d'un modèle fini pour une formule FOL close F est indécidable (et r.e.; on verra ça plus tard)
- *Preuve:* F a un modèle fini ssi $\{x \mid \neg \mathrm{R}(x) \wedge \mathrm{F}\}$ n'est pas indépendante du domaine

Critère d'indépendance du domaine

Critère d'indépendance de domaine

- Toute requête (FOL) **conjonctive**, c'est-à-dire construite à l'aide de $\exists$ et $\wedge$ seulement (pas $\top$, $\vee$, $\neg$, $\forall$) est **automatiquement indépendante du domaine** (exercice — ou appliquer le critère suivant)
- On va définir un critère:
  — s'il réussit sur la requête FOL q,
  alors q est indépendante du domaine
  — s'il échoue, on ne sait pas
  — calculable en temps polynomial
  — il réussit sur les requêtes conjonctives,
  et sur les traductions de requêtes RA
- La définition du [Alice, Section 5.4] n'est **pas** totalement rigoureuse

Critère d'indépendance de domaine

- On définit un ensemble $\mathrm{rr}(F) \subseteq \mathrm{free}(F)$ de variables (« **range restricted** » — je dirai les variables **contraintes**)
  — calculable en temps polynomial
  — tel que:
  pour toute instance $\mathbf{I}$,
  pour tout $\mathrm{dom} \supseteq \mathrm{adom}(F)$ ($\cup\, \mathrm{adom}(\mathbf{I})$),
  pour toute valuation ν (rel. dom pour F),
  si $\mathbf{I} \models_{\mathrm{dom}} F[\nu]$ alors pour toute $x \in \mathrm{rr}(F)$,
  $\nu(x) \in \mathrm{adom}(F) \cup \mathrm{adom}(\mathbf{I})$
- [Alice, Algorithme 5.4.3 p.84, à quelques adaptations près]

Tests d'indépendance de domaine

C'est le cas important

- $\mathrm{rr}(\mathrm{R}(\underline{u})) = \mathrm{free}(\mathrm{R}(\underline{u}))$
  $\mathrm{rr}(x{=}\mathrm{cst}) = \mathrm{rr}(\mathrm{cst}{=}x) = \{x\}$
  $\mathrm{rr}(\mathrm{cst}{=}\mathrm{cst}') = \varnothing \quad \mathrm{rr}(x{=}y) = \varnothing$
- $\mathrm{rr}(F \wedge x{=}y) = \mathrm{rr}(F)$ si $x,y \notin \mathrm{rr}(F)$
  $= \mathrm{rr}(F) \cup \{x,y\}$ si x ou y est dans $\mathrm{rr}(F)$
  $\mathrm{rr}(F \wedge G) = \mathrm{rr}(F) \cup \mathrm{rr}(G)$ si G pas une équation entre variables
- $\mathrm{rr}(\neg F) = \varnothing$ (on ne se fatigue pas)
- $\mathrm{rr}(\exists x \,.\, F) = \mathrm{rr}(F) - \{x\}$
  (défini comme $\perp$ [ou **fail**] si $x \notin \mathrm{rr}(F)$ dans le Alice)
- $\mathrm{rr}(F \vee G) = \mathrm{rr}(F) \cap \mathrm{rr}(G)$

Range restriction

- On dit que $\{\underline{x} \mid F\}$ est **contrainte** (**range-restricted**) ssi:
  — $\mathrm{rr}(F) = \mathrm{free}(F)$
  — dans toute sous-formule $\exists x \,.\, G$, x est dans $\mathrm{rr}(G)$

  (ceci est automatique si on utilise un symbole $\perp$ (ou **fail**) comme dans le Alice)

- **Note:** pour tout requête RA $q{:}n$,
  $\mathrm{rr}(F[q])=\mathrm{free}(F[q])=\{x_1,\ldots,x_n\}$

  Corollaire: $\{\underline{x} \mid F[q]\}$
  est **contrainte**.

Si on est formel, il faut associer les $\wedge$ **à droite**

$F[R] = R(x_1,\ldots,x_n)$ où $R{:}n$
$F[(u_1, \ldots, u_n)] = \wedge_{i=1}^{n} x_i{=}u_i$
$F[\sigma_F(q)] = F[q] \wedge \wedge_{i=j\in F}\, x_i{=}x_j \wedge \wedge_{i=\mathrm{cst}\in F}\, x_i{=}\mathrm{cst}$
$F[\pi_L(q)] = (\exists x_{j[1]},\ldots,x_{j[n-k]} \,.\, F[q])\, [x_{i[1]}{:=}x_1, \ldots, x_{i[k]}{:=}x_k]$
où $L=[i[1],\ldots,i[k]]$, et $j[1],\ldots,j[n-k]$ énumère
les indices de $\{1, \ldots, n\}$ hors de L
$F[q \times q'] = F[q] \wedge (F[q'][x_1{:=}x_{m+1},\ldots,x_n{:=}x_{m+n}])$ où $q{:}m$, $q'{:}n$
$F[q - q'] = F[q] \wedge \neg F[q']$ $F[q \cup q'] = F[q] \vee F[q']$

Range restriction

- Note: la plupart des descriptions commencent par opérer des simplifications préservant la sémantique sur les formules (aplatir les $\wedge$ et les $\vee$, pousser les négations vers l'intérieur, éliminer les doubles négations, etc.)
- Ceci permet d'améliorer la précision de l'approximation calculée par rr(), mais (me semble-t-il) **ne sert à rien** pour les résultats théoriques visés

La sémantique de domaine actif

Sémantique de domaine actif

- Soit q une requête FOL.
  Au lieu de restreindre q **syntaxiquement**,
  on peut utiliser une **sémantique** restreinte

- Sémantique **de domaine actif**:
  $$q_{\mathrm{act}}(\mathbf{I}) = q(\mathrm{adom}(q,\mathbf{I}),\mathbf{I})$$
  … autrement dit, on demande à évaluer q sur le domaine actif $\mathrm{adom}(q,\mathbf{I})$ plutôt qu'un domaine arbitraire dom

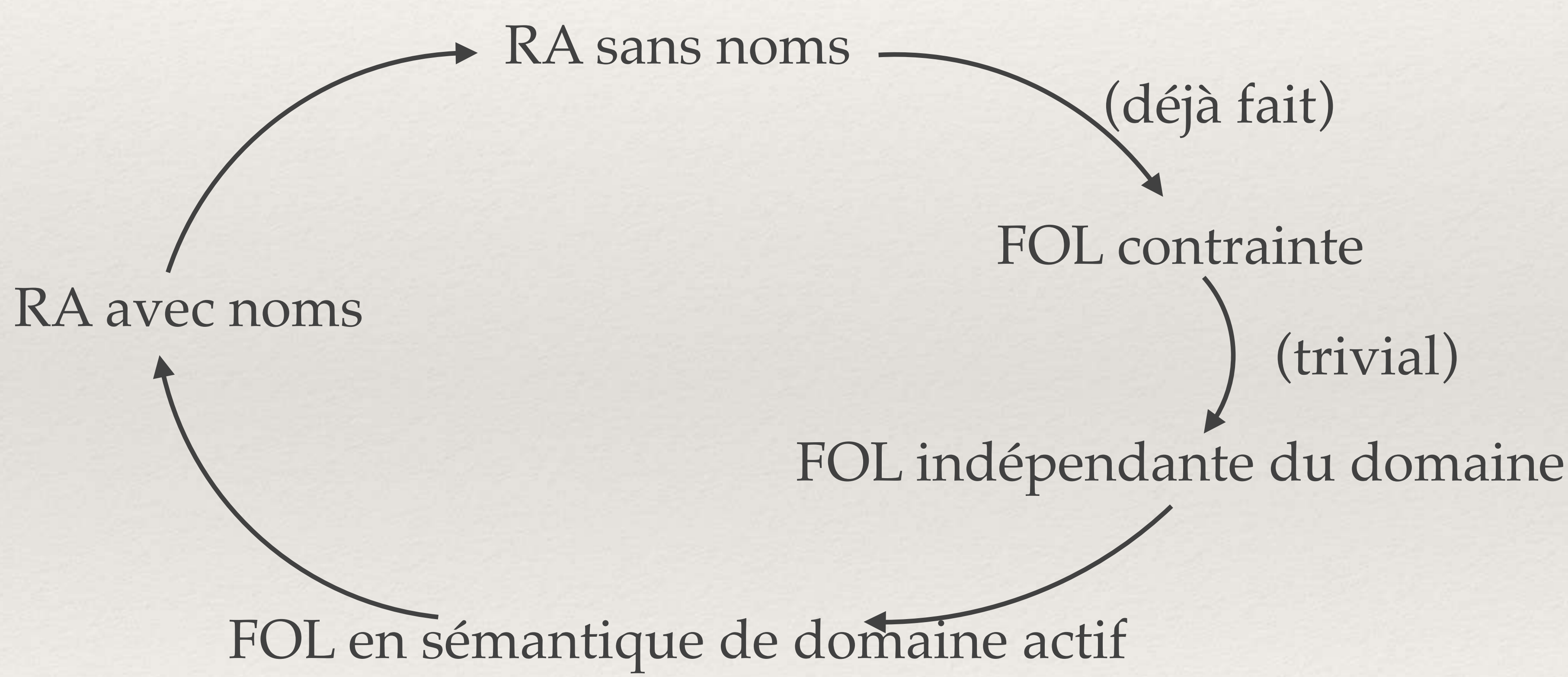
- **Note:** si q est contrainte (donc indépendante du domaine), alors $q(\mathbf{I}) = q(\mathrm{dom},\mathbf{I})$ pour tout $\mathrm{dom} \supseteq \mathrm{adom}(q,\mathbf{I})$

Le théorème de Codd

- **Théorème de Codd (1972):** « Les langages de requêtes suivants ont le même pouvoir expressif:
— RA avec noms
— RA sans noms
— les requêtes FOL contraintes
— les requêtes FOL indépendantes du domaine
— les requêtes FOL en sémantique de domaine actif »

Le théorème de Codd

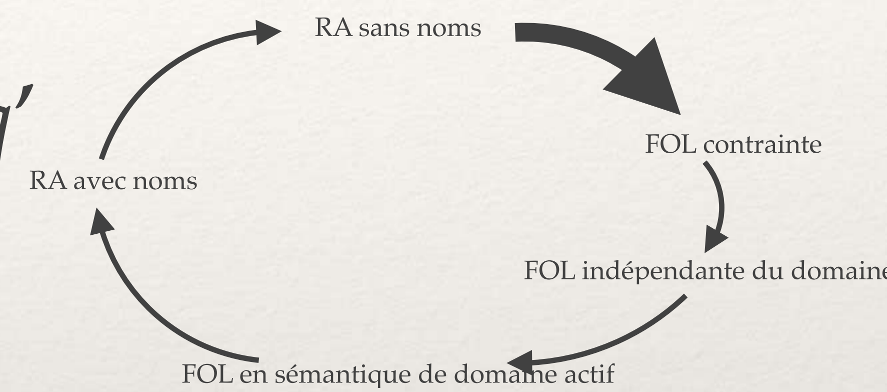
- On va exhiber des traductions:



- …de sorte que les sémantiques correspondent

RA sans noms → FOL contrainte

- Rappel: q requête RA sans noms
  → requête FOL contrainte q'



- $q' = \{\underline{x} \mid \mathrm{F}[q]\}$

- $\mathbf{I}[\![q]\!] = q'(\mathrm{dom},\mathbf{I})$ pour toute instance $\mathbf{I}$ sur dom,
  pour tout domaine $\mathrm{dom} \supseteq \mathrm{adom}(q,\mathbf{I})$

- De plus, on a remarqué que $\mathrm{F}[q]$ est contrainte

$\mathrm{F}[\mathrm{R}] = \mathrm{R}(x_1,\ldots,x_n)$ où $\mathrm{R}{:}n$

$\mathrm{F}[(u_1, \ldots, u_n)] = \wedge_{i=1}^{n}\, x_i{=}u_i$

$\mathrm{F}[\sigma_F(q)] = \mathrm{F}[q] \wedge \wedge_{i=j\in F}\, x_i{=}x_j \wedge \wedge_{i=\mathrm{cst}\in F}\, x_i{=}\mathrm{cst}$

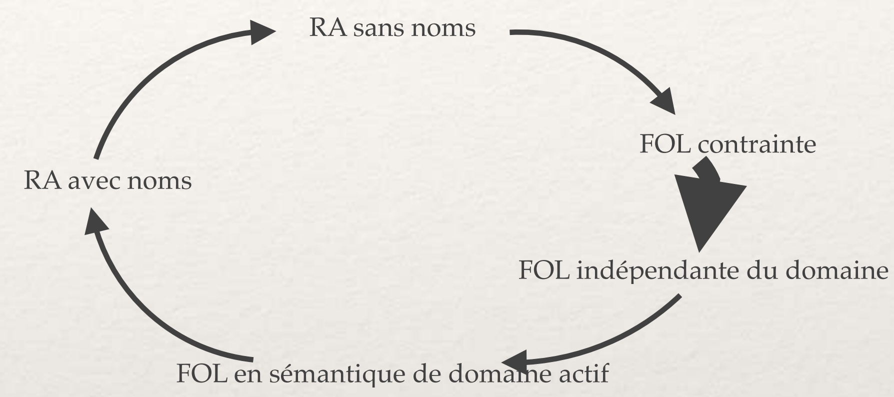
$\mathrm{F}[\pi_L(q)] = (\exists \mathrm{x}_{j[1]},\ldots,\mathrm{x}_{j[n-k]} \,.\, \mathrm{F}[q])\; [x_{i[1]}{:=}x_1, \ldots, x_{i[k]}{:=}x_k]$
où $L{=}[i[1],\ldots,i[k]]$, et $j[1],\ldots,j[n{-}k]$ énumère les indices de $\{1, \ldots, n\}$ hors de L

$\mathrm{F}[q \times q'] = \mathrm{F}[q] \wedge (\mathrm{F}[q'][x_1{:=}x_{m+1},\ldots,x_n{:=}x_{m+n}])$ où $q{:}m$, $q'{:}n$

$\mathrm{F}[q - q'] = \mathrm{F}[q] \wedge \neg\mathrm{F}[q']$ $\qquad$ $\mathrm{F}[q \cup q'] = \mathrm{F}[q] \vee \mathrm{F}[q']$

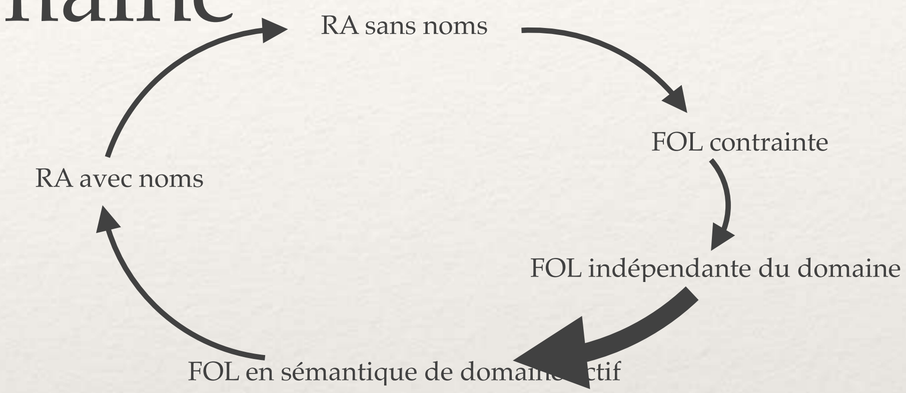
FOL contrainte → indép. du domaine

- q requête FOL contrainte
  → requête FOL
  indépendante du domaine
- … q elle-même, bien sûr



FOL indép. du domaine → domaine actif

- q requête FOL indépendante du domaine → requête FOL en sémantique de domaine actif



- … q elle-même:
- pour tout dom $\supseteq$ adom(q,**I**),

  $$q_{\text{act}}(\mathbf{I})\ [= q(\text{adom}(q,\mathbf{I}),\mathbf{I})] = q(\text{dom},\mathbf{I})$$

FOL domaine actif → RA avec noms

- q requête FOL en sémantique de domaine actif
→ requête RA avec noms

- C'est présenté de façon compliquée dans [Alice, section 5.4], avec des notions de SRNF, de RANF, etc.

- Deux idées:
— **définissabilité du domaine actif** en RA
— on utilise simplement les **variables** comme **attributs**

Le domaine actif est définissable en RA

- Soit b un attribut quelconque, et q **une requête FOL fixe**
- Pour chaque relation R dans T, de sorte A **non vide**,

(pour éviter une union vide plus bas…)

$$\text{adom\_R\_}a(b) = \varrho_{a \to b}\, \pi_{\{a\}}\, \mathrm{R} \quad (\text{pour chaque } a \in A)$$

$$\text{adom\_R}(b) = \textstyle\bigcup_{a \in A} \text{adom\_R\_}a(b) \qquad (\text{… ici!})$$

- Puis:

$$\text{adom}(b) = \textstyle\bigcup_{\mathrm{R} \in \mathrm{T} \text{ de sorte non vide}} \text{adom\_R}(b) \cup \underbrace{\textstyle\bigcup_{u \in \text{adom}(q)} \langle b{:}u \rangle}_{\text{uniquement si adom}(q) \text{ non vide}}$$

- **Propriété:** la sorte de adom(b) est $\{b\}$,

  et $\mathbf{I}[\![\text{adom}(b)]\!] = \{\langle b{:}u \rangle \mid u \in \text{adom}(q,\mathbf{I})\}$

Le domaine actif est définissable

- Il est aussi définissable en FOL, bien sûr, et c'est une des choses que fait Alice, mais on ne va pas passer par la RANF du Alice

- Pour chaque relation R dans T, d'arité $n \geq 1$, on pose

$$\mathrm{adom\_R}_i(x) = \exists x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \,. \qquad (1 \leq i \leq n)$$
$$\mathrm{R}(x_1,\ldots,x_{i-1}, x, x_{i+1},\ldots,x_n)$$
$$\mathrm{adom\_R}(x) = \mathrm{adom\_R}_1(x) \vee \ldots \vee \mathrm{adom\_R}_n(x)$$

- Puis:

$$\mathrm{adom}(x) = \vee_{\mathrm{R} \in \mathrm{T}}\, \mathrm{adom\_R}(x) \vee \vee_{a \in \mathrm{adom}(q)}\, x{=}a$$

- **Propriété:** $\mathbf{I} \vDash_{\mathrm{dom}} \mathrm{adom}(x)[\nu]$ ssi $\nu(x) \in \mathrm{adom}(q,\mathbf{I})$

Cylindrification

- (C'est le nom utilisé par Tarski dans les années 40…)
- Si $q : A$ et $A \subseteq B$,
  $$\mathrm{cyl}_B(q) = q \times \Pi_{b \in B-A}\ \mathrm{adom}(b)$$
- **Propriété:** $\mathrm{cyl}_B(q) : B$ et
  $$\mathbf{I}[\![\mathrm{cyl}_B(q)]\!] = \{\text{uplets} : B \to \mathrm{adom}(q,\mathbf{I}) \text{ dont la projection sur } A \text{ est dans } \mathbf{I}[\![q]\!]\}$$

La traduction (1/4)

- Comme Attr est **infini dénombrable**, à bijection près on peut supposer que **toute variable est un attribut**
- On peut alors identifier toute **valuation** ν sur un ensemble fini de variables B à un **uplet** de sorte B
- On traduit chaque formule F en une requête RA avec noms q[F], telle que:
  — q[F] : free(F)
  — $\mathbf{I}[\![\mathrm{q}[F]]\!] = \{\nu \text{ valuation} : \mathrm{free}(F) \rightarrow \mathrm{adom}(q,\mathbf{I})$
  $\mid \mathbf{I} \vDash_{\mathrm{adom}(q,\mathbf{I})} F[\nu]\}$

La traduction (2/4)

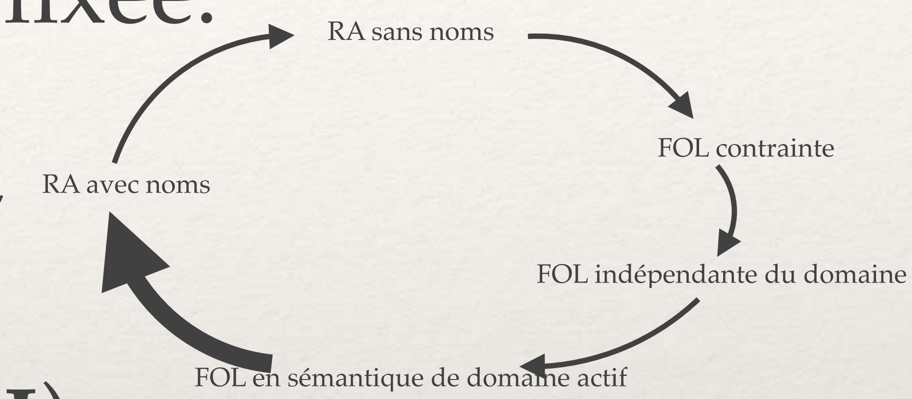
- $q[x{=}\mathrm{cst}] = \langle x{:}\mathrm{cst}\rangle$ $q[\mathrm{cst}{=}x] = \langle x{:}\mathrm{cst}\rangle$
  $q[\mathrm{cst}{=}\mathrm{cst}'] = \langle\rangle - \langle\rangle$ si $\mathrm{cst} \neq \mathrm{cst}'$, $q[\mathrm{cst}{=}\mathrm{cst}] = \langle\rangle$
  $q[x{=}y] = \sigma_{x=y}(\mathrm{adom}(x) \times \mathrm{adom}(y))$ si $x \neq y$, $q[x{=}x] = \mathrm{adom}(x)$
- $q[\top] = \langle\rangle$ $q[\bot] = \langle\rangle - \langle\rangle$
- $q[R(\underline{u})]$… plus tard, c'est un gros morceau
- $q[F \wedge G] = q[F] \bowtie q[G]$
- $q[F \vee G] = \mathrm{cyl}_{\mathrm{free}(G) - \mathrm{free}(F)}(q[F]) \cup \mathrm{cyl}_{\mathrm{free}(F) - \mathrm{free}(G)}(q[G])$
- $q[\exists x \,.\, F] = \pi_{\mathrm{free}(\exists x \,.\, F)}(q[F])$
- $q[\neg F] = \Pi_{x \in \mathrm{free}(F)}\, \mathrm{adom}(x) - q[F]$

La traduction (3/4)

- q[R($\underline{u}$)] = $\varrho_f(\pi_A(\sigma_F(R)))$
  où: schema(R) = $\{a_1 < \ldots < a_n\}$ (rappel: Attr totalement ordonné)
  $\underline{u} = (u_1, \ldots, u_n)$
  $I_{\text{cst}} = \{i \mid 1 \leq i \leq n,\ u_i$ est une constante$\}$
  $I_{\text{var}} = \{i \mid 1 \leq i \leq n,\ u_i$ est une variable$\}$
  pour chaque $i \in I_{\text{var}}$, $\text{fst}(i) = \min \{j \in I_{\text{var}} \mid u_i = u_j\}$
  $F = \bigwedge\{a_i = u_i \mid i \in I_{\text{cst}}\} \wedge \bigwedge\{a_i = a_{\text{fst}(i)} \mid i \in I_{\text{var}},\ i \neq \text{fst}(i)\}$
  $A = \{a_i \mid i \in \text{Im fst}\}$
  $f : a_i \mapsto u_i$, pour chaque $i \in$ Im fst (injectif, d'accord?)
- Exemple: si R : $\{a_1 < a_2 < a_3 < a_4 < a_5 < a_6\}$,
  q[R(« a », x, « b », y, y, x)] = $\varrho_f(\pi_A(\sigma_F(R)))$
  où $F = a_1 =$ « a » $\wedge\ a_3 =$ « b » $\wedge\ a_5 = a_4 \wedge a_6 = a_2$,
  $A = \{a_2, a_4\}$, et $f : a_2 \mapsto x,\ a_4 \mapsto y$

La traduction (4/4)

- On rappelle que la requête FOL q est fixée.

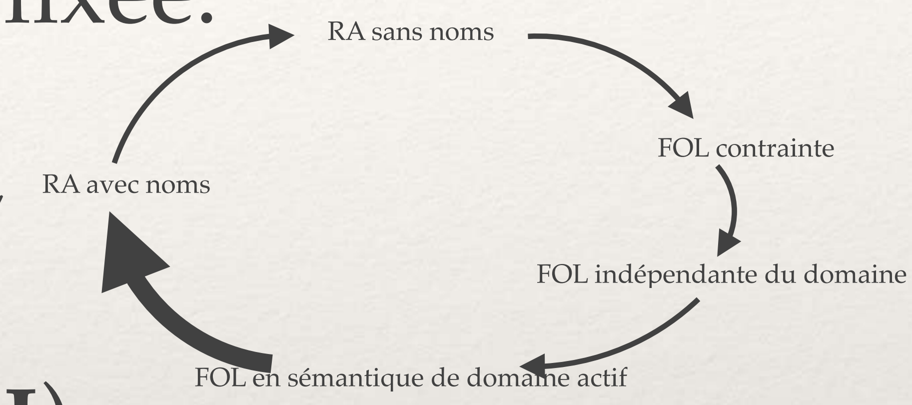


- **Propriété:** pour toute formule FOL F,
  — $q[F]$: $\mathrm{free}(F)$
  — $\mathbf{I}[\![q[F]]\!] = \{\nu \text{ val} : \mathrm{free}(F) \to \mathrm{adom}(q,\mathbf{I})$
  $\qquad\qquad \mid \mathbf{I} \vDash_{\mathrm{adom}(q,\mathbf{I})} F[\nu]\}$

- *Preuve:* récurrence sur F.

La traduction (4/4)

- On rappelle que la requête FOL q est fixée.



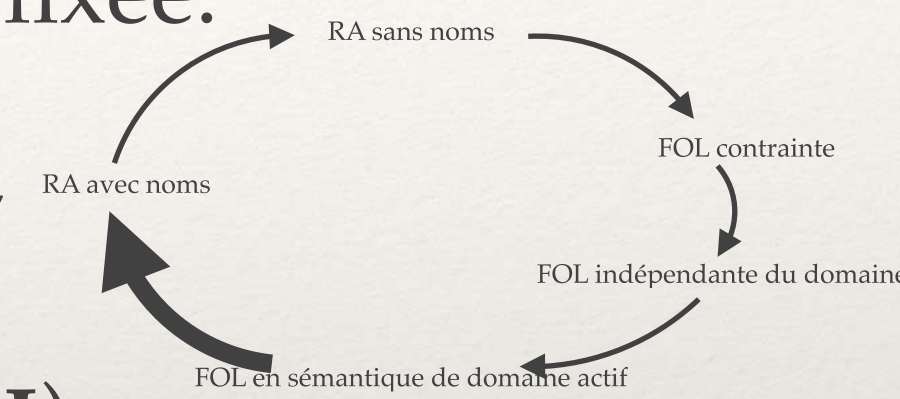
- **Propriété:** pour toute formule FOL F,
  — $\mathrm{q}[F]$: $\mathrm{free}(F)$
  — $\mathbf{I}[\![\mathrm{q}[F]]\!] = \{\nu \text{ val} : \mathrm{free}(F) \rightarrow \mathrm{adom}(q,\mathbf{I}) \mid \mathbf{I} \vDash_{\mathrm{adom}(q,\mathbf{I})} F[\nu]\}$

- *Preuve:* récurrence sur F.

- Maintenant soit $q = \{\underline{x} \mid F\}$. On la traduit en $\mathrm{q}[F]$:

La traduction (4/4)

- On rappelle que la requête FOL q est fixée.



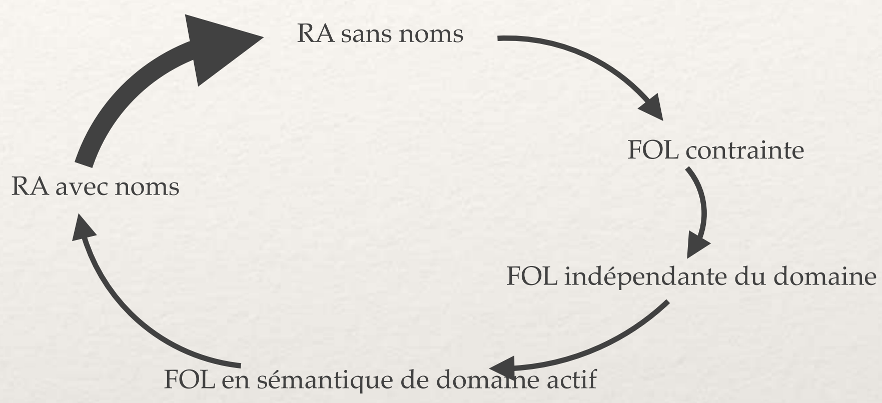
- **Propriété:** pour toute formule FOL F,
  — $\mathrm{q}[F]$: $\mathrm{free}(F)$
  — $\mathbf{I}[\![\mathrm{q}[F]]\!] = \{\nu \text{ val} : \mathrm{free}(F) \to \mathrm{adom}(q,\mathbf{I}) \mid \mathbf{I} \vDash_{\mathrm{adom}(q,\mathbf{I})} F[\nu]\}$

- *Preuve:* récurrence sur F.

- Maintenant soit $q = \{\underline{x} \mid F\}$. On la traduit en $\mathrm{q}[F]$:

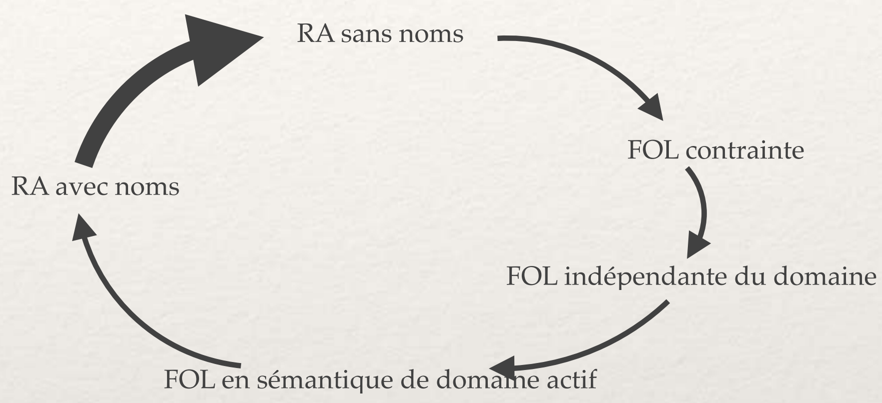
- **Propriété:** $\mathbf{I}[\![\mathrm{q}[F]]\!] = \{\nu \text{ val} : \underline{x} \to \mathrm{adom}(q,\mathbf{I}) \mid \mathbf{I} \vDash_{\mathrm{adom}(q,\mathbf{I})} F[\nu]\}$
  $= q(\mathrm{adom}(q,\mathbf{I}),\mathbf{I}) = q_{\mathrm{act}}(\mathbf{I})$

Boucler la boucle (1/2)



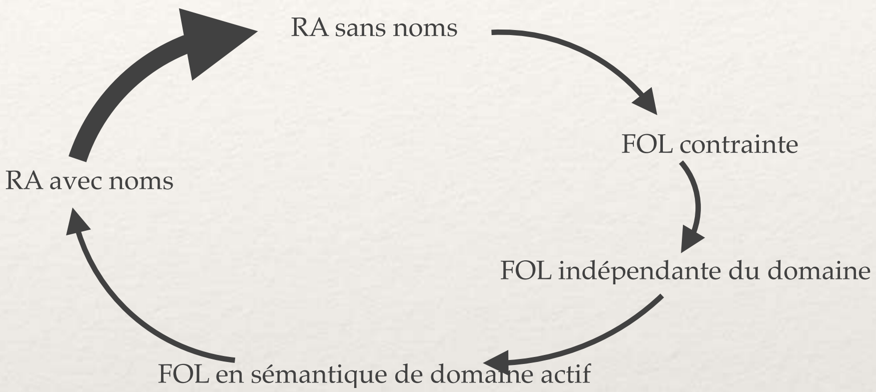
Boucler la boucle (1/2)

- q requête RA avec noms
  → requête RA sans noms q^o



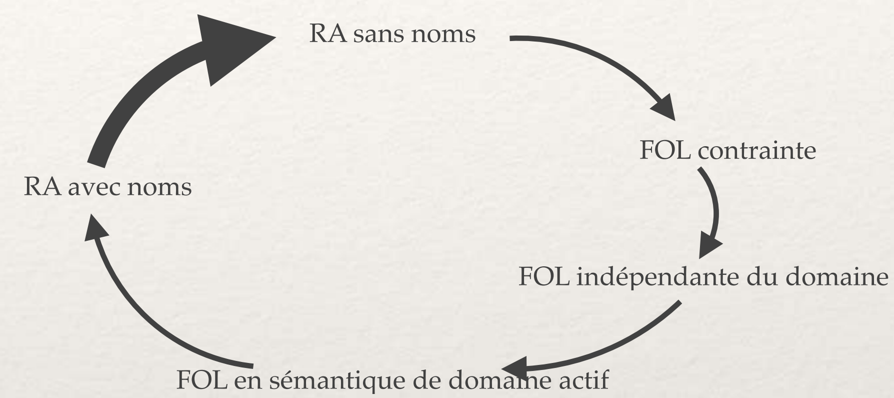
Boucler la boucle (1/2)

- q requête RA avec noms
  → requête RA sans noms q^o



- On va la calculer de sorte que:
  — si $q : A = \{a_1 < \ldots < a_n\}$, alors $q^o : n$
  — $\mathbf{I}[\![q^o]\!] = \{(u_1, \ldots, u_n) \mid \langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle \in \mathbf{I}[\![q]\!]\}$
  (= $\mathbf{I}[\![q]\!]$, mod l'identification de $(u_1, \ldots, u_n)$ avec $\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle$)

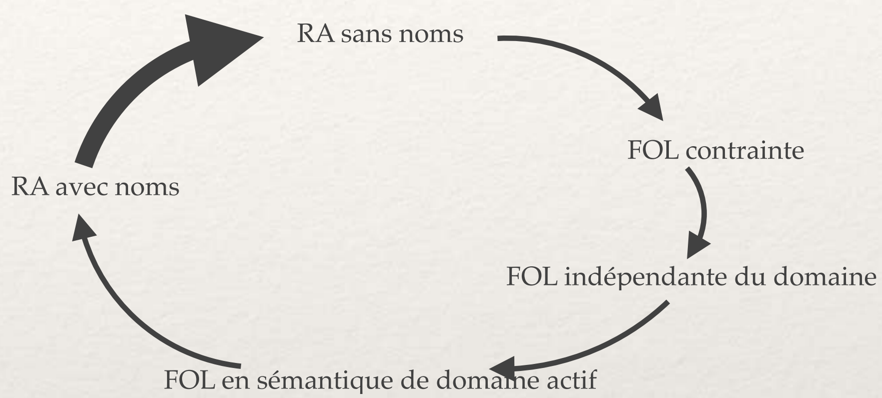
Boucler la boucle (1/2)



- q requête RA avec noms
  $\to$ requête RA sans noms q^o
- On va la calculer de sorte que:
  — si $q : A = \{a_1 < \ldots < a_n\}$, alors $q^o : n$
  — $\mathbf{I}[\![q^o]\!] = \{(u_1, \ldots, u_n) \mid \langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle \in \mathbf{I}[\![q]\!]\}$
  (= $\mathbf{I}[\![q]\!]$, mod l'identification de $(u_1, \ldots, u_n)$ avec $\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle$)
- $R^o = R$ $\quad\quad \langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle^o = (u_1, \ldots, u_n)$
  $(\sigma_F(q))^o = \sigma_{F^o}(q^o)$,
  où F^o est la conjonction des i=j (pour chaque conjoint a_i=a_j de F)
  et des i=cst (pour chaque conjoint a_i=cst de F)
  $(\pi_{\{a_1,\ldots,a_n\}}(q))^o = \pi_{\{f[1], \ldots, f[m]\}}(q^o)$ où $q : \{a_1 < \ldots < a_n\}$ et $A = \{a_{f[1]} < \ldots < a_{f[m]}\}$
  $(q - q')^o = q^o - q'^o \quad\quad (q \cup q')^o = q^o \cup q'^o$

Boucler la boucle (1/2)

- q requête RA avec noms
  $\to$ requête RA sans noms q^o



- On va la calculer de sorte que:
  — si $q : A = \{a_1 < \ldots < a_n\}$, alors $q^o : n$
  — $\mathbf{I}[\![q^o]\!] = \{(u_1, \ldots, u_n) \mid \langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle \in \mathbf{I}[\![q]\!]\}$
  (= $\mathbf{I}[\![q]\!]$, mod l'identification de $(u_1, \ldots, u_n)$ avec $\langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle$)

- $R^o = R$ $\quad \langle a_1{:}u_1, \ldots, a_n{:}u_n\rangle^o = (u_1, \ldots, u_n)$
  $(\sigma_F(q))^o = \sigma_{F^o}(q^o)$,
  où F^o est la conjonction des $i{=}j$ (pour chaque conjoint $a_i{=}a_j$ de F)
  et des $i{=}$cst (pour chaque conjoint $a_i{=}$cst de F)
  $(\pi_{\{a_1,\ldots,a_n\}}(q))^o = \pi_{\{f[1], \ldots, f[m]\}}(q^o)$ où $q : \{a_1 < \ldots < a_n\}$ et $A = \{a_{f[1]} < \ldots < a_{f[m]}\}$
  $(q - q')^o = q^o - q'^o \qquad (q \cup q')^o = q^o \cup q'^o$

- La jointure naturelle se simule à coup de produits, sélections, et projections…

Boucler la boucle (2/2)

Boucler la boucle (2/2)

- $(q \bowtie q')^o = \pi_L(\sigma_F(q^o \times q'^o))$ où $q \bowtie q' : \{a_1 < \ldots < a_p\}$,

  $q : \{a_{f[1]} < \ldots < a_{f[m]}\}$, $q' : \{a_{g[1]} < \ldots < a_{g[n]}\}$

  F est la conjonction des égalités

  $f[i]=g[j]+m$,

  lorsque $1 \leq i \leq m$, $1 \leq j \leq n$, et $f[i]=g[j]$

  L = liste de longueur p,

  dont le $k$ième élément

  est i si $f[i]=k$, $j+m$ si $k \notin \mathrm{Im}\, f$ et $g[j]=k$

Boucler la boucle (2/2)

- $(q \bowtie q')^o = \pi_L(\sigma_F(q^o \times q'^o))$ où $q \bowtie q' : \{a_1 < \ldots < a_p\}$,

  $q : \{a_{f[1]} < \ldots < a_{f[m]}\}$, $q' : \{a_{g[1]} < \ldots < a_{g[n]}\}$

  F est la conjonction des égalités

  $f[i]=g[j]+m$,

  lorsque $1 \leq i \leq m$, $1 \leq j \leq n$, et $f[i]=g[j]$

  L = liste de longueur p,

  dont le $k$ième élément

  est i si $f[i]=k$, $j+m$ si $k \notin \mathrm{Im}\, f$ et $g[j]=k$

- **Exemple:** si R : $\{a < b < d\}$ et S : $\{b < c < e\}$,

  $(R \bowtie S)^o = \pi_{[1,2,5,3,6]}(\sigma_{2=4}(R^o \times S^o))$ (f:1↦1,2↦2,3↦4, g:1↦2,2↦3,3↦5)

Requêtes conjonctives

- Seuls connecteurs logiques autorisés: $\exists$ et $\wedge$ [Alice, chapitre 4], égalité, $\top$ interdits
- Les requêtes les plus utiles en pratique
- Forme normale: $\{\underline{x} \mid \exists\, x_1, \ldots, x_n \,.\, \bigwedge_{i=1}^{n} R_i(\underline{u_i})\}$, $n \geq 1$

Propriétés des requêtes conjonctives

- Toutes les requêtes conjonctives q sont:
  — **monotones** [Alice, Prop. 4.2.2 p.42]:
  pour toutes instances **I** et **J** sur (T, schema) et dom
  telles que **I**(R) ⊆ **J**(R) pour toute R dans T,
  $q(\text{dom},\mathbf{I}) \subseteq q(\text{dom},\mathbf{J})$
  — **satisfiables** [ibid.]:
  il existe toujours une instance **I** telle que $q(\text{dom},\mathbf{I}) \neq \emptyset$
  — **indépendantes du domaine** (et même **contraintes**)

Satisfiabilité des requêtes conjonctives

- Soit $q = \{\underline{x} \mid \exists\, x_1, \ldots, x_n \,.\, \bigwedge_{i=1}^{n} R_i(\underline{u_i})\}$
- On rappelle que dom est non vide
  On définit $\mathbf{I}(R_i) = \mathrm{dom}^{n_i}$, si $R_i : n_i$, par exemple
- Ceci contraste avec la question de la satisfiabilité des requêtes FOL (même contraintes),
  qui est indécidable par le théorème de Trakhtenbrot
  (voir transparents de R. Pichler, chapitre 4)
- Pour plus d'informations sur les requêtes conjonctives, voir transparents de R. Pichler, chapitre 6