Louis Jachiet
louis.jachiet@telecom-paris.fr

Bases de données

Une alternative à l'enseignement de SQL

Mini CV

Louis Jachiet
MCF Télécom Paris
depuis 2019

- Recherche autour de
  - Algorithmique
  - Bases de données
  - modèle RAM
- Membre de France-IOI
- Examinateur TP d'info au CCMP

Programme de ce cours

Premier temps

Un cours de SQL tel qu'enseigné à Télécom

Second temps

Une exploration de la théorie sous-jacente :

- Bases de données canoniques
- Homomorphismes
- etc.

Avant propos

Theaters

Name	Address	nbRooms
“La Nef”	“bd Édouard Rey”	7
“Le Mélies”	“caserne de Bonne”	3
“Le Club”	“rue Phalanstère”	3

Casting

Movie	Person	Role
“Inception”	“Elliot Page”	Actor
“Inception”	“Leonardo DiCaprio”	Actor
“Inception”	“Christopher Nolan”	Director
“Toy Story 3”	“Tom Hanks”	Voice Actor
“Mamma Mia”	“Meryl Streep”	Actor
“Mamma Mia”	“ Phyllida Lloyd”	Director

Projection

Title	Date	Theater
“Inception”	12/08/2010 20h	“Le Mélies”
“Toy Story 3”	13/08/2010 17h	“Le Club”
“Toy Story 3”	13/08/2010 20h	“Le Club”
“Toy Story 3”	10/08/2010 17h	“Le Mélies”
“Akmareul boatda”	10/08/2010 16h	“Le Club”
“How to train your dragon”	12/03/2010 18h	“Pathé Chavant”

Writing your first queries

A query is a question

You should think about ***what*** you want, not ***how*** to get it.

Doing examples

Forget about SQL, write a **minimal** database that has an answer to your query :

- Is your example **correct** ?
- Are all tuples **required** ?
- Is your example **generic** ?

Writing your first queries

A query is a question

You should think about ***what*** you want, not ***how*** to get it.

Doing examples

Forget about SQL, write a **minimal** database that has an answer to your query :

- Is your example **correct** ?
- Are all tuples **required** ?
- Is your example **generic** ?

Information

Minimal here means **no tuple** can be removed without removing the solution, it is not minimal in number of tuples. Generic means we are looking for the minimal database that has the most tuples with the least number of constants or re-use of a variable.

Example 1 : Size of France

Question

What is the size of France ?

Example database

name	continent	area	population	gdp	capital	tld	flag

Example 1 : Size of France

Question

What is the size of France ?

Example database

name	continent	area	population	gdp	capital	tld	flag
France		$\underline{\alpha}$					

Example 2 : Friend of a friend

Question

What are the people that are followed by people that Alice follows on social network Y ?

Example

follower	followee

Example 2 : Friend of a friend

Question

What are the people that are followed by people that Alice follows on social network Y ?

Example

follower	followee
Alice	α
α	$\underline{\beta}$

Example 2 : Friend of a friend

Question

What are the people that are followed by people that Alice follows on social network Y ?

Example

follower	followee
Alice	α
α	$\underline{\beta}$

Information

Be careful about being generic here, a single tuple (Alice,Alice) is minimal but not generic !

Example 3 : Movies with Meryl Streep acting

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role

Example 3 : Movies with Meryl Streep acting

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role
$\underline{\alpha}$	"Meryl Streep"	Actor

Example 4 : When can I see Meryl Streep acting ?

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role

Example 4 : When can I see Meryl Streep acting ?

Projection

movie	date	theater
β	$\underline{\alpha}$	

Theaters

name	address	nbRooms

Casting

movie	person	role
β	"Meryl Streep"	Actor

Example 5 : Where and when can I see Meryl Streep acting ?

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role

Example 5 : Where and when can I see Meryl Streep acting ?

Projection

movie	date	theater
β	$\underline{\alpha}$	γ

Theaters

name	address	nbRooms
γ	$\underline{\theta}$	

Casting

movie	person	role
β	"Meryl Streep"	Actor

Example 6 : Is there a movie with Meryl Streep and Pierce Brosnan ?

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role

Example 6 : Is there a movie with Meryl Streep and Pierce Brosnan ?

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role
$\underline{\alpha}$	"Meryl Streep"	Actor
$\underline{\alpha}$	"Pierce Brosnan"	Actor

Translating examples into queries

Example table world

name	continent	area	population	gdp	capital
France		$\underline{\alpha}$			

t

Becomes the following query

```
SELECT DISTINCT t.area
FROM world t
WHERE t.name = 'France' ;
```

Translating examples into queries

General translation into queries

- give a name to each tuple in the example
- add `SELECT DISTINCT` with underlined elements
- add `FROM table1 name1, ..., tablek namek`
- with `WHERE` and `AND` add conditons for :
  - each constant
  - each re-use of a variable

Information

Utile pour plus tard : cette transformation est inversible !

Example 2 : Friend of a friend

Table Follows

follower	followee	
Alice	β	t1
β	$\underline{\alpha}$	t2

```
SELECT DISTINCT t2.followee
FROM Follows t1, Follows t2
WHERE t1.follower = 'Alice'
  AND t1.followee = t2.follower
```

Example 3 : Movies with Meryl Streep acting

Casting

movie	person	role
α	"Meryl Streep"	"Actor"

c

Solution

```
SELECT DISTINCT c.movie
FROM casting c
WHERE c.person = 'Meryl Streep'
  AND c.role = 'Actor'
```

Example 4 : When can I see Meryl Streep acting ?

Projection

movie	date	theater
β	$\underline{\alpha}$	

p

Casting

movie	person	role
β	"Meryl Streep"	"Actor"

c

Solution

```
SELECT DISTINCT p.date
FROM casting c, projection p
WHERE c.person = 'Meryl Streep'
  AND c.role = 'Actor'
  AND c.movie = p.movie
```

Example 5 : Where and when can I see Meryl Streep acting ?

Projection

movie	date	theater
β	$\underline{\alpha}$	γ

p

Theaters

name	address	nbRooms
γ	$\underline{\theta}$	

t

Casting

movie	person	role
β	"Meryl Streep"	Actor

c

Example 5 : Where and when can I see Meryl Streep acting ?

Translation to SQL

```
SELECT DISTINCT p.date, t.address
FROM projection p,
     theater t,
     casting c
WHERE c.person = 'Meryl Streep'
  AND c.role = 'Actor'
  AND c.movie = p.movie
  AND p.theater = t.theater
```

Example 6 : Is there a movie with Meryl Streep and Pierce Brosnan ?

Projection

movie	date	theater

Theaters

name	address	nbRooms

Casting

movie	person	role
$\underline{\alpha}$	"Meryl Streep"	Actor
$\underline{\alpha}$	"Pierce Brosnan"	Actor

Example 6 : Is there a movie with Meryl Streep and Pierce Brosnan ?

Translation to SQL

```
SELECT DISTINCT 1c.movie
FROM casting c1,
     casting c2
WHERE c1.person = 'Meryl Streep'
  AND c2.person = 'Pierce Brosnan'
  AND c1.movie = c2.movie
```

Is this technique enough ?

No, it supports only conjunctive queries with constants :

- No union
- No aggregation
- No negation (inequalities between variables is not enough)

More on this on a later lesson !

Idiomatic SQL

In most programming languages you break down your problems into small steps ; you can do it in SQL but it is better to (try to) think globally.

Transforming ill defined ideas into code is a key difficulty for this class.

Que faire à partir de là

- mettre le plus vite possible les élèves face à du code et de la pratique
- beaucoup d'exercices avec un niveau crescendo
- dans ces exercices, on commence à rappeler que SQL n'est pas limité à ce que l'on a vu en introduisant quelques fonctionnalités supplémentaires.

Voir par exemple les exercices actuellement sur `https://sql.jachiet.com`

Section 1 : **Requêtes conjonctives**

Modèle relationnel comme logique du premier ordre

Définition (Schéma)

- des prédicats $R_1, \ldots, R_k$
- avec une arité $a_1, \ldots, a_k$

Pas de fonction !

Définition (Contenu)

Un modèle du schéma : un domaine et la donnée, pour chaque prédicat, de quand il est vrai.

Modèle relationnel comme logique du premier ordre

Définition (Schéma)

- des prédicats $R_1, \ldots, R_k$
- avec une arité $a_1, \ldots, a_k$

Pas de fonction !

Définition (Contenu)

Un modèle du schéma : un domaine et la donnée, pour chaque prédicat, de quand il est vrai.

Exemple

- `Casting(m,p,r)` est vrai quand la personne `p` intervient dans le film `m` avec le rôle `r`.
- `Projection(m,d,t)` est vrai quand le film `m` est projeté dans le cinéma `t` à la date `d`.

Modèle relationnel comme logique du premier ordre

Définition (Schéma)

- des prédicats $R_1, \ldots, R_k$
- avec une arité $a_1, \ldots, a_k$

Pas de fonction !

Définition (Contenu)

Un modèle du schéma : un domaine et la donnée, pour chaque prédicat, de quand il est vrai.

Définition (Requête)

Une formule de la logique du premier ordre sur la schéma.

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Dans quel film Meryl Streep joue ?

$$Q(m) = Casting(m, \text{"Meryl Streep"}, \text{"Actor"})$$

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Quand puis-je voir un film Meryl Streep ?

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Quand puis-je voir un film Meryl Streep ?

$$Q(d) = \exists m, t \quad Casting(m, \text{"Meryl Streep"}, \text{"Actor"}) \wedge Projection(m, d, t)$$

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Où et quand puis-je voir un film de Meryl Streep ?

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Où et quand puis-je voir un film de Meryl Streep ?

$$\begin{aligned} Q(d,o) \quad = \exists m,t,n \quad & Casting(m, \text{"Meryl Streep"}, \text{"Actor"}) \\ & \wedge Projection(m,d,t) \\ & \wedge Theaters(t,o,n) \end{aligned}$$

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Quels sont les films avec Meryl Streep et Pierce Brosnan ?

Exemple

Schéma

- Projection(movie,date,theater)
- Theaters(name,address,nbRooms)
- Casting(movie,person,role)

Les noms des arguments ne font pas partie du schéma !

Quels sont les films avec Meryl Streep et Pierce Brosnan ?

$$Q(m) = \exists r1, r2 \quad Casting(m, \text{"Meryl Streep"}, r1) \wedge Casting(m, \text{"Pierce Brosnan"}, r2)$$

Requête conjonctive

Définition (Requête conjonctive (CQ))

Une requête conjonctive (CQ) est une requête de la forme :

$$Q(\vec{r}) = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$$

Où les $\vec{v_i}$ peuvent contenir des constantes ou des variables mais $\vec{r}$ doivent correspondre à l'ensemble des variables libres de l'ensemble de droite.

On peut écrire Q au lieu $Q(\vec{r})$

Requête conjonctive

Définition (Requête conjonctive (CQ))

Une requête conjonctive (CQ) est une requête de la forme :

$$Q(\vec{r}) = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$$

Où les $\vec{v_i}$ peuvent contenir des constantes ou des variables mais $\vec{r}$ doivent correspondre à l'ensemble des variables libres de l'ensemble de droite.

On peut écrire Q au lieu $Q(\vec{r})$

Définition (Solutions d'une requête)

Les solutions d'une requête Q sur une base de données D, c'est l'ensemble $Q[D]$ des fonctions de $\vec{r}$ vers le domaine de D qui rendent Q vrai.

Requête conjonctive et SQL

Fait

Les requêtes conjonctives correspondent exactement aux requêtes de la forme `SELECT DISTINCT ... FROM ... WHERE ...` *qui n'utilisent que des égalités dans le* `WHERE`*.*

Preuve informelle :

- les variables (après `DISTINCT`) sont les variables libres
- il y a une entrée dans le `FROM` pour chaque terme $R_{i_j}(\vec{v_j})$
- les égalités servent à encoder les constantes et les variables qui apparaissent plusieurs fois

Section 2 : Base de données associée à une requête conjonctive

Base de données associée à une requête conjonctive

Définition (Base de données d'une requête conjonctive)

Pour $Q(\vec{r}) = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$, on définit DB(Q) comme la base de données dont le domaine comprend les constantes et les variables de Q et un $R(\vec{y})$ est vrai s'il apparaît dans Q.

Base de données associée à une requête conjonctive

Définition (Base de données d'une requête conjonctive)

Pour $Q(\vec{r}) = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$, on définit DB(Q) comme la base de données dont le domaine comprend les constantes et les variables de Q et un $R(\vec{y})$ est vrai s'il apparaît dans Q.

Exemple

La requête $\exists m, t, n\ Casting(m, \text{"Meryl Streep"}, \text{"Actor"}) \wedge Projection(m, d, t) \wedge Theaters(t, o, n)$ devient

Projection

movie	date	theater
m	$\underline{d}$	t

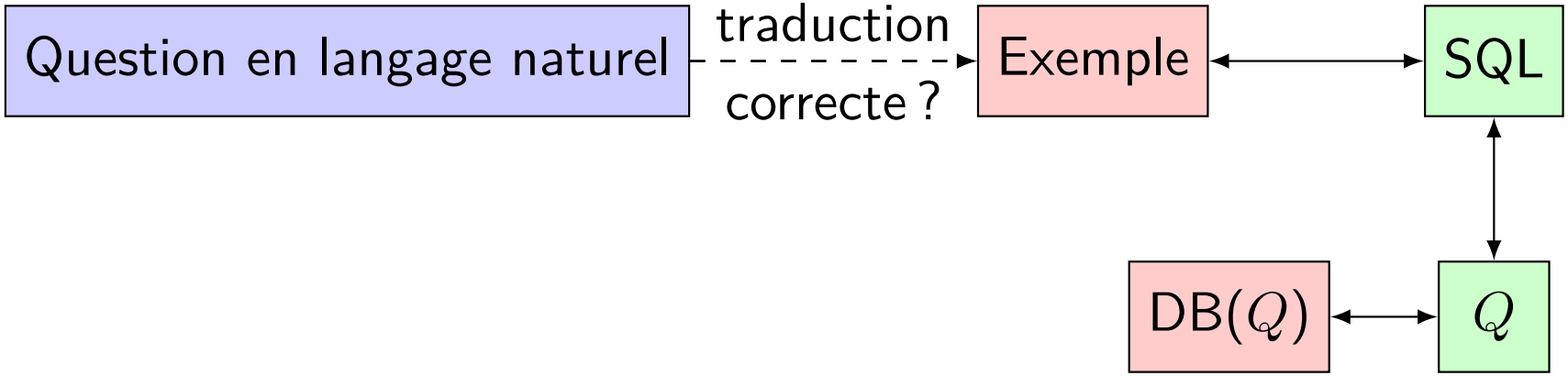
Casting

movie	person	role
m	"Meryl Streep"	"Actor"

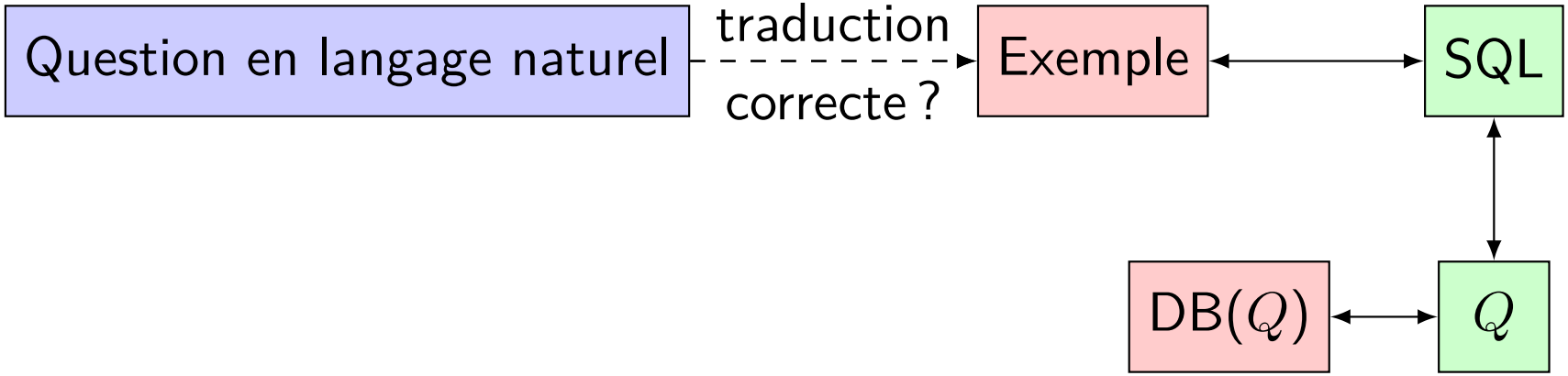
Theaters

name	address	nbRooms
t	$\underline{o}$	n

En résumé


Question en langage naturel
traduction
correcte ?
Exemple
SQL
DB(Q)
Q

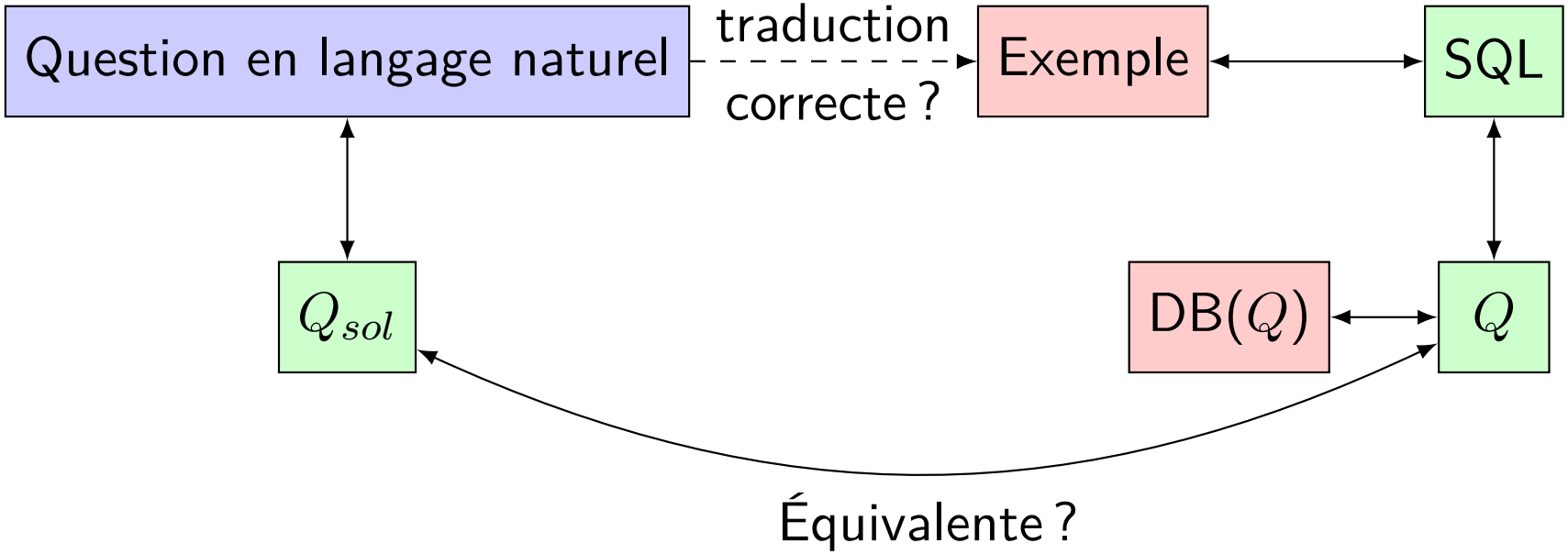
En résumé



Théorème

Si la question correspond à une requête conjonctive et si l'exemple trouvé est maximal (en nombre de tuples) parmi les exemples minimaux (au sens où l'on ne peut pas enlever de tuple sans enlever de solutions), alors la traduction est correcte.

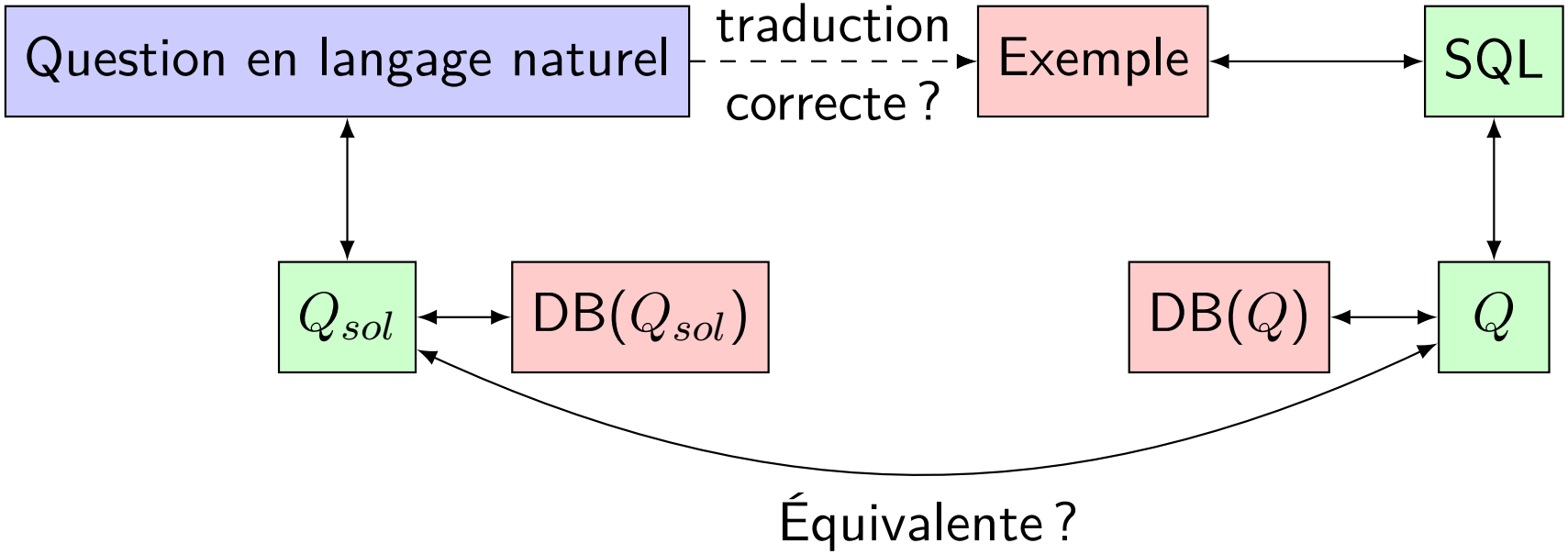
En résumé



Théorème

Si la question correspond à une requête conjonctive et si l'exemple trouvé est maximal (en nombre de tuples) parmi les exemples minimaux (au sens où l'on ne peut pas enlever de tuple sans enlever de solutions), alors la traduction est correcte.

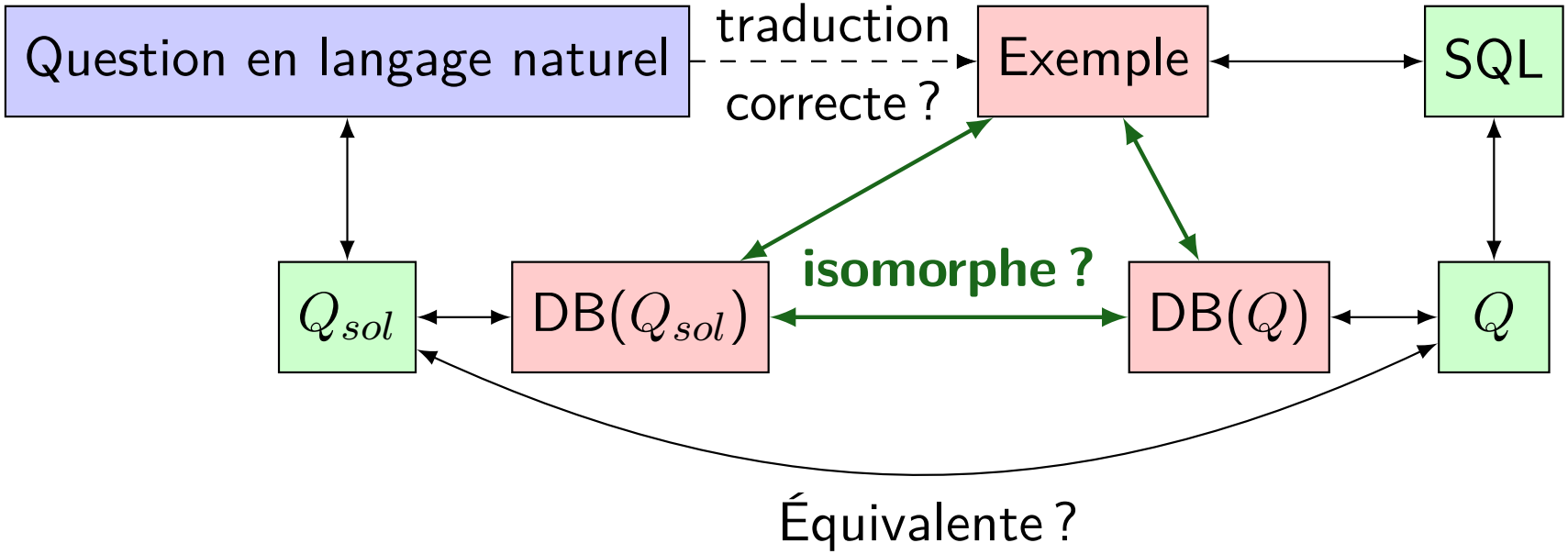
En résumé



Théorème

Si la question correspond à une requête conjonctive et si l'exemple trouvé est maximal (en nombre de tuples) parmi les exemples minimaux (au sens où l'on ne peut pas enlever de tuple sans enlever de solutions), alors la traduction est correcte.

En résumé



Théorème

Si la question correspond à une requête conjonctive et si l'exemple trouvé est maximal (en nombre de tuples) parmi les exemples minimaux (au sens où l'on ne peut pas enlever de tuple sans enlever de solutions), alors la traduction est correcte.

Section 3 : **Homomorphisme et équivalence de requêtes**

Équivalence de requête

Définition (Inclusion)

On dit que la requête Q_1 est incluse dans Q_2, noté $Q_1 \sqsubseteq Q_2$ quand $\forall D, Q_1[D] \subseteq Q_2[D]$.

Définition (Équivalence)

On dit que la requête Q_1 est équivalente à Q_2, noté $Q_1 \equiv Q_2$ quand $Q_1 \sqsubseteq Q_2$ et $Q_2 \sqsubseteq Q_1$.

Tout ceci n'est bien défini que si ce sont les mêmes variables libres !

Définition (Homomorphisme de requêtes)

Étant données requêtes conjonctives $Q_1(\vec{r}_1)$ et $Q_2(\vec{r}_2)$, un homomorphisme h de Q_1 dans Q_2 est une fonction des variables de Q_1 vers celles de Q_2 tels que si $R(v_1, \ldots, v_k)$ apparaît dans Q_1 alors $R(h(v_1), \ldots, h(v_k))$ apparaît dans D_2 et tel que $\vec{r}_2 = h(\vec{r}_1)$.

Définition (Isomorphisme de requêtes)

Deux requêtes $Q_1(\vec{r}_1)$ et $Q_2(\vec{r}_2)$ sont isomorphes lorsqu'il y un homomorphisme de Q_1 dans Q_2 qui est bijectif et dont la réciproque est un homomorphisme.

Résultats de Chandra et Merlin 1977

Théorème

Il existe un homomorphisme de $Q_1(\vec{r}_1)$ *dans* $Q_2(\vec{r}_2)$ *ssi* $\vec{r}_2 \in Q_1[DB(Q_2)]$.

Idée : on exhibe une bijection entre les homomorphismes et l'évaluation de la requête

Résultats de Chandra et Merlin 1977

Théorème

Il existe un homomorphisme de $Q_1(\vec{r}_1)$ dans $Q_2(\vec{r}_2)$ ssi $\vec{r}_2 \in Q_1[DB(Q_2)]$.

Théorème

Il existe un homomorphisme de Q_1 dans Q_2 ssi $Q_2 \sqsubseteq Q_1$

Idée : si on a un homomorphisme, on peut montrer que les résultats de Q_2 sont résultats de Q_1. Si $Q_2 \sqsubseteq Q_1$ en appliquant le théorème précédent, on a le résultat.

Résultats de Chandra et Merlin 1977

Théorème

Il existe un homomorphisme de $Q_1(\vec{r}_1)$ dans $Q_2(\vec{r}_2)$ ssi $\vec{r}_2 \in Q_1[DB(Q_2)]$.

Théorème

Il existe un homomorphisme de Q_1 dans Q_2 ssi $Q_2 \sqsubseteq Q_1$

Théorème

S'il existe une requête équivalente à Q avec moins d'atomes, on peut trouver une requête équivalente à Q en lui retirant des atomes.

Idée : prenons Q' minimale en nombre d'atomes et équivalente à Q, on a h_1 de Q dans Q' et h_2 de Q' dans Q. On regarde alors $Q'' = h_2(h_1(Q))$, clairement Q'' est un sous-ensemble des atomes de Q. On a un homomorphisme de Q'' dans Q (l'identité) et un autre de Q dans Q'' $(h_2 \circ h_1)$ donc $Q \equiv Q''$.

Retours aux exemples

On peut montrer que si l'élève choisi une solution correcte, minimale de taille maximale et générique alors c'est la bonne solution. En effet :

- par maximalité de Q on a $|Q| \geq |Q_{sol}|$;
- par minimalité, on a $|Q_{sol}| \geq |Q|$;
- Par la correction on déduit que $Q \sqsubseteq Q_{sol}$, et donc un homomorphisme de Q_{sol} dans Q ;
- par la généricité et minimalité, on va montrer que cette homomorphisme est bijectif et sa réciproque est inversible.

Section 4 : Détour sur la complexité

Complexité combinée et complexité en données

Définition (Complexité combinée)

La *complexité combinée* de l'évaluation de requêtes pour une classe de requête $\mathcal{C}$ c'est la complexité de calculer $Q[D]$ pour $Q \in \mathcal{C}$ en fonction de $|Q| + |D|$.

Pour avoir des problèmes de décision on regarde souvent des requêtes booléennes

Définition (Complexité en données)

La *complexité en données* de l'évaluation de requêtes pour une classe de requête $\mathcal{C}$ c'est la complexité de calculer $Q[D]$ pour une requête $Q \in \mathcal{C}$ fixée en fonction de $|D|$.

Complexité en données pour les requêtes conjonctives

Rappel

$$Q(\vec{r}) = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$$

Pour une requête fixée

Il suffit de tester toutes les valeurs du domaine pour chacune des variables dans $\vec{r}$ et $\vec{x}$, ça nous donne un algorithme en $O(|D|^k)$ où $k = |\vec{r}| + |\vec{x}|$ est fixé (ça ne dépend que de Q).

Complexité combinée pour les requêtes conjonctives

Rappel

$$Q = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$$

Quelle complexité

- P ? P-complet ?
- NP ? NP-complet ?
- PSPACE ? PSPACE-complet ?

Preuve d'appartenance à NP

Entrée

- $Q = \exists \vec{x}\ R_{i_1}(\vec{v_1}) \wedge \cdots \wedge R_{i_\ell}(\vec{v_\ell})$
- D

Certificat

Une fonction qui envoie les variables de $\vec{x}$ dans D

Vérificateur

On teste pour tout j si $R_{i_j}(f(\vec{v_j})) \in D$

NP-difficile : réduction depuis 3-SAT

Définition (3-SAT)

3-SAT c'est le problème de satisfaction d'une formule en forme normale conjonctive dans laquelle chaque conjoint contient 3 littéraux (v_i ou $\neg v_i$)

$$\varphi = \exists v_1, \ldots, v_k \bigwedge_i l_1^i \vee l_2^i \vee l_3^i$$

Réduction

- D contient deux prédicats :
  - neg d'arité 2 avec $neg(0,1)$ et $neg(1,0)$
  - or d'arité 3 avec $or(a,b,c)$ qui est vrai pour $(a,b,c) \in \{0,1\}^3 \setminus \{0\}^3$
- $Q = \exists x_1 ... x_k, y_1 ... y_k, \bigwedge_i neg(x_i, y_i) \bigwedge or(\ell_1^i, \ell_2^i, \ell_3^i)$ avec $\ell_j^i = x_k$ quand $l_j^i v_k$ and $\ell_j^i = y_k$ quand $l_j^i = \neg v_k$

Ce qu'il faut retenir

- On a présenté une manière d'introduire les requêtes SQL
- Cette une méthode assez restreinte (requêtes conjonctives)
- mais elle peut permettre de mieux comprendre les jointures, notamment les auto-jointures et d'éviter le passage par les requêtes imbriquées
- Cette technique puise sa source dans des constructions de théorie de base de données qui sont assez élégantes
- Pour qui viendrait à mon cours cet après-midi :
  - les notions de complexité en données et de complexité combinée
  - le lien entre les requêtes et la logique
  - nous verrons ce qu'il se passe quand on rajoute la négation, l'union ou la récursion