

BDD Cours 8 : Traduire TRC en SQL grâce aux lois d'équivalences logiques

C. Kuttler

21 octobre 2015

Nous traduisons TRC en SQL en utilisant trois lois d'équivalence (\equiv) logique :

1. définition de l'implication $A \Rightarrow B \equiv (\text{not } A) \text{ or } B$
2. définition de la quantification universelle, en passant par la quantification existentielle et la négation : $\forall x : P(x) \equiv \text{not } \exists x : \text{not } P(x)$ et une variante sous négation : $\text{not } \forall x : P(x) \equiv \exists x : \text{not } P(x)$
3. de Morgan
 - $\text{not } (A \text{ and } B) \equiv \text{not } A \text{ or } \text{not } B$
 - $\text{not } (A \text{ or } B) \equiv \text{not } A \text{ and } \text{not } B$

1 Portée maximale d'un avion

Problème 1. Quelle est la portée maximale d'un avion ?

- la portée d'un avion tel que pour tous les avions, la portée est égale ou inférieure
- la portée d'un avion tel qu'il N'existe PAS d'avion, tel que NOT(la portée est égale ou inférieure)
- la portée d'un avion telle qu'il n'existe pas d'avion, dont la portée est supérieure

$$\begin{aligned} & \{t | \exists a \in \text{avions} : t[\text{portee}] = a[\text{portee}] \\ & \text{and } \forall a2 \in \text{avions} : a2[\text{portee}] \leq a[\text{portee}]\} \\ & \equiv \end{aligned}$$

(élimination de \forall)

$$\begin{aligned} & \{t | \exists a \in \text{avions} : t[\text{portee}] = a[\text{portee}] \\ & \text{and } \text{not } \exists a2 \in \text{avions} : a2[\text{portee}] > a[\text{portee}]\} \end{aligned}$$

Cette dernière version se réécrit directement en SQL :

Requête 1.

```
select t.portee
from avions t
where not exists (select * from avions a2 where a2.portee > t.portee);
```

Remarque : cette première requête permet de facilement récupérer le nom de l'avion avec la portée minimale, en modifiant la clause select. Nous pouvons obtenir le même résultat, avec une requête SQL équivalente utilisant une sous-requete avec ALL :

Requête 2.

```
select t.portee
from avions t
where t.portee >=ALL (select portee from avions);
```

Regardons la tentative d'un debutant en SQL, qui démarre de la manière la plus simple d'obtenir la portée maximale d'un avion :

Requête 3.

```
select max(portee) from avions;
```

Dans ce cas, il faut imbriquer la requête initiale, pour obtenir l'identifiant de l'avion avec cette portée maximale :

Requête 4.

```
select aid from avions
where portee in (select max(portee) from avions);
```

2 Comment trouver le pilote maxivalent ?

Problème 2. Nous cherchons des pilotes certifié(s) pour TOUS les avions

- un pilote (eid) pour qui, pour tous les avions (aid), il existe une certification(aid,eid)
- un pilote (eid) pour qui, il n'existe pas d'avion (aid), tel qu'il n'existe pas de certification(aid,eid)

$$\begin{aligned}
 &\{t|\exists e \in \text{employees} : t[\text{enom}] = e[\text{enom}] \text{ and} \\
 &\quad \forall a \in \text{avions} : \\
 &\quad \exists c \in \text{certifications} : c[\text{aid}] = a[\text{aid}] \text{ and } c[\text{eid}] = e[\text{eid}]\} \\
 &\quad \equiv
 \end{aligned}$$

élimination de la quantification universelle

$$\begin{aligned}
 &\{t|\exists e \in \text{employees} : t[\text{enom}] = e[\text{enom}] \text{ and} \\
 &\quad \text{not } \exists a \in \text{avions} : \\
 &\quad \text{not } \exists c \in \text{certifications} : c[\text{aid}] = a[\text{aid}] \text{ and } e[\text{eid}] = c[\text{eid}]\}
 \end{aligned}$$

Requête 5.

```
select t.enom
from employees t
where not exists (
  select * from avions a
  where not exists (
    select * from certifications c where c.aid=a.aid and t.eid=c.eid));
```

Donc nous avons exprimé en SQL, comment identifier un pilote qualifié pour tous les avions en passant par deux sous-requetes NOT EXISTS, qui correspondent en TRC à deux quantifications existentielles sous controle d'une négation. Ce qui est equivalent à une quantification du genre *pour tous*. Quel details supplémentaire faut-il encore ajouter dans la requete ci-haut, et pourquoi?

3 Vendeur pas bon marché

Problème 3. Nous cherchons des vendeurs offrant uniquement des articles à plus de 100 € Nous pouvons exprimer cela de différentes manières équivalentes :

1. vendeur dont tous les articles coûtent plus de 100 €
2. un vendeur, tel que pour tous les articles, si l'article est offert par ce vendeur, alors son prix est >100 €
3. vendeur qui n'offre pas d'article à <100 €
4. vendeur pour qui il n'existe pas d'article à <100 €

Remarque : Il faut également assurer que le vendeur offre au moins un article à plus de 100 €. Les formules suivantes ne contiennent pas cette condition, pour simplifier.

Voilà la phrase (2) exprimée en TRC :

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[\text{fnom}] = f[\text{fnom}] \\ & \forall c \in \text{catalogue} : c[\text{fid}] = f[\text{fid}] \Rightarrow c[\text{prix}] > 100\} \\ & \equiv \end{aligned}$$

élimination de la quantification universelle

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[\text{fnom}] = f[\text{fnom}] \\ & \text{not } \exists c \in \text{catalogue} : \text{not } (c[\text{fid}] = f[\text{fid}] \Rightarrow c[\text{prix}] > 100)\} \\ & \equiv \end{aligned}$$

élimination de l'implication

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[\text{fnom}] = f[\text{fnom}] \\ & \text{not } \exists c \in \text{catalogue} : \underbrace{\text{not}}_{\text{prochaine suppression}} (c[\text{fid}] \neq f[\text{fid}] \text{ or } c[\text{prix}] > 100)\} \\ & \equiv \end{aligned}$$

de Morgan

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[\text{fnom}] = f[\text{fnom}] \\ & \text{not } \exists c \in \text{catalogue} : (c[\text{fid}] = f[\text{fid}] \text{ and } c[\text{prix}] \leq 100)\} \end{aligned}$$

Cette dernière version correspond à la phrase (4), et contrairement à (2), peut être réécrite en SQL comme suit :

```
Requête 6. select t.fnom
from fournisseurs t
where not exists
  (select * from catalogue c where c.fid=t.fid and c.prix <=100);
```

Remarque : cette requête rend des fournisseurs n'ayant pas d'articles à moins de 100 €, même si ces vendeurs n'offrent aucun article. Il faut vérifier que le fournisseur offre également (au moins) un article à plus de 100 €. On peut le faire explicitement en ajoutant une deuxième condition existentielle :

Requête 7. `select t.fnom
from fournisseurs t
where not exists
 (select * from catalogue c where c.fid=t.fid and c.prix <=100)
and exists
 (select * from catalogue c where c.fid=t.fid and c.prix >100)
;`

Ou, plus simplement :

Requête 8. `select t.fnom
from fournisseurs t, catalogue c
where c.fid=t.fid and c.prix >100 and
 not exists
 (select * from catalogue c2 where c2.fid=t.fid and c2.prix <=100);`

4 Vendeur n'offrant aucun article vert

Problème 4. Nous cherchons un vendeur (ou des vendeurs), qui n'offre(nt) pas d'articles verts

- tous les articles offerts par ce vendeur sont d'une autre couleur que vert
- pour tous les articles du catalogue, si cet article est offert par ce fournisseur, alors cet article n'est pas vert
- il n'existe pas d'article de ce vendeur qui soit vert

$$\begin{aligned} & \{t | \exists f \in \text{fournisseurs} : f[\text{fnom}] = t[\text{fnom}] \text{ and } \\ & \quad \forall c \in \text{catalogue}, \\ & \quad \forall a \in \text{articles} : (c[\text{fid}] = f[\text{fid}] \text{ and } a[\text{aid}] = c[\text{aid}]) \Rightarrow a[\text{acoul}] \neq \text{vert}'\} \\ & \quad \equiv \end{aligned}$$

élimination de la quantification universelle $\forall c \in \text{catalogue}$

$$\begin{aligned} & \{t | \exists f \in \text{fournisseurs} : f[\text{fnom}] = t[\text{fnom}] : \\ & \quad \text{not } \exists c \in \text{catalogue} : \\ & \quad \underbrace{\text{not}}_{\text{prochaine réécriture}} \left(\forall a \in \text{articles} : (c[\text{fid}] = f[\text{fid}] \text{ and } a[\text{aid}] = c[\text{aid}]) \Rightarrow a[\text{acoul}] \neq \text{vert}' \right) \} \\ & \quad \equiv \end{aligned}$$

avec la variante de notre règle pour l'élimination du pour tous (un NOT en amont) $NOT \forall x : P(x) \equiv \exists x : not P(x)$

$$\begin{aligned} & \{t | \exists f \in \text{fournisseurs} : f[\text{fnom}] = t[\text{fnom}] : \\ & \quad \text{not } \exists c \in \text{catalogue}, \\ & \quad (\exists a \in \text{articles} : NOT(\\ & \quad \quad (c[\text{fid}] = f[\text{fid}] \text{ and } a[\text{aid}] = c[\text{aid}]) \underbrace{\Rightarrow}_{\text{prochaine réécriture}} a[\text{acoul}] \neq \text{vert}') \} \\ & \quad \equiv \end{aligned}$$

réécriture de l'implication

$$\begin{aligned} & \{t | \exists f \in \text{fournisseurs} : f[\text{fnom}] = t[\text{fnom}] : \\ & \text{not } \exists c \in \text{catalogue}, (\exists a \in \text{articles} : \underbrace{\text{NOT}}_{\text{prochaine réécriture}} (\\ & \text{NOT}(c[\text{fid}] = f[\text{fid}] \text{ and } a[\text{aid}] = c[\text{aid}]) \text{ or } a[\text{acoul}] \neq \text{vert}') \} \\ & \equiv \end{aligned}$$

de Morgan

$$\begin{aligned} & \{t | \exists f \in \text{fournisseurs} : f[\text{fnom}] = t[\text{fnom}] : \\ & \text{not } \exists c \in \text{catalogue}, (\exists a \in \text{articles} : (\\ & \text{not } \text{not } (c[\text{fid}] = f[\text{fid}] \text{ and } a[\text{aid}] = c[\text{aid}]) \text{ and } \text{not } a[\text{acoul}] \neq \text{vert}') \} \end{aligned}$$

suppression de la double négation, remplacement de (not ≠) par =

$$\begin{aligned} & \{t | \exists f \in \text{fournisseurs} : f[\text{fnom}] = t[\text{fnom}] : \\ & \text{not } \exists c \in \text{catalogue}, (\exists a \in \text{articles} : (\\ & (c[\text{fid}] = f[\text{fid}] \text{ and } a[\text{aid}] = c[\text{aid}]) \text{ and } a[\text{acoul}] = \text{vert}') \} \end{aligned}$$

La dernière version se réécrit directement en SQL :

Requête 9. `select t.fnom
from fournisseurs t
where not exists (
 select * from catalogue c
 where exists (
 select * from articles a
 where c.fid=t.fid and a.aid=c.aid and a.acoul='vert'
)
);`

qu'on peut simplifier en :

Requête 10. `select t.fnom
from fournisseurs t
where not exists(
 select * from catalogue c, articles a
 where a.aid=c.aid and c.fid=t.fid and a.acoul='vert'
);`

5 Le cas inverse

Finalement nous cherchons l'inverse du probleme precedent :

Problème 5. Comment identifier le vendeur qui offre uniquement des articles verts ?
— vendeur offrant uniquement des articles verts

- si un article est vendu par ce fournisseur, il est vert
- un fournisseur tel que, pour tous les articles, si cet article est vendu par ce fournisseur, il est vert
- un fournisseur qui ne vend pas d'articles dont la couleur n'est pas verte

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[f\text{nom}] = f[f\text{nom}] \\ & \quad \text{and } \forall c \in \text{catalogue}, \forall a \in \text{articles} : \\ & \quad c[f\text{id}] = f[f\text{id}] \text{ and } a[a\text{id}] = c[a\text{id}] \Rightarrow a[a\text{coul}] = 'vert'\} \\ & \equiv \end{aligned}$$

élimination de la quantification universelle

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[f\text{nom}] = f[f\text{nom}] \\ & \quad \text{and not } \exists c \in \text{catalogue}, \text{not } (\forall a \in \text{articles} : c[f\text{id}] = f[f\text{id}] \text{ and } a[a\text{id}] = c[a\text{id}] \\ & \quad \Rightarrow a[a\text{coul}] = 'vert')\} \\ & \equiv \\ & \{t | \exists f \in \text{fournisseur} : t[f\text{nom}] = f[f\text{nom}] \\ & \quad \text{and not } \exists c \in \text{catalogue}, (\exists a \in \text{articles} : \text{not } (c[f\text{id}] = f[f\text{id}] \text{ and } a[a\text{id}] = c[a\text{id}] \\ & \quad \Rightarrow a[a\text{coul}] = 'vert'))\} \\ & \equiv \end{aligned}$$

implication

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[f\text{nom}] = f[f\text{nom}] \\ & \quad \text{and not } \exists c \in \text{catalogue}, (\exists a \in \text{articles} : \text{not } (\text{not } (c[f\text{id}] = f[f\text{id}] \text{ and } a[a\text{id}] = c[a\text{id}]) \\ & \quad \text{or } a[a\text{coul}] = 'vert'))\} \\ & \equiv \end{aligned}$$

de Morgan, éliminer la double négation

$$\begin{aligned} & \{t | \exists f \in \text{fournisseur} : t[f\text{nom}] = f[f\text{nom}] \\ & \quad \text{and not } \exists c \in \text{catalogue}, (\exists a \in \text{articles} : c[f\text{id}] = f[f\text{id}] \text{ and } a[a\text{id}] = c[a\text{id}] \\ & \quad \text{and } a[a\text{coul}] \neq 'vert')\} \end{aligned}$$

Requête 11. `select t.fnom`
`from fournisseurs t`
where not exists (
 `select * from catalogue c`
 where exists (
 `select * from articles a`
 where c.fid=t.fid and a.aid=c.aid and a.acoul <> 'vert')
);

version directe et intuitive : il ne vend pas d'article qui n'est pas vert

Requête 12. `select t.fnom`
`from fournisseurs t`
where not exists (
 (`select * from catalogue c, articles a`
 where c.aid=a.aid and c.fid=t.fid and a.acoul <> 'vert')
);

Même remarque que pour l'exemple 3. Il faut assurer que le vendeur offre au moins un article vert, dans la requête. Autrement, on obtient le fournisseur qui n'a pas d'article dans le catalogue. Nous avons omis cette condition en TRC pour simplifier.