Abstract

For now : I don't consider full-join. To do : A lot :) (SAT/SMT SOLVER ???) to evaluate what we loose ? UNION and INTERSECTION VERIFY EXCEPT.

1 Preliminaries

Definition 1. *We denote the Set of well formed select query without agregation, full join and null test by* $[\![SQL]\!]$
We denote the Set of well formed select query without agregation and full join by $[\![SQL]\!]_\perp$

Definition 2. *Let's a Select query* $Q \in [\![SQL]\!]$ *a tuple* (Σ, R, H, P) *such that*

$$\Sigma \subseteq \{r_i.a_i | \exists r_i \in R \wedge \exists a_i \in r_i\}$$
$P \subseteq \{p_i = r_j.a_j | r_j \notin R\}$ *a set of external parameter.*
R *a set of relation.*
H *belongs to the following grammar*

$$\begin{aligned} H ::= {} & r_i.a_i = c_i \mid r_i.a_i \neq c_j \mid r_i.a_i = r_j.a_j \mid r_i.a_i \neq r_j.a_j \mid r_i.a_i = p_i \mid \\ & exists(Q) \mid notexists(Q) \mid in(r_i.a_i, Q) \mid notin(r_i.a_i, Q) \mid \\ & H \wedge H \mid H \vee H \end{aligned}$$

Definition 3. *Let's a Select query* $Q \in [\![SQL]\!]_\perp$ *a tuple* (Σ, R, H, P) *such that*

$$\Sigma \subseteq \{r_i.a_i | \exists r_i \in R \wedge \exists a_i \in r_i\}$$
$P \subseteq \{p_i = r_j.a_j | r_j \notin R\}$ *a set of external parameter.*
R *a set of relation.*
$H_\perp$ *belongs to the following grammar*

$$\begin{aligned} H_\perp ::= {} & r_i.a_i = c_i \mid r_i.a_i \neq c_j \mid r_i.a_i = r_j.a_j \mid r_i.a_i \neq r_j.a_j \mid \\ & r_i.a_i = p_i \mid null(r_i.a_i) \mid const(r_i.a_i) \\ & exists(Q_\perp) \mid notexists(Q_\perp) \mid in(r_i.a_i, Q_\perp) \mid notin(r_i.a_i, Q_\perp) \mid \\ & H_\perp \wedge H_\perp \mid H_\perp \vee H_\perp \end{aligned}$$

We denote $(\Sigma, R, H, P)[x]$ the query $(\Sigma, R, H, P \cup x)$
We denote $(\Sigma, R, H, P)_*$ the query $(*, R, H, P)$

Proposition 1.

$$[\![SQL]\!] \subset [\![SQL]\!]_\perp$$

Proof. immediate i guess ? □

Definition 4. *We call a bag B a function $D \to \mathbb{N}$ such that $B(x)$ represents the multiplicity of x in the bag B.*

Definition 5.

$$\begin{aligned} \forall x, \emptyset(x) &= 0 \\ \forall x, (B_1 \cap B_2)(x) &= min(B_1(x), B_2(x)) \\ \forall x, (B_1 \cup B_2)(x) &= max(B_1(x), B_2(x)) \\ \forall x, (B_1 \uplus B_2)(x) &= B_1(x) + B_2(x) \\ \forall x, (B_1 \setminus B_2)(x) &= max(0, B_1(x) - B_2(x)) \\ \forall x, [\![a]\!](x) &= \begin{cases} 1 \ if \ x = a \\ 0 \ otherwise \end{cases} \\ \forall x, [\![a^n]\!](x) &= \begin{cases} n \ if \ x = a \\ 0 \ otherwise \end{cases} \\ \forall x, [\![y^n | P(y, n)]\!](x) &= max(\{i | P(x, i)\}) \\ x \in B &\iff B(x) \geq 1 \\ x \in^n B &\iff B(x) \geq n \\ x \notin B &\iff B(x) = 0 \\ B_1 = B_2 &\iff \forall x, B_1(x) = B_2(x) \\ B_1 \subseteq B_2 &\iff \forall x, B_1(x) \leq B_2(x) \\ \{B\} &= \{x | B(x) \geq 1\} \end{aligned}$$

2 Semantics

Definition 6.

$$\sigma_\Sigma(x) = (x[r_i.a_i] | r_i.a_i \in \Sigma)$$

$$\sigma_*(x) = x$$

Definition 7.

$$\sigma_\Sigma(B) = [\![y^n | n = \sum_{x \in \{z | z \in \{B\} \wedge \sigma_\Sigma(z) = y\}} B(x)]\!]$$

Definition 8.

$$\begin{aligned}
Eval_{SQL}((\Sigma, R, H_1 \wedge H_2, P), D) &= \sigma_\Sigma(Eval_{SQL}((*, R, H_1, P), D) \cap Eval_{SQL}((*, R, H_2, P), D)) \\
Eval_{SQL}((\Sigma, R, H_1 \vee H_2, P), D) &= \sigma_\Sigma(Eval_{SQL}((*, R, H_1, P), D) \cup Eval_{SQL}((*, R, H_2, P), D)) \\
Eval_{SQL}((\Sigma, R, null(r_i.a_i), P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] = \bot]\!]) \\
Eval_{SQL}((\Sigma, R, const(r_i.a_i), P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, r_i.a_i = c_i, P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] = c_i]\!]) \\
Eval_{SQL}((\Sigma, R, r_i.a_i = r_j.a_j, P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] = x[r_j.a_j] \wedge x[r_i.a_i] \neq \bot \wedge x[r_j.a_j] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, r_i.a_i = p_i, P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] = P[p_i] \wedge x[r_i.a_i] \neq \bot \wedge P[p_i] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, r_i.a_i \neq c_i, P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] \neq c_i \wedge x[r_i.a_i] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, r_i.a_i \neq r_j.a_j, P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] \neq x[r_j.a_j] \wedge x[r_i.a_i] \neq \bot \wedge x[r_j.a_j] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, r_i.a_i \neq p_i, P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge x[r_i.a_i] \neq P[p_i] \wedge x[r_i.a_i] \neq \bot \wedge P[p_i] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, exists(Q), P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge Eval_{SQL}(Q[x], D) \neq \emptyset]\!]) \\
Eval_{SQL}((\Sigma, R, notexists(Q), P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \wedge Eval_{SQL}(Q[x], D) = \emptyset]\!]) \\
Eval_{SQL}((\Sigma, R, in(\Sigma_1, Q), P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \\
&\qquad \wedge \forall y \in Eval_{SQL}(Q, D), \forall i \in [\![1; |\Sigma_1|]\!], \sigma_{\Sigma_1}(x)[i] = y[i] \\
&\qquad\qquad \wedge \sigma_{\Sigma_1}(x)[i] \neq \bot \wedge y[i] \neq \bot]\!]) \\
Eval_{SQL}((\Sigma, R, notin(\Sigma_1, Q), P), D) &= \sigma_\Sigma([\![x^n | R(x) = n \\
&\qquad \wedge \forall y \in Eval_{SQL}(Q, D), \forall i \in [\![1; |\Sigma_1|]\!], \sigma_{\Sigma_1}(x)[i] \neq y[i] \\
&\qquad\qquad \vee \sigma_{\Sigma_1}(x)[i] = \bot \vee y[i] = \bot]\!])
\end{aligned}$$

$$\begin{aligned}
Eval_{SQL}(Q \setminus Q', D) &= Eval_{SQL}(Q, D) \setminus Eval_{SQL}(Q', D) \\
Eval_{SQL}(Q \cup Q', D) &= Eval_{SQL}(Q, D) \cup Eval_{SQL}(Q', D) \\
Eval_{SQL}(Q \uplus Q', D) &= Eval_{SQL}(Q, D) \uplus Eval_{SQL}(Q', D) \\
Eval_{SQL}(Q \cap Q', D) &= Eval_{SQL}(Q, D) \cap Eval_{SQL}(Q', D) \\
Eval_{SQL}(distinct(Q), D) &= [\![x^1 | x \in \{Eval_{SQL}(Q, D)\}]\!]
\end{aligned}$$

Definition 9.

$$\forall Q \in [\![SQL]\!], cert_{bad\bot}((\Sigma, R, H, P), D) = \sigma_\Sigma([\![x^n | \forall h, h(x) \in^n Eval_{SQL}((*, R, H, h(P)), h(D))]\!])$$

Not the definition we want as we are going to loose multiplicity as soon as there is a null in the row. (Codd Null)

Definition 10.

$\forall Q \in [\![SQL]\!], cert_{\perp}((\Sigma, R, H, P), D) = \sigma_{\Sigma}([\![x^n | R(x) = n \wedge \forall h, h(x) \in \{Eval_{SQL}((*, R, H, h(P)), h(D))\}]\!])$

It's equivalent for complete row, and over incomplete row we keep multiplicity.

Definition 11.

$\forall Q \in [\![SQL]\!], posi_{\perp}((\Sigma, R, H, P), D) = \sigma_{\Sigma}([\![x^n | \exists h, h(x) \in^n Eval_{SQL}((*, R, H, h(P)), h(D))]\!])$

Same issue here, we can increase multiplicity fictivly when null occur.

Definition 12.

$\forall Q \in [\![SQL]\!], posi_{\perp}((\Sigma, R, H, P), D) = \sigma_{\Sigma}([\![x^n | R(x) = n \wedge \exists h, h(x) \in \{Eval_{SQL}((*, R, H, h(P)), h(D))\}]\!])$

It's equivalent for complete row, and over incomplete row we keep multiplicity.

3 Translation

$$(\Sigma, R, H, P)^+ \rightarrow (\Sigma, R, H^*, P)$$
$$(\Sigma, R, H, P)^? \rightarrow (\Sigma, R, H^{**}, P)$$

$$((\Sigma_1, R_1, H_1, P_1) \setminus (\Sigma_2, R_2, H_2, P_2))^+ \rightarrow (\Sigma_1, R_1, H_1 \wedge notexists(\Sigma_2, R_2, H_2 \wedge \Sigma_1 = \Sigma_2, P_2 \cup \Sigma_1))^+$$
$$((\Sigma_1, R_1, H_1, P_1) \setminus (\Sigma_2, R_2, H_2, P_2))^? \rightarrow (\Sigma_1, R_1, H_1 \wedge notexists(\Sigma_2, R_2, H_2 \wedge \Sigma_1 = \Sigma_2, P_2 \cup \Sigma_1))^?$$

As the Relation R_1 and R_2 might be different :
For inter we should drop every row which contains NULL (special case with close world ?) for certain for possible ??
For union we ??

$$
\begin{aligned}
(H_1 \wedge H_2)^* &\rightarrow H_1^* \wedge H_2^* \\
(H_1 \vee H_2)^* &\rightarrow H_1^* \vee H_2^* \\
(r_i.a_i = c_i)^* &\rightarrow r_i.a_i = c_i \\
(r_i.a_i \neq c_i)^* &\rightarrow r_i.a_i \neq c_i \\
(r_i.a_i = r_j.a_j)^* &\rightarrow r_i.a_i = r_j.a_j \\
(r_i.a_i \neq r_j.a_j)^* &\rightarrow r_i.a_i \neq r_j.a_j \\
null(r_i.a_i)^* &\rightarrow null(r_i.a_i) \\
const(r_i.a_i)^* &\rightarrow const(r_i.a_i) \\
exists(Q)^* &\rightarrow exists(Q^+) \\
notexists(Q)^* &\rightarrow notexists(Q^?) \\
in(\Sigma_1, Q)^* &\rightarrow in(\Sigma, Q^+) \\
notin(\Sigma_1, (\Sigma, R, H, P))^* &\rightarrow notexists(\Sigma, R, H \wedge (\Sigma = \Sigma_1), P \cup \Sigma_1)^*
\end{aligned}
$$

$$
\begin{aligned}
(H_1 \wedge H_2)^{**} &\rightarrow H_1^{**} \wedge H_2^{**} \\
(H_1 \vee H_2)^{**} &\rightarrow H_1^{**} \vee H_2^{**} \\
(r_i.a_i = c_i)^{**} &\rightarrow r_i.a_i = c_i \vee null(r_i.a_i) \\
(r_i.a_i \neq c_i)^{**} &\rightarrow r_i.a_i \neq c_i \vee null(r_i.a_i) \\
(r_i.a_i = r_j.a_j)^{**} &\rightarrow r_i.a_i = r_j.a_j \vee null(r_i.a_i) \vee null(r_j.a_j) \\
(r_i.a_i \neq r_j.a_j)^{**} &\rightarrow r_i.a_i \neq r_j.a_j \vee null(r_i.a_i) \vee null(r_j.a_j) \\
null(r_i.a_i)^{**} &\rightarrow null(r_i.a_i) \\
const(r_i.a_i)^{**} &\rightarrow const(r_i.a_i) \\
exists(Q)^{**} &\rightarrow exists(Q^?) \\
notexists(Q)^{**} &\rightarrow notexists(Q^+) \\
in(r_i.a_i, Q)^{**} &\rightarrow in(r_i.a_i, Q^?) \\
notin(\Sigma_1, (\Sigma, R, H, P))^{**} &\rightarrow notexists(\Sigma, R, H \wedge (\Sigma = \Sigma_1), P \cup \Sigma_1)^{**}
\end{aligned}
$$

Proposition 2.

$$\forall Q \in [\![SQL]\!], Eval_{SQL}(Q^+, D) \subseteq cert_\perp(Q, D)$$

Proposition 3.

$$\forall Q \in [\![SQL]\!], posi_\perp(Q, D) \subseteq Eval_{SQL}(Q^?, D)$$

Proof. Assume (5).

By induction :

$$\begin{aligned}
Q &= (\Sigma, R, H_1 \wedge H_2, P) \\
x \in^n Eval_{SQL}(Q^+, D) &\Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q^+_*, D)(z) \geq n \\
Moreover & \\
Eval_{SQL}(Q^+_*, D)(z) = k &\Rightarrow Eval_{SQL}((*, R, H_1^* \wedge H_2^*, P), D)(z) = k \\
&\Rightarrow (Eval_{SQL}((*, R, H_1^*, P), D) \cap Eval_{SQL}((*, R, H_2^*, P), D))(z) = k \\
&\Rightarrow (Eval_{SQL}((*, R, H_1, P)^+, D) \cap Eval_{SQL}((*, R, H_2, P)^+, D))(z) = k \\
&\Rightarrow (Eval_{SQL}((*, R, H_1, P)^+, D))(x) \geq k \wedge (Eval_{SQL}((*, R, H_2, P)^+, D))(z) \geq k \\
&\Rightarrow (cert_\perp((*, R, H_1, P), D))(x) \geq k \wedge (cert_\perp((*, R, H_2, P), D))(z) \geq k \\
&\Rightarrow (cert_\perp((*, R, H_1, P), D) \cap cert_\perp((*, R, H_2, P), D))(z) \geq k \\
&\Rightarrow (cert_\perp((*, R, H_1 \wedge H_2, P), D))(z) \geq k \\
&\Rightarrow (cert_\perp(Q_*, D))(z) \geq k \\
Then & \\
x \in^n Eval_{SQL}(Q^+, D) &\Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} cert_\perp(Q_*, D)(z) \geq n \\
&\Rightarrow x \in^n cert_\perp(Q, D)
\end{aligned}$$

$$\begin{aligned}
Q &= (\Sigma, R, H_1 \vee H_2, P) \\
x \in^n Eval_{SQL}(Q^+, D) &\Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q^+_*, D)(z) \geq n \\
Moreover & \\
Eval_{SQL}(Q^+_*, D)(z) = k &\Rightarrow Eval_{SQL}((*, R, H_1^* \vee H_2^*, P), D)(z) = k \\
&\Rightarrow (Eval_{SQL}((*, R, H_1^*, P), D) \cup Eval_{SQL}((*, R, H_2^*, P), D))(z) = k \\
&\Rightarrow (Eval_{SQL}((*, R, H_1, P)^+, D) \cup Eval_{SQL}((*, R, H_2, P)^+, D))(z) = k \\
&\Rightarrow (Eval_{SQL}((*, R, H_1, P)^+, D))(x) \geq k \vee (Eval_{SQL}((*, R, H_2, P)^+, D))(z) \geq k \\
&\Rightarrow (cert_\perp((*, R, H_1, P), D))(x) \geq k \vee (cert_\perp((*, R, H_2, P), D))(z) \geq k \\
&\Rightarrow (cert_\perp((*, R, H_1, P), D) \cup cert_\perp((*, R, H_2, P), D))(z) \geq k \\
&\Rightarrow (cert_\perp((*, R, H_1 \vee H_2, P), D))(z) \geq k \\
&\Rightarrow (cert_\perp(Q_*, D))(z) \geq k \\
Then & \\
x \in^n Eval_{SQL}(Q^+, D) &\Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} cert_\perp(Q_*, D)(z) \geq n \\
&\Rightarrow x \in^n cert_\perp(Q, D)
\end{aligned}$$

$$
\begin{aligned}
&Q = (\Sigma, R, notexists(Q'), P)\\
&\quad x \in^n Eval_{SQL}(Q^+, D) \Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q^+_*, D)(z) \geq n\\
&\quad Moreover\\
&\quad Eval_{SQL}(Q^+_*, D)(z) = k \Rightarrow Eval_{SQL}((*, R, notexists(Q')^*, P), D)(z) = k\\
&\quad \Rightarrow Eval_{SQL}((*, R, notexists(Q'^?), P), D)(z) = k\\
&\quad \Rightarrow R(z) = k \wedge Eval_{SQL}(Q'[z]^?, D) = \emptyset\\
&\quad \Rightarrow R(z) = k \wedge posi_\perp(Q'[z], D) = \emptyset\\
&\quad \Rightarrow R(z) = k \wedge \forall h, \forall w, h(w) \notin Eval_{SQL}(Q'[h(z)], h(D))\\
&\quad \Rightarrow R(z) = k \wedge \forall h, Eval_{SQL}(Q'[h(z)], h(D)) = \emptyset\\
&\quad \Rightarrow R(z) = k \wedge \forall h, h(z) \in Eval_{SQL}((*, R, notexists(Q'), h(P)), h(D))\\
&\quad \Rightarrow cert_\perp(Q_*, D)(z) = k\\
&\quad Then\\
&\quad x \in^n Eval_{SQL}(Q^+, D) \Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} cert_\perp(Q_*, D)(z) \geq n\\
&\quad \Rightarrow x \in^n cert_\perp(Q, D)
\end{aligned}
$$

$$
\begin{aligned}
&Q = (\Sigma, R, r_i.a_i \neq p_i, P)\\
&\quad x \in^n Eval_{SQL}(Q^+, D) \Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q^+_*, D)(z) \geq n\\
&\quad Moreover\\
&Eval_{SQL}(Q^+_*, D)(z) = k \Rightarrow Eval_{SQL}((*, R, (r_i.a_i \neq p_i)^*, P), D)(z) = k\\
&\quad \Rightarrow Eval_{SQL}((*, R, r_i.a_i \neq p_i, P), D)(z) = k\\
&\quad \Rightarrow R(x) = k \wedge x[r_i.a_i] \neq P[p_i] \wedge x[r_i.a_i] \neq \perp \wedge P[p_i] \neq \perp\\
&\quad Moreover\\
&x[r_i.a_i] \neq \perp \wedge P[p_i] \neq \perp \Rightarrow \forall h, h(x)[r_i.a_i] = x[r_i.a_i] \wedge h(P)[p_i] = P[p_i]\\
&\quad Then\\
&Eval_{SQL}(Q^+_*, D)(z) = k \Rightarrow R(x) = k \wedge \forall h, h(x)[r_i.a_i] \neq h(P)[p_i] \wedge h(x)[r_i.a_i] \neq \perp \wedge h(P)[p_i] \neq \perp\\
&\quad \Rightarrow cert_\perp(Q_*, D)(z) = k\\
&\quad Then\\
&\quad x \in^n Eval_{SQL}(Q^+, D) \Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q^+_*, D) \wedge \sigma_\Sigma(y) = x\}} cert_\perp(Q_*, D)(z) \geq n\\
&\quad \Rightarrow x \in^n cert_\perp(Q, D)
\end{aligned}
$$

$\square$

Proof. Assume (4).
By induction ...

$$
\begin{aligned}
Q = (\Sigma, R, H_1 \wedge H_2, P) & \\
x \in^n posi_\perp(Q, D) &\Rightarrow R(x) \geq n \wedge \exists h, h(x) \in Eval_{SQL}(Q, h(P)) \\
&\Rightarrow \exists h, \sum_{z \in \{y | y \in Eval_{SQL}(Q_*, h(D)) \wedge \sigma_\Sigma(y) = h(x)\}} Eval_{SQL}(Q_*, h(D))(z) \geq n \\
Moreover & \\
Eval_{SQL}(Q_*, h(D))(z) = k &\Rightarrow Eval_{SQL}((*, R, H_1 \wedge H_2, h(P)), h(D))(z) = k \\
&\Rightarrow Eval_{SQL}((*, R, H_1, h(P)), h(D))(z) \geq k \wedge Eval_{SQL}((*, R, H_2, h(P)), h(D))(z) \geq \\
&\Rightarrow posi_\perp((*, R, H_1, P), D)(z) \geq k \wedge posi_\perp((*, R, H_2, P), D)(z) \geq k \\
&\Rightarrow Eval_{SQL}((*, R, H_1, P)^?, D)(z) \geq k \wedge Eval_{SQL}((*, R, H_2, P)^?, D)(z) \geq k \\
&\Rightarrow Eval_{SQL}((*, R, H_1^{**}, P), D)(z) \geq k \wedge Eval_{SQL}((*, R, H_2^{**}, P), D)(z) \geq k \\
&\Rightarrow Eval_{SQL}((*, R, H_1^{**} \wedge H_2^{**}, P), D)(z) \geq k \\
&\Rightarrow Eval_{SQL}((*, R, H_1 \wedge H_2, P)^?, D)(z) \geq k \\
&\Rightarrow Eval_{SQL}(Q_*^?, D)(z) \geq k \\
Then & \\
x \in^n posi_\perp(Q, D) &\Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q_*^?, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q_*^?, D)(z) \geq n \\
&\Rightarrow x \in^n Eval_{SQL}(Q^?, D)
\end{aligned}
$$

$$
\begin{aligned}
&Q = (\Sigma, R, notexists(Q'), P)\\
&\qquad x \in^n posi_\perp(Q, D) \Rightarrow \exists h, h(x) \in^n Eval_{SQL}(Q, h(P))\\
&\qquad\qquad \Rightarrow \exists h, \sum_{z \in \{y | y \in Eval_{SQL}(Q_*, h(D)) \wedge \sigma_\Sigma(y) = h(x)\}} Eval_{SQL}(Q_*, h(D))(z) \geq n\\
&\qquad Moreover\\
&\qquad Eval_{SQL}(Q_*, h(D))(z) = k \Rightarrow Eval_{SQL}((*, R, notexists(Q'), h(P)), h(D))(z) = k\\
&\qquad\qquad \Rightarrow R(z) = k \wedge Eval_{SQL}(Q'[z], h(D)) = \emptyset\\
&\qquad\qquad \Rightarrow R(z) = k \wedge [\![w^n | \forall g, g(w) \in^n Eval_{SQL}((Q'[g(z)]), g(D))]\!] = \emptyset \text{ A corriger}\\
&\qquad\qquad \Rightarrow R(z) = k \wedge cert_\perp(Q'[z], D) = \emptyset\\
&\qquad\qquad \Rightarrow R(z) = k \wedge Eval(Q'[z]^+, D) = \emptyset\\
&\qquad\qquad \Rightarrow Eval_{SQL}((*, R, notexists(Q'^+), P), D)(z) = k\\
&\qquad\qquad \Rightarrow Eval_{SQL}((*, R, notexists(Q')^{**}, P), D)(z) = k\\
&\qquad\qquad \Rightarrow Eval_{SQL}((*, R, notexists(Q'), P)^?, D)(z) = k\\
&\qquad Then\\
&\qquad x \in^n posi_\perp(Q, D) \Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q_*^?, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q_*^?, D)(z) \geq n\\
&\qquad\qquad \Rightarrow x \in^n Eval_{SQL}(Q^?, D)
\end{aligned}
$$

$$
\begin{aligned}
&Q = (\Sigma, R, r_i.a_i = r_j.a_j, P)\\
&\qquad x \in^n posi_\perp(Q, D) \Rightarrow R(x) = n \wedge \exists h, h(x) \in Eval_{SQL}(Q, h(P))\\
&\qquad\qquad \Rightarrow \exists h, \sum_{z \in \{y | y \in Eval_{SQL}(Q_*, hf(D)) \wedge \sigma_\Sigma(y) = h(x)\}} Eval_{SQL}(Q_*, h(D))(z) \geq n\\
&\qquad Moreover\\
&\qquad Eval_{SQL}(Q_*, h(D))(z) = k \Rightarrow Eval_{SQL}((*, R, r_i.a_i = r_j.a_j, h(P)), h(D))(z) = k\\
&\qquad\qquad \Rightarrow R(z) = k \wedge z[r_i.a_i] = z[r_j.a_j] \wedge z[r_i.a_i] \neq \perp \wedge z[r_j.a_j] \neq \perp\\
&\qquad\qquad \Rightarrow R(z) = k \wedge (z[r_i.a_i] = z[r_j.a_j] \wedge z[r_i.a_i] \neq \perp \wedge z[r_j.a_j] \neq \perp) \vee z[r_i.a_i] = \perp \vee z[r_j
\\
&\qquad Then\\
&\qquad x \in^n posi_\perp(Q, D) \Rightarrow \sum_{z \in \{y | y \in Eval_{SQL}(Q_*^?, D) \wedge \sigma_\Sigma(y) = x\}} Eval_{SQL}(Q_*^?, D)(z) \geq n\\
&\qquad\qquad \Rightarrow x \in^n Eval_{SQL}(Q^?, D)
\end{aligned}
$$

$\square$

4 Optimization

The translation $Q \to Q^+$ has an heavy cost has explained in the paper ..., in order to remove some useless null test we offer an optimization.

Definition 13. *For each Query and nested Query we maintain a set of attributes that have to be not null in order for the query to be true resp. false we denote this set* $\perp_Q^T$ *resp.*$\perp_Q^F$.

Definition 14.

$$
\begin{aligned}
\perp^T_{(\Sigma,R,H_1\wedge H_2,P)} &= \perp^T_{(\Sigma,R,H_1,P)} \cup \perp^T_{(\Sigma,R,H_2,P)} \\
\perp^T_{(\Sigma,R,H_1\vee H_2,P)} &= \perp^T_{(\Sigma,R,H_1,P)} \cap \perp^T_{(\Sigma,R,H_2,P)} \\
\perp^T_{(\Sigma,R,r_i.a_i=c_i,P)} &= \{r_i.a_i\} \\
\perp^T_{(\Sigma,R,r_i.a_i\neq c_i,P)} &= \{r_i.a_i\} \\
\perp^T_{(\Sigma,R,r_i.a_i=r_j.a_j,P)} &= \{r_i.a_i, r_j.a_j\} \\
\perp^T_{(\Sigma,R,r_i.a_i\neq r_j.a_j,P)} &= \{r_i.a_i, r_j.a_j\} \\
\perp^T_{(\Sigma,R,r_i.a_i=p_i,P)} &= \{r_i.a_i, p_i\} \\
\perp^T_{(\Sigma,R,r_i.a_i\neq p_i,P)} &= \{r_i.a_i, p_i\} \\
\perp^T_{(\Sigma,R,null(r_i.a_i),P)} &= \emptyset \\
\perp^T_{(\Sigma,R,const(r_i.a_i),P)} &= \{r_i.a_i\} \\
\perp^T_{(\Sigma,R,exists(Q),P)} &= \perp^T_{Q[?]} \\
\perp^T_{(\Sigma,R,notexists(Q),P)} &= \perp^F_{Q[?]}
\end{aligned}
$$

$$
\begin{aligned}
\perp^F_{(\Sigma,R,H_1\wedge H_2,P)} &= \perp^F_{(\Sigma,R,H_1,P)} \cap \perp^F_{(\Sigma,R,H_2,P)} \\
\perp^F_{(\Sigma,R,H_1\vee H_2,P)} &= \perp^F_{(\Sigma,R,H_1,P)} \cup \perp^F_{(\Sigma,R,H_2,P)} \\
\perp^F_{(\Sigma,R,r_i.a_i=c_i,P)} &= \emptyset \\
\perp^F_{(\Sigma,R,r_i.a_i\neq c_i,P)} &= \emptyset \\
\perp^F_{(\Sigma,R,r_i.a_i=r_j.a_j,P)} &= \emptyset \\
\perp^F_{(\Sigma,R,r_i.a_i\neq r_j.a_j,P)} &= \emptyset \\
\perp^F_{(\Sigma,R,r_i.a_i=p_i,P)} &= \emptyset \\
\perp^F_{(\Sigma,R,r_i.a_i\neq p_i,P)} &= \emptyset \\
\perp^F_{(\Sigma,R,null(r_i.a_i),P)} &= \emptyset \\
\perp^F_{(\Sigma,R,const(r_i.a_i),P)} &= \emptyset \\
\perp^F_{(\Sigma,R,exists(Q),P)} &= \perp^F_{Q[?]} \\
\perp^F_{(\Sigma,R,notexists(Q),P)} &= \perp^T_{Q[?]}
\end{aligned}
$$

Proposition 4.

$$x \in Eval_{SQL}(Q_*, D) \Rightarrow \forall r_i.a_i \in \perp^T_Q, x[r_i.a_i] \neq \perp$$

$$x \notin Eval_{SQL}(Q_*, D) \Rightarrow \forall r_i.a_i \in \perp^F_Q, x[r_i.a_i] \neq \perp$$

Proof. □

Definition 15.

$$
\begin{aligned}
nested^+(H_1 \wedge H_2) &= \{H_1 \wedge H_2\} \cup nested^+(H_1) \cup nested^+(H_2) \\
nested^+(H_1 \vee H_2) &= \{H_1 \vee H_2\} \cup nested^+(H_1) \cup nested^+(H_2) \\
nested^+(r_i.a_i = c_i) &= \{r_i.a_i = c_i\} \\
nested^+(r_i.a_i \neq c_i) &= \{r_i.a_i \neq c_i\} \\
nested^+(r_i.a_i = r_j.a_j) &= \{r_i.a_i = r_j.a_j\} \\
nested^+(null(r_i.a_i)) &= \{null(r_i.a_i)\} \\
nested^+(const(r_i.a_i)) &= \{const(r_i.a_i)\} \\
nested^+(exists(Q)) &= \{exists(Q)\} \cup nested^+(Q) \\
nested^+(notexists(Q)) &= \{notexists(Q)\} \cup nested^-(Q)
\end{aligned}
$$

$$\begin{aligned}
nested^-(H_1 \wedge H_2) &= nested^-(H_1) \cup nested^-(H_2)\\
nested^-(H_1 \vee H_2) &= nested^-(H_1) \cup nested^-(H_2)\\
nested^-(r_i.a_i = c_i) &= \emptyset\\
nested^-(r_i.a_i \neq c_i) &= \emptyset\\
nested^-(r_i.a_i = r_j.a_j) &= \emptyset\\
nested^-(null(r_i.a_i)) &= \emptyset\\
nested^-(const(r_i.a_i)) &= \emptyset\\
nested^-(exists(Q)) &= nested^-(Q)\\
nested^-(notexists(Q)) &= nested^+(Q)
\end{aligned}$$

$$nested(Q) = nested^-(Q) \cup nested^+(Q)$$

Definition 16.

$$\begin{aligned}
notexists(Q')^{\perp}_{Q} &\rightarrow notexists(Q'^{\perp}_{Q})\\
exists(Q')^{\perp}_{Q} &\rightarrow exists(Q'^{\perp}_{Q})
\end{aligned}$$

Definition 17.

$$\begin{aligned}
&(\Sigma, R, H \vee null(r_i.a_i), P)^{\perp}_{Q} \rightarrow (\Sigma, R, H^{\perp}_{Q}, P)\\
&\qquad \textit{if } \exists Q' \in nested^+(Q), (H \vee null(r_i.a_i)) \in nested(Q'), r_i.a_i \in \perp^{T}_{Q'}\\
&\qquad \textit{if } \exists Q' \in nested^-(Q), (H \vee null(r_i.a_i)) \in nested(Q'), r_i.a_i \in \perp^{F}_{Q'}
\end{aligned}$$

Proposition 5.

$$Eval_{SQL}(Q, D) = Eval_{SQL}(Q^{\perp}_{Q}, D)$$

Proof.

Definition 18.

$$\begin{aligned}
&(\Sigma, R, H \vee null(r_i.a_i), P)^{\perp^F}_{Q} \rightarrow (\Sigma, R, H^{\perp^F}_{Q}, P)\\
&\qquad \textit{if } \exists Q' \in nested^-(Q), (H \vee null(r_i.a_i)) \in nested(Q'), r_i.a_i \in \perp^{T}_{Q'}\\
&\qquad \textit{if } \exists Q' \in nested^+(Q), (H \vee null(r_i.a_i)) \in nested(Q'), r_i.a_i \in \perp^{F}_{Q'}
\end{aligned}$$

Proposition 6.

$$\forall Q' \in nested^+(Q), Eval(Q, D) = Eval(Q^{\perp}_{Q'}, D)$$

Proposition 7.

$$\forall Q' \in nested^-(Q), Eval(Q, D) = Eval(Q^{\perp^F}_{Q'}, D)$$

Proof. Assume (10).
By Induction ...
$H_1 \wedge H_2 \in nested^+(Q)$
$\exists r_i.a_i \in (\perp^T_{H_1} \setminus \perp^T_{H_2}) \wedge (H \vee null(r_i.a_i) \in nested(H_2)$
Then if $x[r_i.a_i] = \perp$, $Eval(H^T_{2,H_2}, D, x)$ might be different from $Eval(H^T_{2,H_1 \wedge H_2}, D, x)$ □

□