Introduction to Databases
Translations between RA and RC

Dr Paolo Guagliardo

(v26.0)

Problem 1. Consider the schema with tables Customer over attributes *ID*, *Name*, *City* (in this order) and Account over attributes *Number*, *Branch*, *CustID*, *Balance* (in this order). Express the following relational algebra query in relational calculus:

$$\textsc{Customer} \bowtie \big(\pi_{\text{ID,City}}(\textsc{Customer}) \cap \rho_{\text{CustID}\to\text{ID, Branch}\to\text{City}}\big(\pi_{\text{Branch,CustID}}(\textsc{Account})\big)\big) \tag{1}$$

Use the translation rules from RA to RC we have seen in class.

Solution. We first need to (equivalently) rewrite the given RA expression into one that uses only primitive operations, for which we have translation rules into RC. This would be the standard procedure to follow[1] but, as a matter of fact, by first rewriting $\cap$ in terms of difference, then translating into RC and finally applying some first-order logic equivalences, we can obtain a direct translation rule for $\cap$, which we can then reuse in other translations from RA to RC.

To this end, let us consider the RA expression $e \cap e'$, in which e and e' are arbitrary RA expressions (over the same set of attributes). From the lectures, we know that $e \cap e' \equiv e - (e - e')$. By applying the translation rule for difference (twice), we get $\varphi \wedge \neg(\varphi \wedge \neg\varphi')$, where φ is the translation of e and φ' is the translation of e'. We now make use of some FO equivalences:

$$\begin{aligned}
\varphi \wedge \neg(\varphi \wedge \neg\varphi') &\equiv \varphi \wedge (\neg\varphi \vee \neg\neg\varphi') && [\text{by DeMorgan's law for } \wedge] \\
&\equiv \varphi \wedge (\neg\varphi \vee \varphi') && [\text{by double negation elimination}] \\
&\equiv (\varphi \wedge \neg\varphi) \vee (\varphi \wedge \varphi') && [\text{by distributivity of } \wedge \text{ over } \vee] \\
&\equiv \varphi \wedge \varphi' && [\text{because } \varphi \wedge \neg\varphi \text{ is false}]
\end{aligned}$$

Therefore, we obtain the following rule for translating $e \cap e'$ into RC:

1. Translate the expression e into a formula φ,
2. Translate the expression e' into a formula φ',
3. Translate $e \cap e'$ into $\varphi \wedge \varphi'$.

[1] You must do this for all other derived operations except for intersection.

This rule is the same as the one for Cartesian product, but there is one major difference: for intersection the formulas φ and φ' of which we take the conjunction have the same free variables, whereas in the case of Cartesian product they have no free variables in common.

Let us now go back to the translation of (1). By rewriting the natural join we obtain:

$$\pi_{\text{ID, Name, City}}\Big(\sigma_{\text{ID}=\text{ID}' \wedge \text{City}=\text{City}'}\big(\underbrace{\rho_{\text{ID}\to\text{ID}', \text{City}\to\text{City}'}(\textsc{Customer})}_{E_1} \times (\underbrace{\pi_{\text{ID, City}}(\textsc{Customer}) \cap E}_{E_2})\big)\Big)$$

where $E = \rho_{\text{CustID}\to\text{ID, Branch}\to\text{City}}\big(\pi_{\text{Branch,CustID}}(\textsc{Account})\big)$.
Then we associate each attribute in the schema with a unique variable name:

$$\eta = \{\, \text{ID} \mapsto x_1, \text{Name} \mapsto x_2, \text{City} \mapsto x_3, \text{Number} \mapsto y_1, \text{Branch} \mapsto y_2, \text{CustID} \mapsto y_3, \text{Balance} \mapsto y_4 \,\}$$

We start the translation from the innermost expressions:

- The translation of $\textsc{Customer}$ is $\textsc{Customer}(x_1, x_2, x_3)$
- The translation of $\textsc{Account}$ is $\textsc{Account}(y_1, y_2, y_3, y_4)$

To translate E_1 we first map ID' and City' to new, arbitrarily chosen[2] variables x'_1 and x'_3, respectively, because the environment η does not have bindings for these attributes:

$$\eta := \eta \cup \{\, \text{ID}' \mapsto x'_1, \text{City}' \mapsto x'_3 \,\}$$

Then, since there are no occurrences of x'_1 and x'_3 already present in the translation of $\textsc{Customer}$,[3] we can simply replace x_1 by x'_1 and x_3 by x'_3:

$$\textsc{Customer}(x'_1, x_2, x'_3) \tag{φ_1}$$

The translation of $\pi_{\text{Branch, CustID}}(\textsc{Account})$ is $\exists y_1, y_4\ \textsc{Account}(y_1, y_2, y_3, y_4)$ and, by renaming y_3 to x_1 and y_2 to x_3, we obtain the translation of E, which is

$$\exists y_1, y_4\ \textsc{Account}(y_1, x_3, x_1, y_4) \tag{φ}$$

The translation of $\pi_{\text{ID,City}}(\textsc{Customer})$ is $\exists x_2\ \textsc{Customer}(x_1, x_2, x_3)$ and, by taking the conjunction with the translation of E (according to the direct rule for $\cap$ we proved earlier), we get the translation of E_2:

$$\big(\exists x_2\ \textsc{Customer}(x_1, x_2, x_3)\big) \wedge \underbrace{\big(\exists y_1, y_4\ \textsc{Account}(y_1, x_3, x_1, y_4)\big)}_{\varphi} \tag{φ_2}$$

To translate the selection, we replace the attributes in the condition by their corresponding variables and we take the conjunction with the translation of $E_1 \times E_2$, which is $\varphi_1 \wedge \varphi_2$, obtaining:

$$\varphi_1 \wedge \varphi_2 \wedge (x_1 = x'_1 \wedge x_3 = x'_3)$$

Finally, to translate the outermost projection, we existentially quantify variables x'_1 and x'_3 in the above formula. The final translation (where unnecessary parentheses are removed) is:

$$\exists x'_1, x'_3\ \textsc{Customer}(x'_1, x_2, x'_3) \wedge \exists x_2\ \textsc{Customer}(x_1, x_2, x_3) \\ \wedge \exists y_1, y_4\ \textsc{Account}(y_1, x_3, x_1, y_4) \wedge x_1 = x'_1 \wedge x_3 = x'_3$$

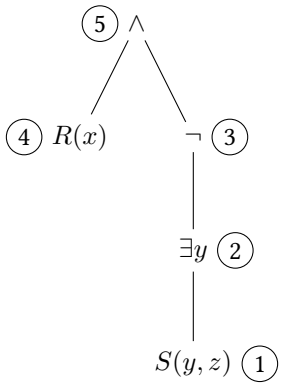
[2] As long as they have not been used already.

[3] If there were such occurrences, say of x'_1, then we would have to replace all of them by a fresh new variable, say w (or whichever other name was not, and will not, be used anywhere else).

Problem 2. Given a relation R over attribute A, and a relation S over attributes A, B (in this order), translate the following relation calculus query to relational algebra:

$$\{\, x, z \mid R(x) \wedge \neg \exists y\ S(y, z) \,\}$$

Solution. The syntax tree of the (body of the) given relational calculus query is:



Under the mapping $\eta = \{x \mapsto A,\ y \mapsto B,\ z \mapsto C\}$, we get the following translations of each subexpression (corresponding to the subtrees above labelled with circled numbers):

$(1) \mapsto \rho_{B \to C,\, A \to B}(S)$

$(2) \mapsto \pi_C\left((1)\right)$

$(3) \mapsto \mathsf{Adom}_C - (2)$

$(4) \mapsto R$

$(5) \mapsto \left((4) \times \mathsf{Adom}_C\right) \cap \left((3) \times \mathsf{Adom}_A\right)$

Putting it all together, we obtain the following:

$$(R \times \mathsf{Adom}_C) \cap \Big(\mathsf{Adom}_A \times \big(\mathsf{Adom}_C - \pi_C\big(\rho_{B \to C,\, A \to B}(S)\big)\big)\Big)$$

Problem 3. Given two relations R and S, each over attributes A, B (in this order), express the following relational calculus query in relational algebra:

$$\big\{x \mid \neg\big(\forall y\ R(x, y) \to S(x, y)\big) \wedge \neg\big(\exists z\ S(x, z) \wedge R(z, x)\big)\big\}$$

Use only the translation rules from RC to RA we have seen in class.

Solution. We first need to put the formula in the following form:

$$\underbrace{\big(\exists y\ R(x, y) \wedge \neg S(x, y)\big)}_{\varphi_1} \wedge \neg\ \underbrace{\big(\exists z\ S(x, z) \wedge R(z, x)\big)}_{\varphi_2}$$

Then we associate each variable with a distinct attribute name:

$$\{x \mapsto A, y \mapsto B, z \mapsto C\}$$

The translation of $R(x, y)$ is simply R and the translation of $S(x, y)$ is S. The translation of $\neg S(x, y)$ is $\mathsf{Adom}_A \times \mathsf{Adom}_B - S$. As $R(x, y)$ and $\neg S(x, y)$ have the same free variables, $R(x, y) \wedge \neg S(x, y)$ translates to $R \cap (\mathsf{Adom}_A \times \mathsf{Adom}_B - S)$. In turn, the translation of φ_1 is

$$\pi_A\big(R \cap (\mathsf{Adom}_A \times \mathsf{Adom}_B - S)\big) \tag{E_1}$$

The translation of $S(x, z)$ is $\rho_{B\to C}(S)$ and the translation of $R(z, x)$ is $\rho_{A\to C,\, B\to A}(R)$. As $S(x, z)$ and $R(z, x)$ have the same free variables, $S(x, z) \wedge R(z, x)$ translates to $\rho_{B\to C}(S) \cap \rho_{A\to C,\, B\to A}(R)$ and, in turn, the translation of φ_2 is

$$\pi_A\big(\rho_{B\to C}(S) \cap \rho_{A\to C,\, B\to A}(R)\big) \tag{E_2}$$

The translation of $\neg\varphi_2$ is $\mathsf{Adom}_A - E_2$. As φ_1 and φ_2 have the same free variables, the final translation is $E_1 \cap (\mathsf{Adom}_A - E_2)$, which fully expanded gives us the following:

$$\pi_A\big(R \cap (\mathsf{Adom}_A \times \mathsf{Adom}_B - S)\big) \cap \Big(\mathsf{Adom}_A - \pi_A\big(\rho_{B\to C}(S) \cap \rho_{A\to C,\, B\to A}(R)\big)\Big)$$

Problem 4.

(a) Can we simplify the relational algebra expression obtained in Problem 3 into an equivalent expression that does not mention the active domain? If yes, give such an expression. Otherwise, explain why this is the case.

(b) How would you translate the relational calculus query of Problem 3 if the output tuple (i.e., the head of the query) were x, x rather than x?

Solution.

(a) Yes. Let E be the final expression obtained in Problem 3. The relations R and S (which are over attributes A, B) are both trivially subsets of $\mathsf{Adom}_A \times \mathsf{Adom}_B$. Then, $R \cap \big((\mathsf{Adom}_A \times \mathsf{Adom}_B) - S\big)$ is precisely $R - S$. So E can be simplified to

$$\underbrace{\pi_A(R - S)}_{E'} \cap \Big(\mathsf{Adom}_A - \underbrace{\pi_A\big(\rho_{B\to C}(S) \cap \rho_{A\to C,\, B\to A}(R)\big)}_{E''}\Big)$$

With a similar reasoning, since both E' and E'' (which are over attribute A) are subsets of Adom_A, we can simplify $E' \cap (\mathsf{Adom}_A - E'')$ to $E' - E''$. Thus, E is equivalent to

$$\pi_A(R - S) - \pi_A\big(\rho_{B\to C}(S) \cap \rho_{A\to C,\, B\to A}(R)\big)$$

(b) In general, if you have a relational calculus query with repetitions in the head, say $\{x, y, y, x, z, x \mid \psi\}$, it suffices to translate the body of the following equivalent query:

$$\big\{x, y, y', x', z, x'' \mid x = x' \wedge x = x'' \wedge y = y' \wedge \psi\big\}$$

For $\{x, x \mid \varphi\}$, where φ is the formula in the original query in Problem 3, we would have $\{x, x' \mid x = x' \wedge \varphi\}$. The translation of φ is the RA expression obtained in Problem 3, let us call it E, so if we map x' to attribute A', we obtain:

$$\sigma_{A=A'}(\mathsf{Adom}_A \times \mathsf{Adom}_{A'}) \cap (E \times \mathsf{Adom}_{A'})$$

The above expression is equivalent to $\sigma_{A=A'}\big(E \bowtie \rho_{A\to A'}(E)\big)$, but you are not required to rewrite it into this form.