1

Grading scheme: 1 for a perfectly correct answer and 0 otherwise.

1.3

Note that it is incorrect to *only* return tuples where both the entity and the value are identical. Consider the relation: $\{(a, b), (b, a)\}$. The answer to the query should be 2 but if one only equates the first and the second attribute the returned answer is 0.

2

Grading scheme: 1 for a perfectly correct answer, 0.5 if the view is correctly computed but the view has extraneous or missing attributes, 0 otherwise.

3

Grading scheme: 1 for a perfectly correct answer, 0.5 if the Relational Calculus expression is correct but the tranlation is incorrect, 0 otherwise.

3.1

There are two possible ways to interpret this question. We accept both interpretations as valid.

Interpretation 1

$$c : \exists c_2, s, s_2.CompetesWith(c, c_2) \wedge CompanyEconomicSector(c, s)$$
$$\wedge CompanyEconomicSector(c_2, s_2) \wedge \neg(s = s_2)$$

For brevity, we use `CES` for $CompanyEconomicSector$ and `CW` for $CompetesWith$. The corresponding Relational Algebra expression is:

$$\Pi_c \sigma_{s<>s_2} \left((\rho_{e/c,v/c_2}(CW) \bowtie \rho_{e/c,v/s}(CES)) \bowtie \rho_{e/c_2,v/s_2} CES \right)$$

Interpretation 2

$$c : \exists c_2, s_2.CompetesWith(c, c_2) \wedge \neg CompanyEconomicSector(c, s_2)$$
$$\wedge CompanyEconomicSector(c_2, s_2)$$

$$\Pi_c \left(\Pi_{c,s_2}(\rho_{e/c_2,v/s_2}(\texttt{CES}) \bowtie \rho_{e/c,v/c_2}(\texttt{CW})) - \rho_{e/c,v/s_2}(\texttt{CES}) \right)$$

Differences

The two cases provide different answers. Consider the database where c_2 competes with c_1 and the $CompanyEconomicSector$ looks like $\{(c_1, s_1), (c_2, s_1), (c_2, s_2)\}$.

The first interpretation yields c_2 as an answer while the second one doesn't.

3.2

$$p : \exists c.ProducesProduct(c, p) \wedge \neg\exists c_2.CompetesWith(c, c_2)$$

$$\Pi_v(\texttt{PP} \bowtie (\Pi_e \texttt{PP} - \Pi_e \texttt{CW}))$$

3.3

$$p : \exists c.ProducesProduct(c, p) \wedge HeadquarteredIn(c, `concept : city : san\_jose')$$

$$\Pi_v(\texttt{PP} \bowtie (\Pi_e \sigma_{v=`concept:city:san\_jose'} \texttt{HQ}))$$

3.4

$$p : \forall c.ProducesProduct(c, p) \implies (\forall c_2.CompetesWith(c, c_2) \implies ProducesProduct(c_2, p))$$
$$i.e. \quad p : \neg\exists c, c_2.ProducesProduct(c, p) \wedge CompetesWith(c, c_2) \wedge \neg ProducesProduct(c_2, p)$$

For brevity, we use PP for $ProducesProduct$ and CW for $CompetesWith$. Translating to DataLog:

$$\begin{aligned} Q(p) &:- \neg Q_1(p) \\ Q_1(p) &:- \texttt{PP}(c, p), \texttt{CW}(c, c_2), \neg\texttt{PP}(c_2, p) \end{aligned}$$

In terms of Relational Algebra $Q_1(p)$ translates to the following:

$$\Pi_p \left(\Pi_{c_2,p}(\rho_{e/c,v/p}(\texttt{PP}) \bowtie \rho_{e/c,v/c_2}(\texttt{CW})) - \rho_{e/c_2,v/p}(\texttt{PP})\right)$$

Thus, $Q(p)$ translates to

$$\Pi_p(\rho_{v/p}(\texttt{PP})) - \Pi_p \left(\Pi_{c_2,p}(\rho_{e/c,v/p}(\texttt{PP}) \bowtie \rho_{e/c,v/c_2}(\texttt{CW})) - \rho_{e/c_2,v/p}(\texttt{PP})\right)$$

3.5

$$p : \forall c.ProducesProduct(c, p) \implies (\exists c_2.CompetesWith(c, c_2) \wedge ProducesProduct(c_2, p))$$
$$i.e. \quad p : \neg\left(\exists c ProducesProduct(c, p) \wedge \neg\exists.c_2(CompetesWith(c, c_2) \wedge ProducesProduct(c_2, p))\right)$$

For brevity, we use PP for $ProducesProduct$ and CW for $CompetesWith$. Translating to DataLog:

$$\begin{aligned} Q(p) &:- \neg Q_1(p) \\ Q_1(p) &:- \texttt{PP}(c, p), \neg\texttt{CW}(c, c_2), \neg\texttt{PP}(c_2, p) \end{aligned}$$

In terms of Relational Algebra $Q_1(p)$ translates to the following:

$$\Pi_p \left(\Pi_{c,p}(\rho_{e/c,v/p}(\texttt{PP})) - \Pi_{c,p} \left(\rho_{e/c,v/c_2}(\texttt{CW}) \bowtie \rho_{e/c_2,v/p}(\texttt{PP})\right)\right)$$

Thus, $Q(p)$ translates to

$$\Pi_p(\rho_{v/p}(\texttt{PP})) - \Pi_p \left(\Pi_{c,p}(\rho_{e/c,v/p}(\texttt{PP})) - \Pi_{c,p} \left(\rho_{e/c,v/c_2}(\texttt{CW}) \bowtie \rho_{e/c_2,v/p}(\texttt{PP})\right)\right)$$

4

Grading scheme: For each subpart: 1 for a perfectly correct answer and 0 otherwise.

5

Grading scheme: For each subpart: 1 for a perfectly correct answer and 0 otherwise.

6

Grading scheme: For each subpart: 1 for the correct answer, 0.5 for incorrectly propagating probabilities, and 0 for not treating all tuples as probabilistic.

The main note for this section is that *each* tuple that is used in the answer has to be treated with a probability of being true (or false). For example, in 6.2 it is incorrect to remove (from the set of all companies) the companies with competitors using the view $CompetesWith$ and then report the probability of a product being manufactured by the non-competing companies. One has to include (with the associated probabilities) those companies that have a tuple with a competitor.

6.1

One can get the correct output by averaging probabilities for this question. But such an answer is theoretically unsound. It works with our dataset since for this answer, there are not more than one tuple to OR.

6.2

One must also consider companies that have no competitors with probability 1. To do so, one has to compute the companies that don't find a mention in the $CompetesWith$ view.