Data Models and Query Languages  
Summerterm 2013

2. Exercise Sheet: Chase & Datalog

Discussion: 14.05.2013

Exercise 1 (Constraints in First-Order Logic)
Consider the following database schema.

\[
\begin{align*}
\text{hasAirport}(c.id) \\
\text{fly}(c.id1, c.id2, \text{dist}) \\
\text{rail}(c.id1, c.id2, \text{dist})
\end{align*}
\]

Specify the constraints below in First-order Logic and indicate if your specification is a tuple-generating dependency, equality-generating dependency, or none of both. In case of tuple-generating or equality-generating dependencies additionally give their \textit{body} and \textit{head}.

a) \(a_1\): If a city has an airport, then there is at least one flight departing from this city.

b) \(a_2\): The distance of a rail connection functionally depends on the departure and destination station, i.e. there is only one unique distance for every rail connection.

c) \(a_3\): There is at least one flight and one train connection listed in the database.

d) \(a_4\): Starting from Frankfurt, all cities with an airport can be reached either by direct flight or by a flight with only one intermediate stop.

e) \(a_5\): All pairs of cities with an airport that have a direct train connection also have a direct flight connection.

Exercise 2 (Chase Application)
Consider the schema from Exercise 1, the constraint set \(\Sigma := \{a_1, a_2, a_3\}\) with

\[
\begin{align*}
a_1 := \forall c_1, c_2, c_3, d_1, d_2 (\text{rail}(c_1, c_2, d_1), \text{rail}(c_2, c_3, d_2) \rightarrow \exists d_3 \text{ rail}(c_1, c_3, d_3)) \\
a_2 := \forall c_1, c_2, d_1, d_2 (\text{fly}(c_1, c_2, d_1) \land \text{fly}(c_2, c_1, d_2) \rightarrow d_1 = d_2) \\
a_3 := \forall c_1, c_2, d_1 (\text{fly}(c_1, c_2, d_1) \rightarrow \exists d_2 \text{ fly}(c_2, c_1, d_2))
\end{align*}
\]

and the Conjunctive Query

\[
\begin{align*}
Q: \quad \text{ans}(C_3) \leftarrow \text{rail}(Freiburg, C_1, D_1), \text{rail}(C_1, C_2, D_2), \text{fly}(C_2, C_3, D_3).
\end{align*}
\]
a) Describe the semantics of the constraints and the query informally.
b) Which constraints from \( \Sigma \) are satisfied by \( \text{body}(Q) \)? Does \( \text{body}(Q) \) satisfy \( \Sigma \)?
c) Chase query \( Q \) with \( \Sigma \). Provide all intermediate results (= chase steps). Does it hold that \( \text{body}(Q^{\Sigma}) \models \Sigma \)?

Exercise 3 (Datalog, Transitive Closure)
Given a directed graph \( G \) with edge relation \( E(a, b) \), which means there is an edge from \( a \) to \( b \) in \( G \).

a) Give three different Datalog\(^+\) programs which compute the transitive closure of \( G \).
b) Let \( k \) be the length of the longest path in \( G \). Determine the number of iterations which is needed for each version to compute the transitive closure.
c) Apply the naive algorithm evaluation of the three programs on the database:
\[
E(1, 2), E(2, 3), E(3, 4), E(4, 5)
\]

Exercise 4 (Datalog, Equivalence)
Consider the following two Datalog\(^+\) programs

\( \Pi_1: \)
\[
\text{Buys}(X,Y) \leftarrow \text{Likes}(X,Y) \\
\text{Buys}(X,Y) \leftarrow \text{Knows}(X,Z), \text{Buys}(Z,Y)
\]

\( \Pi_2: \)
\[
\text{Buys}(X,Y) \leftarrow \text{Likes}(X,Y) \\
\text{Buys}(X,Y) \leftarrow \text{Knows}(X,Z), \text{Likes}(Z,Y)
\]

Prove or falsify that \( \Pi_1 \equiv \Pi_2 \).

Exercise 5 (Datalog, Shortest Paths)
Given a directed acyclic graph \( G \) with edge relation \( E(a, b) \), which means there is an edge from \( a \) to \( b \) in \( G \).

Give a stratified Datalog\(^+\) program which computes the shortest paths in \( G \) wrt. the number of edges.

Exercise 6 (Datalog, 3-Coloring of Graphs)
Let \( G = (V,E) \) be a graph with \( V = \{1, 2, 3, 4, 5\} \) and \( E = \{(1, 2), (1, 4), (1, 5), (2, 3), (2, 4), (3, 4), (4, 5)\} \).

Consider the following Datalog\(^+\) program

\( \Pi: \)
\[
\text{Color}(N, \text{blue}) \leftarrow \text{V}(N), \text{not Color}(N, \text{green}), \text{not Color}(N, \text{red}) \\
\text{Color}(N, \text{green}) \leftarrow \text{V}(N), \text{not Color}(N, \text{blue}), \text{not Color}(N, \text{red}) \\
\text{Color}(N, \text{red}) \leftarrow \text{V}(N), \text{not Color}(N, \text{green}), \text{not Color}(N, \text{blue}) \\
\text{NonColoring}(N) \leftarrow E(N,M), \text{Color}(N, C), \text{Color}(M, C)
\]

Give a stable model of \( \Pi \) where \( \text{NonColoring} \) is empty, i.e. demonstrate that the given graph is 3-colorable.