Universität Freiburg
Institut für Informatik
Michael Meier
Fang Wei

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg
Tel. (0761) 203-8126
Tel. (0761) 203-8125

Webbasierte Informationssysteme
Winter Term 2010/2011
December 8, 2010

Seventh Exercise Set: SPARQL

Exercise 1
We write $P_1 \equiv P_2$ for two SPARQL queries P_1, P_2 if and only if P_1 and P_2 yield the same result on every possible RDF document.

Let A, B, and C be SPARQL graph patterns, i.e. they are composed of BGPs, Union, Join and Leftjoin.

Note that we use $\textsc{Leftjoin}(A, B)$ as an abbreviation for $\textsc{Leftjoin}(A, B, true)$. Assume that filter conditions are built from variables, URIs, literals, the bound operator, equality and the logical connectives $!, \wedge, \vee$

Let A be a SPARQL graph pattern. Function $cVars(A)$ extracts the set of so-called *certain variables* and is recursively defined as

$$
\begin{aligned}
cVars(BGP(T)) &:= vars(T), \text{ where } vars(\mathrm{T}) \text{ is defined as the set of variables in T} \\
cVars(Join(A_1, A_2)) &:= cVars(A_1) \cup cVars(A_2) \\
cVars(Union(A_1, A_2)) &:= cVars(A_1) \cap cVars(A_2) \\
cVars(Minus(A_1, A_2) &:= cVars(A_1) \\
cVars(Leftjoin(A_1, A_2)) &:= cVars(Join(A_1, A_2)) \cap cVars(Minus(A_1, A_2)) \\
cVars(Filter(A_1, F)) &:= cVars(A_1)
\end{aligned}
$$

Function $pVars(A)$ extracts the set of so-called *possible variables* and is recursively defined as

$$
\begin{aligned}
pVars(BGP(T)) &:= vars(T), \text{ where } vars(\mathrm{T}) \text{ is defined as the set of variables in T} \\
pVars(Join(A_1, A_2)) &:= pVars(A_1) \cup pVars(A_2) \\
pVars(Union(A_1, A_2)) &:= pVars(A_1) \cup pVars(A_2) \\
pVars(Minus(A_1, A_2) &:= pVars(A_1) \\
pVars(Leftjoin(A_1, A_2)) &:= pVars(Join(A_1, A_2)) \cup pVars(Minus(A_1, A_2)) \\
pVars(Filter(A_1, F)) &:= pVars(A_1)
\end{aligned}
$$

We define the class $\widetilde{A}$ of SPARQL graph patterns recursively as follows. A SPARQL graph pattern $\widetilde{A}$ is contained in $\widetilde{A}$ iff

- $\widetilde{A} := BGP(T)$ is a basic graph pattern,
- $\widetilde{A} := Join(\widetilde{A_1}, \widetilde{A_2})$, where $\widetilde{A_1}$ and $\widetilde{A_2}$ are $\widetilde{A}$ expressions,
- $\widetilde{A} := Leftjoin(\widetilde{A_1}, \widetilde{A_2})$, where $\widetilde{A_1}$ and $\widetilde{A_2}$ are $\widetilde{A}$ expressions,
- $\widetilde{A} := Filter(\widetilde{A_1}, F)$, where F is a filter condition and $\widetilde{A_1} \in \widetilde{A}$,

- $\widetilde{A} := Union(\widetilde{A_1}, \widetilde{A_2})$, where $\widetilde{A_1}$ $\widetilde{A_2}$ are $\widetilde{A}$ expressions and $pVars(\widetilde{A_1})$=$cVars(\widetilde{A_1})$=$pVars(\widetilde{A_2})$=$cVars(\widetilde{A_2})$.

Let A be a SPARQL graph pattern and let Ω_A denote the mapping set obtained when evaluating A on any document D. Prove the following statements:

- If $?x \in cVars(A)$ then $\forall \mu \in \Omega_A : ?x \in dom(\mu)$.
- For all $\mu \in \Omega_A : ?x \in dom(\mu)$ then $?x \in pVars(A)$.
- Describe in your own words the intuition of certain and possible variables. Hint: look closely at the first two bullets of this exercise.
- If $pVars(A) = cVars(A)$ then $\forall \mu_1, \mu_2 \in \Omega_A : dom(\mu_1) = dom(\mu_2)$.
- Assume that $A \in \widetilde{A}$. Show that every two mappings $\mu_1, \mu_2 \in \Omega_A$ such that $\mu_1 \neq \mu_2$ it holds that they are incompatible.

Exercise 2
Let A, B, and C be SPARQL graph patterns from $\widetilde{A}$ as defined in the previous exercise. Prove that the following equivalences holds.

a) $\textsc{Leftjoin}(A, A) \equiv A$
b) $\textsc{Join}(A, A) \equiv A$
c) $\textsc{Leftjoin}(A, B) \equiv \textsc{Leftjoin}(A, \textsc{Join}(A, B))$
d) Let $?x \in cVars(B) \backslash pVars(A)$. $\textsc{Filter}(\textsc{Leftjoin}(A, B), bound(?x)) \equiv \textsc{Join}(A, B)$
e) Let for all variables $?x$ in F be either $?x \in cVars(A)$ or $?x \notin pVars(B)$. $\textsc{Filter}(\textsc{Join}(A, B), F) \equiv \textsc{Join}(\textsc{Filter}(A, F), B)$
f) Let for all variables $?x$ in F be either $?x \in cVars(A)$ or $?x \notin pVars(B)$. $\textsc{Filter}(\textsc{Leftjoin}(A, B), F) \equiv \textsc{Leftjoin}(\textsc{Filter}(A, F), B)$

Due by: December 15, 2010 before the tutorial starts.