Exam 2015 – Web Data Management

S. Abiteboul & P. Senellart

February 2015

The exam is 2 hours. All documents are allowed. Internet access and communication devices are disallowed.

1 Mapreduce and Datalog (7 points)

1. (1.5 points) Assume n peers $p_1, \ldots, p_n$ each contain a relation Parent@p_i (Parent(x, y) means that the person x is a parent of the person y). We want to compute, for each person, his or her number of children using MapReduce. Explain in a few lines of pseudo-code how one would achieve this.

   The transitive closure of the union of two binary relations $G_1@p$ and $G_2@q$ with attributes AB can be computed by using auxiliary relations $G@p, S@p, T@p$ with attributes AB in the following manner:

```
S@p := set();
G@p := G_1@p union G_2@q;
T@p := G@p;
while S@p != T@p do
     { S@p := T@p;
       T@p := S@p union select A: u.A, B : v.B
                          from u in S@p, v in G@p
                         where u.B = v.A; }
```

2. (1.5 points) Using the relations ParentOf@p_i, write a program in the same fashion that computes the set of all pairs (a, b) such that a and b have a common ancestor and are of the same generation with respect to this ancestor. (Two persons are of the same generation if they have a common parent or if their parents are of the same generation.)

3. (1 point) Explain why the program above for the transitive closure of the union is very inefficient (at most three lines).

4. (1.5 point) Describe a possible optimization (at most five lines – do not give any code).

5. (1.5 point) The program for the transitive closure of the union can be written in Webdamlog as follows:

```
G@p(x,y) :- G_1@p(x,y)
G@p(x,y) :- G_2@q(x,y)
T@p(x,y) :- G@p(x,y)
T@p(x,y) :- T@p(x,z), G@p(z,y)
```

   Write in Webdamlog the program "cousin of the same generation" of question 2.

2 Probabilistic XML (13 points)

Let $\mathcal{L}$ be a countable set of labels.

In this exercise, we see an XML document as a finite labeled, *unordered*, unranked tree: formally, an XML document d is a triple (V, E, r, λ) where V is a finite set of nodes, $E \subseteq V^2$ is a set of edges such that (V, E) is a directed tree rooted at $r \in V$, and $\lambda : V \to \mathcal{L}$ is a labeling function. In other words, we disregard the following features of the XML format and of the Document Object Model: order between siblings, nodes of type different from Element (in particular, there are no attribute, document, or text nodes). Two trees $d = (V, E, r, \lambda)$ and $d' = (V', E', r', \lambda')$ are isomorphic (denoted $d \sim d'$) if there is a bijection $\varphi : E \to E'$ such that $(\varphi(u), \varphi(v)) \in E'$ if and only if $(u, v) \in E$; $\varphi(r) = \varphi(r')$; and $\lambda(\varphi(u)) = \lambda(u)$ for all $u \in V$.

We say that a document d matches a Boolean query Q, denoted $d \models Q$, if Q is true over d.

We consider the following (Boolean) query Q_0 over XML documents: "there exists a node labeled by a that has *no* child labeled by b".

1. (1 point) Express in the XPath 1.0 language the query Q_0 (you can assume the query will be interpreted in a Boolean context).

2. (1 point) Give an example of an XQuery query Q_1 not expressible in XPath 1.0. No proof is required.

3. (1.5 points) Propose a linear-time algorithm to, given a XML document d, decide whether $d \models Q_0$. Prove the correction and the complexity of your algorithm.

4. (1 point) Does your algorithm extend to Q_1? Discuss why this is or is not the case.

   A *probabilistic XML space*, or *px-space*, is a finite probability distribution over XML documents, i.e., a pair $(D, \Pr)$ where D is a finite set of non-isomorphic XML documents *with the same root label* and $\Pr : D \to (0; 1]$ assigns a *rational*

probability to every document in D, such that $\sum_{d \in D} \Pr(d) = 1$. The *probability* of query Q in a px-space $(D, \Pr)$, noted $Q(D, \Pr)$, is the probability $\sum_{\substack{d \in D \\ d \models Q}} \Pr(d)$.

A *simple probabilistic XML document*, or *sp-document*, is a pair $\mathcal{D} = (d, p)$ where $d = (V, E, r, \lambda)$ is an XML document and $p : V \to (0; 1]$ assigns to every node of d a *rational* probability, with $p(r) = 1$. Given a *valuation* $\nu : V \to \{0, 1\}$ with $\nu(r) = 1$, $\nu(\mathcal{D})$ is the subtree of d obtained by removing all nodes that ν maps to 0, *along with their descendants*. The *semantics* of an *sp-document* $\mathcal{D} = (d, p)$, denoted $[\![\mathcal{D}]\!]$, is the px-space $(D, \Pr)$ obtained as follows: D is the set of all subtrees of d rooted at r, keeping only one representative per isomorphism class for $\sim$; and, for $d \in D$:

$$\Pr(d) \coloneqq \sum_{\substack{\nu \\ \nu(\mathcal{D}) \sim d}} \prod_{\substack{u \in d \\ \nu(u)=1}} p(u) \prod_{\substack{u \in d \\ \nu(u)=0}} (1 - p(u)).$$

5. (1.5 point) Prove that there exists a px-space $(D, \Pr)$ such that there is no sp-document $\mathcal{D}$ with $[\![\mathcal{D}]\!] = (D, \Pr)$.

6. (2 points) Propose a linear-time algorithm to, given an sp-document document $\mathcal{D}$, compute $Q_0([\![\mathcal{D}]\!])$. Prove the correction and the complexity of your algorithm.

An *event probabilistic XML document*, or *ep-document*, is a 4-uple $\mathcal{D} = (d, \Omega, f, p)$ where $d = (V, E, r, \lambda)$ is an XML document, Ω is a finite set of *events*, f assigns to every node of d a propositional logic formula (with the true formula assigned to r) over the variables Ω, and $p : \Omega \to (0; 1]$ assigns to every event of Ω a *rational* probability. Given a *valuation* $\nu : \Omega \to \{0, 1\}$, $\nu(\mathcal{D})$ is the subtree of d obtained by removing all nodes u such that $\nu(f(u))$ is false, *along with their descendants*. The *semantics* of an *ep-document* $\mathcal{D} = (d, \Omega, f, p)$, denoted $[\![\mathcal{D}]\!]$, is the px-space $(D, \Pr)$ obtained as follows: D is the set of all subtrees of d rooted at r, keeping only one representative per isomorphism class for $\sim$; and, for $d \in D$:

$$\Pr(d) \coloneqq \sum_{\substack{\nu \\ \nu(\mathcal{D}) \sim d}} \prod_{\substack{\omega \in \Omega \\ \nu(\omega)=1}} p(\omega) \prod_{\substack{\omega \in \Omega \\ \nu(\omega)=0}} (1 - p(\omega)).$$

7. (1.5 points) Prove that, for every px-space $(D, \Pr)$, there exists an ep-document $\mathcal{D}$ with $[\![\mathcal{D}]\!] = (D, \Pr)$.

A problem is in the counting complexity class #P if it can be expressed as counting the number of accepting paths of a non-deterministic polynomial-time Turing machine. A problem is #P-hard if any #P problem reduces to it, and #P-complete if it is both in #P and #P-hard. An example of #P-complete problem is #DNF: counting the number of satisfying assignments of a propositional formula in disjunctive normal form (disjunction of conjunctions of literals). A problem is in $\mathrm{FP}^{\#\mathrm{P}}$ if it is solvable in polynomial time using a #P oracle.

8. (2 points) Prove that, given an ep-document $\mathcal{D}$, computing $Q_0([\![\mathcal{D}]\!])$ is a #P-hard problem, even if f is restricted to map nodes to conjunctions of literals.

9. (1.5 points) Prove that, given an ep-document $\mathcal{D}$, computing $Q_0([\![\mathcal{D}]\!])$ is in $\mathrm{FP}^{\#\mathrm{P}}$.