Exam 2018 – Web Data Management

S. Abiteboul & P. Senellart

March 2018

The exam is 2 hours. The only document allowed is a single A4 sheet. Internet access and communicating devices are disallowed. The two exercises should by done on separate sheets, as they will be graded separately.

1 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 $\mathrm{FP}^{\#\mathrm{P}}$.

2 Diversity (7 points)

Search engines, recommender systems, information discovery systems usually favor relevance. To ensure that a recommended item is perceived as relevant, personalization methods tend to be conservative – they select items (Web page, products, persons, tasks, etc.) that are relevant to a query, and very similar to what you liked before. While this indeed increases the likelihood that you will like the items being recommended to you, there are also clearly negative effects: This strategy focuses your attention on a small number of items, and leads you to ignore the rest of the items. The effect of aggressive personalization is that it limits your horizons, and brings the risk of locking you into an information bubble.

But software can act differently. It can be designed to give less importance to relevance, allowing you to discover a broader variety of items. It can steer you away from your usual experiences and

bring serendipity – help you discover new information, meet people that you never thought you would like, and form new experiences.

1. (2 points) Describe a software application, and a context of use, where diversity is important, and propose an informal description of diversity in this setting.
2. (2 points) Propose a formal definition of the diversity criterion proposed in the previous question.
3. (2 points) Propose an algorithm for your software application that satisfies the diversity criterion, while still providing relevant results.
4. (1 point) Discuss the complexity of your algorithm, its limitations, etc.