Exam 2014 – Web Data Management

S. Abiteboul & P. Senellart

March 2014

2 hours.

No documents allowed. No computer. No calculator.

1 Full text indexing (7 points)

Consider the following corpus:

d_1	The Eiffel tower is the most famous monument in Paris.
d_2	If you go to Paris, do not miss the Louvre museum!
d_3	The Anatomy museum, in Pavia, is very interesting.
d_4	The Ashmolean museum (in Oxford) is the oldest museum in Europe.
d_5	Paris has numerous museums, such as the Orsay museum.
d_6	The jewels of the Tower of London are famous.
d_7	The Luxembourg gardens are in Paris, not in Luxembourg.

1. Apply the following preprocessing steps to it: *tokenizing* (splitting the document into tokens), *morphologic stemming* (removing bound morphemes from word forms), *stop word removal* (removing grammatical words). Give, for each document, the set of remaining terms at the end (not after each step).

2. Build the inverted index of the resulting set of documents. To simplify, terms occurring in one document only will be ignored. One will use tf-idf to sort the index. (It is not necessary to actually perform the exact computation; more precisely, it is OK to give scores using fractions and logarithms; no need for floating point approximations.) Reminder:

$$\text{tf-idf}(t,d) = \frac{n_{t,d}}{\sum_{t'} n_{t',d}} \times \log \frac{|D|}{|\{\, d \in D \mid n_{t,d} > 0 \,\}|}$$

   where $n_{t,d}$ is the number of occurrences of term t in document d and D is the corpus.

3. Assuming the corpus is distributed on a cluster of machines, one would like to build the inverted index using MapReduce. Propose some possible pseudo-code for the functions Map and Reduce that construct a distributed inverted index.

2 Ontologies (5 points)

For this exercise, here is some standard notation for description logics that you can use:

Symbol	Description	Example	Read
$\top$	all concept names	$\top$	top
$\bot$	empty concept	$\bot$	bottom
$\sqcap$	conjunction of concepts	$C \sqcap D$	C and D
$\sqcup$	disjunction of concepts	$C \sqcup D$	C or D
$\neg$	complement of concepts	$\neg C$	not C
$\forall$	universal restriction	$\forall R.C$	all R-successors are in C
$\exists$	existential restriction	$\exists R.C$	an R-successor exists in C
$\exists^{\bullet n}$	cardinality restriction	$\exists^{\leq 2} R.C$	at most 2 R-successors are in C
$\sqsubseteq$	Concept inclusion	$C \sqsubseteq D$	all C are D
$\equiv$	Concept equivalence	$C \equiv D$	C is equivalent to D
$\doteq$	Concept definition	$C \doteq D$	C is defined to be equal to D
:	Concept assertion	$a : C$	a is a C
:	Role assertion	$(a, b) : R$	a is R-related to b

(with $\bullet \in \{\leq, = \geq\}$)

1. We have extracted in some RDF classes and relations knowledge on monuments from the documents of the previous section. (It may help to look at the next question.)

   Propose a class hierarchy and some relations such as `:monument` and `:locatedIn`.

2. Suppose we have the relation `:locatedIn`, and its inverse `:hasAttraction`. Express the following concepts in OWL-DL:
   a) A superb city is a city with at least 4 attractions, of which at least one is a museum.
   b) Each city has at least one attraction.
   c) No museum is in more than one location.
   d) No city has only gardens as attractions.

3 XML typing (5 points)

Consider the following four sets of XML documents:

D_1: ⟨a⟩ ⟨b⟩ ⟨c⟩x⟨/c⟩ ⟨/b⟩ ⟨b⟩ ⟨e/⟩ ⟨/b⟩ ⟨/a⟩
D_2: ⟨a⟩ ⟨b⟩ ⟨c⟩x⟨/c⟩ ⟨/b⟩ ⟨b/⟩ ⟨/a⟩
D_3: ⟨a⟩ ⟨b⟩ ⟨c/⟩ ⟨/b⟩ ⟨b⟩ ⟨c⟩x⟨/c⟩ ⟨c/⟩ ⟨/b⟩ ⟨/a⟩
D_4: ⟨a⟩ ⟨b⟩ ⟨c⟩x⟨/c⟩ ⟨/b⟩ ⟨d⟩ ⟨e⟩y⟨/e⟩ ⟨/d⟩ ⟨/a⟩

where `x` and `y` stand for arbitrary text nodes. Call these sets D_1, D_2, D_3, D_4.

1. For each $D_i \in \{D_1, D_2, D_3, D_4\}$, give a DTD (if one exists) that accepts exactly D_i. Otherwise explain briefly what cannot be captured. The syntax you use for the DTD does not need to be the standard one.

2. For each $D_i \in \{D_1, D_2, D_3, D_4\}$, is there an XML schema that accepts exactly D_i? If yes, you do not need to give it. Otherwise, explain briefly what cannot be captured.

3. Summarize your results of the first two questions in a table of the form:

	D_1	D_2	D_3	D_4
DTD	yes/no	yes/no	yes/no	yes/no
XML Schema	yes/no	yes/no	yes/no	yes/no

4. Each time you answered "no" in the previous question, give the schema (DTD or XML Schema, according to the case) that is as restrictive as you can and validates D_i.

5. Give a DTD that is as restrictive as you can and validates the four sets of documents (i.e., $\bigcup_{i=1}^{4} D_i$).

6. Describe in words (10 lines maximum) an XML Schema as restrictive as you can that validates the four sets of documents (i.e., $\bigcup_{i=1}^{4} D_i$).

4 Probabilitic generation (3 points)

(For this exercise, give short answers. Less than 20 lines total. No proof.)

We are given a top-down deterministic tree automaton A that specifies the type of a set of XML documents. We place probabilities on the transitions of this automaton. This generates documents by choosing transitions randomly according to these probabilities. We see such a top-down deterministic tree as a probabilistic generator of XML documents.

Now see x and y as literal strings, so that each D_i is now a single document. How can we choose the probabilities of a probabilistic generator to maximize the likelihood of the set of documents $\{D_1, \ldots, D_4\}$?

What is the probability for such a probabilistic generation *not* to terminate?