1. Consider the following application description.

A Publisher publishes many Books, but a Book is published by only one Publisher. A Publisher is uniquely identified by a Name and has offices in several Cities. A CityName and State together uniquely identify a City. A Book has a Title, but the Title does not uniquely identify the Book. A Book, however, does have a unique identifying BookNumber. A Book may be adopted by several Universities, and we also want the Date the Book was first adopted by each University. A University is uniquely identified by its UniversityName and is also uniquely identified by its Address.

(a) Identify the functional dependencies in this description. Each noun starting with a capital letter identifies an attribute that must appear in at least one of your FDs. (Use singular names for attributes even if the noun is plural in the description.)

(b) Convert your FDs into a hypergraph. Each FD should be a directed edge in the hypergraph. Add non-functional edges to capture missing many-many relationships among attributes. (In this description there should only be one non-functional edge—a binary edge between Publisher and City.) The nodes in your diagram are to all be rectangles that enclose the name of the node. For lexical object sets (sets whose textual elements denote themselves such as UnivesityName, e.g., “BYU”, and Date, e.g. “September 2011”) make the boarders of the boxes dashed; for non-lexical object sets (sets whose elements require an object identifier such as University—as opposed to UniversityName—and City—as opposed to CityName)

(c) Suppose that the description also includes soft-cover books and hard-cover books: “Books are either SoftCoverBooks or HardCoverBooks, but not both.” Using the triangle symbol for ISA as in the ER model, add to your hypergraph the generalization/specialization to indicate that SoftCoverBooks and HardCoverBooks are both subsets of Book. Add an appropriate symbol to the triangle to indicate that the set of SoftCoverBooks and the set of HardCoverBooks together partition the set of Books.

(d) It should not be hard to see that this exercise yields a conceptual-model instance that captures all the 1-1, 1-m, m-1, and m-n relationships of the application as well as the isa relationships. This process works in general. The result is always a conceptual-model instance that captures the attributes and constraints we need to generate a relational-database schema. Using the general principles for mapping conceptual models to relational database schemas, create a database schema for the publishing application. Although either choice is possible, choose to omit nonlexical attributes when the schema has a lexical attribute (or a composite set of lexical attributes) as a key. This choice may result in making some relation schemas redundant (if so, be sure to drop them). For the ISA hierarchy, choose to collapse the hierarchy to the root and add one new attribute whose values are the names of the specializations. Underline keys.

As a challenge for yourself, you might try to crisply specify an algorithm to generate a relational-database schema from this kind of conceptual model.
2. Based on the description in Problem (1), compute the following functional closures. (Note that you should be able to spot the functional closures immediately from the hypergraph using the hypergraph reachability algorithm discussed in class.) The answers for (a) and (b) should be the same; explain why.

(a) Compute the functional closure of Book.
(b) Compute the functional closure of BookNumber.

3. List all the nontrivial FDs satisfied by the following relation. (An FD $X \rightarrow Y$ is trivial if $Y \subseteq X$.)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

4. Consider the following proposed FD derivation rules. For each rule that is sound, show that it is sound by computing the closure of the left-hand-side of the consequent FD and observing that its right-hand-side attributes are all included in the closure. For each rule that is not sound, give a counterexample—a relation over the attributes that satisfies the premise FDs but not the consequent FD. (Note that the rules mention single attributes, rather than attribute sets. Thus, if the rule mentions $A$, $B$, and $C$, for example, the schema for the relation is $ABC$.)

(a) If $A \rightarrow B$ and $A \rightarrow C$, then $B \rightarrow C$.
(b) If $A \rightarrow B$ and $C \rightarrow D$, then $A \rightarrow D$.
(c) If $A \rightarrow B$ and $BC \rightarrow D$, then $A \rightarrow BD$.
(d) If $A \rightarrow B$ and $C \rightarrow AD$, then $C \rightarrow B$.

5. Let $F = \{A \rightarrow G, AB \rightarrow CD, D \rightarrow EF, BE \rightarrow FG, G \rightarrow C\}$ be a set of FDs.

(a) Give a TAP derivation sequence for $BD \rightarrow C$.
(b) Compute $BD^+$.
(c) Convert the FDs in $F$ into a hypergraph and then mark the nodes that would be marked by running the closure algorithm starting with $BD$.
(d) Show that $F$ does not imply $A \rightarrow E$.

6. Let $F$ and $F'$ be sets of FDs. If $F$ implies each FD $f'$ of $F'$, $F$ implies $F'$, sometimes written $F \models F'$. If $F \models F'$ and $F' \models F$, $F$ and $F'$ are equivalent, and we write $F \equiv F'$. Show: $\{A \rightarrow B, AB \rightarrow C, B \rightarrow CD, A \rightarrow C\} \equiv \{A \rightarrow BC, B \rightarrow CD\}$. 